

# Программчлалын “Си” хэл

---

## Оршил

Анх “Bell Laboratores” фирмийн төлөөлөгчид гаргасан. 1983 онд "Си" хэлний стандартыг боловсруулах хороо байгуулагдсан. 1986 онд энэ хороо "Си" хэлний өөрийн ажлын тайланг хэвлүүлсэн ба 10 сард бүх нийтээр хэлэлцүүлсэн. 1989 онд "Си" хэлний стандартыг баталсан. Энэ хувилбарт ойролцоо дөхсөн нь Turbo C юм. Одоо TC 2.0, TC++ 1.0 гэх мэт мөн Borland C хувилбарууд хэрэглэгдэж байна. Энэхүү ном нь ХоМТС-н 1,2,3-р курсынханд зориулагдав. Энэ номны давуу талд нь номон дээрх жишээ файлууд нь дискэн дээр өгөгдсөн нэрээрээ (жишээ нь prog6\_1.cpp) байгаа учраас та хэрэгтэй программаа дуудаж ажиллуулж түүний талаар ойлголт авч болно.

## I бүлэг

### “Си” хэлний элементүүд

Программчлалын хэлний элемент гэдгээр тухайн хэлний ашиглаж байгаа суурь бүтцийг нэрлэнэ. “Си” хэлний элементүүдэд:

1. Цагаан толгой
2. Тогтмолууд
3. Идентификатор
4. Түлхүүр үгс
5. Тайлбарууд гэх мэт орно.

Компиляторыг “Си” хэл дээр бичигдсэн текст, текстэн мөрөнд ашиглана.

“Си” компилятор нь текстэе мөр мөрөөр нь салгаад, мөрүүдийг дотор нь лексем болгоно. Лексем нь тухайн компиляторын хувьд биеэ даасан утга бүхий программын текстэн нэгж байна. "Си" хэлний элемент болох дээрх идентификатор, тогтмолууд, түлхүүрүүд нь лексем болно. Мөн нээсэн, хаасан хаалтууд “,” болно. "Си" хэлний элементүүдийн баримтлах зүйл нь хоосон зай үлдээж болохгүй. Харин лексем болох зүйлийн тэмдэгт мөр нь хоосон зайгаар тасалдан бичигдэж болно.

Жишээ: a---b. Энэ жишээний хувьд эхлээд a-b тоог хасна. Дараа нь a— үйлдэл хийгдэх буюу a хувьсагчийн утга 1-ээр хорогдоно.

Жишээ: a- --v Энэ жишээний хувьд эхлээд b— үйлдэл хийгдэх буюу b хувьсагчийн утга 1-ээр хорогдоно. Дараа нь a-b тоог хасна.

# Программчлалын “Си” хэл

Дээрх 2 жишээний хувьд авч үзэхэд хоосон зайнаас хамаараад программын үр дүн өөр өөр гарч болох талтай байна. Иймд дурын газар хоосон зай тавьж болохгүй нь харагдаж байна.

*Жишээ программ: Prog1\_0.cpp*

```
#include<conio.h> // Программын өмнө заавал бичигдэх толгой файлууд.
#include<stdio.h>
#include<stdlib.h>
main() // Үндсэн функц.
{
int i=3, j=5; // Хувьсагчууд зарлаж байгаа хэсэг.
clrscr(); // Дэлгэц цэвэрлэх функц.
printf("\ni=%i\tj=%i",i,j); // i, j хувьсагчуудын утгуудыг хэвлэх функц.
printf("\ti+++j=%i",i+++j); // i+++j -н үр дүнг дэлгэцэнд хэвлэх.
printf("\n*After result*\ti=%i\tj=%i",i,j); //хувьсагчуудын өөрчлөлтыг үзэх
i=3; j=5;
printf("\ni=%i\tj=%i",i,j); // i, j хувьсагчуудын утгуудыг хэвлэх функц.
printf("\ti+ ++j=%i",i+ ++j); // i+ ++j -н үр дүнг дэлгэцэнд хэвлэх.
printf("\n*After result*\ti=%i\tj=%i",i,j); //хувьсагчуудын өөрчлөлтыг үзэх
i=3; j=5;
printf("\ni=%i\tj=%i",i,j); // i, j хувьсагчуудын утгуудыг хэвлэх функц.
printf("\ti++ +j=%i",i++ +j); // i++ +j -н үр дүнг дэлгэцэнд хэвлэх.
printf("\n*After result*\ti=%i\tj=%i",i,j); //хувьсагчуудын өөрчлөлтыг үзэх
getch();
}
```

Дэлгэцэнд:

```
i=3    j=5    i+++j=8
*After result*  i=4        j=5
i=3    j=5    i+ ++j=9
*After result*  i=3        j=6
i=3    j=5    i++ +j=8
*After result*  i=4        j=5
```

**“Си” хэлний цагаан толгой**

Үүнд : Том, бага латин үсгүүд, 0-9 цифрүүд, хоосон зайн тэмдэгт, тав, мөр шилжүүлэх, мөрийн төгсгөлийн тэмдэгтүүд гэх мэт орно.

# Программчлалын “Си” хэл

---

Тусгаарлагч элементүүд:

, . ; ? ‘ ( ) { } [ ] ! | \ / ~ # \$ % & ^ \_ + \* гэх мэт тэмдэгтүүд орно.

Эдгээр нь янз бүрийн үүрэг гүйцэтгэнэ.

Шинэ тусгай тэмдэгт:

\n- Шинэ мөрөнд шилжих. Enter товчлуур дарсантай ижил үүрэг гүйцэтгэнэ.

\t -Хэвтээ мөрөнд (tab) зай шилжинэ.

\v -Босоо мөрөнд (tab) зай шилжинэ.

\b -Нэг тэмдэгт уригшаа шилжинэ.

\r -Мөрийн эхэнд шилжинэ.

\f -Шинэ хуудсанд шилжинэ.

\a -Веер Сигнал өгөх.

\’ -Апостроф хэвлэнэ.

\” -Давхар хашилт хэвлэнэ

\\ -Урвуу слэш хэвлэнэ.

\ddd - 8-тын тооллын системд байт бичих.

\xddd- 16-тын тооллын системд байт бичих.

Үйлдлийн тэмдэг:

! -Логикийн үгүй (NOT).

~ -Урвуу кодыг гаргана.

+ -Нэмэх. 2 операндыг нэмэх үйлдэл.

- -Хасах. 2 операндыг хасах үйлдэл.

\* -Үржих. 2 операндыг үржих үйлдэл. Мөн тухайн хаяг доторх утга авах үйлдэлийг хийнэ.

/ -Хуваах. 2 операндыг хуваах үйлдэл.

% -Үлдэгдэл тооцоох. 2 операндыг хувааж үлдэгэлийг нь авна.

<< -Зүүн тийш нэг битээр шилжүүлэх.

>> -Баруун тийш нэг битээр шилжүүлэх.

< -Бага.

<= -Бага буюу тэнцүү.

> -Их.

>= -Их буюу тэнцүү.

== -Тэнцүү.

!= -Тэнцүү биш.

& -Хаягийн үйлдэл хийнэ. Тоонд бит бүрээр AND хийх үйлдэл.

| -Тоонд бит бүрээр нь OR хийх үйлдэл.

# Программчлалын “Си” хэл

---

- $\wedge$  -Тоонд бит бүрээр нь XOR хийх үйлдэл.
- $++$  -Нэгээр нэмэгдүүлэх үйлдэл.
- $--$  -Нэгээр хорогдуулах үйлдэл.
- $+=$  -Нэмэгдүүлж утга оноох үйлдэл.
- $-=$  -Хорогдуулж утга оноох үйлдэл.
- $*=$  -Үржүүлж утга оноох үйлдэл.
- $/=$  -Хувааж утга оноох үйлдэл.
- $>>=$  -Тоог баруун тийш нэг битээр шилжүүлж утга оноох үйлдэл.
- $<<=$  -Тоог зүүн тийш нэг битээр шилжүүлж утга оноох үйлдэл.
- $\%=$  -Үлдэгдэл тооцож утга оноох үйлдэл.
- $\&=$  -Тоонд бит бүрээр AND хийж утга онооно.
- $|=$  -Тоонд бит бүрээр OR хийж утга онооно.
- $\wedge=$  -Тоонд бит бүрээр XOR хийж утга онооно.

## Тогтмолууд

"Си" хэлэнд 4 төрлийн тогтмол байдаг.

- a. Бүхэл тоон тогтмол.
- b. Бодит тоон тогтмол.
- c. Тэмдэгтэн тогтмол.
- d. Тэмдэгтэн мөрөн тогтмол.

### а. Бүхэл тоон тогтмол.

Бүхэл тоон тогтмолууд бутархай хэсэггүй бөгөөд 8-т, 10-т, 16-тын тооллын системүүдээр илэрхийлэгдэнэ.

Бичигдэх хэлбэр:

- $\langle \text{цифр} \rangle$  - 10-тын тооллын системд байгаа тоо
- $0\langle \text{цифр} \rangle$  - 8-тын тооллын системд байгаа тоо
- $0x\langle \text{цифр} \rangle$  - 16-тын тооллын системд байгаа тоо

Жишээ: 10 тоог авч үзье.

10- 10-тын тооллын системд

017- 8-тын тооллын системд

0xF (0XF)- 16-тын тооллын системд гэж тус тус бичигдэнэ.

Бүхэл тоон төрлийг дотор нь хэмжээнээс нь хамааруулж int, unsigned int, long, unsigned long гэсэн төрлүүдэд хуваана.

## Программчлалын “Си” хэл

---

төрөл	урт	хэмжээ
Int	16 бит	-32768 – 32767
Unsigned int	16 бит	0 – 65535
long	32 бит	-2147483648 – 147483647
unsigned long	32 бит	0 – 4294967295

### *Хувьсагчийн зарлалт*

Бичигдэх хэлбэр:

```
int <хувьсагчийн нэр>[<=утга олгох>;
unsigned int <хувьсагчийн нэр>[<=утга олгох>;
long <хувьсагчийн нэр>[<=утга олгох>;
unsigned long <хувьсагчийн нэр>[<=утга олгох>;
```

Жишээ: int n=10, num, number=32000;  
unsigned int age, size=65000, count=5;  
long phone\_number=99197878;  
unsigned long pac=321422, mul;

### *Жишээ программ: Prog1\_1.cpp*

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
{
int n=10; // Бүхэл тоон хувьсагч зарлах хэсэг
unsigned int num=65000;
long phone_num;
int age;
clrscr(); // Дэлгэц цэвэрлэдэг функц
phone_num=99198070; // phone_num хувьсагчид утга оноох
age=30; // Бүхэл тоон age хувьсагчид утга оноох
printf("integer n=%i\tunsigned int num=%u",n,num);
printf("\nage=%d\t\tToo=%i",age,234);
printf("\nLong type=%ld",phone_num); // Long төрлийн хувьсагчийг хэвлэх
/* Long төрлийн хувьсагчийг integer төрлөөр хэвлэх*/
printf("\nLong type print to integer=%d",phone_num);
```

## Программчлалын “Си” хэл

```
getch();  
}
```

Дэлгэцэнд:

```
integer n=10    unsigned int num=65000
```

```
age=30         Тоо=234
```

```
Long type=99198070
```

Long type print to integer=-23434 /\* long төрлийг integer төрөлд хэвлэхэд хасах тоо гарна. Учир нь integer төрөл нь long төрлөөс бага юм.\*/

### б. Бодит тоон тогтмол.

Хөвөгч таслалтай тогтмолууд орно.

Бичигдэх хэлбэр:

<цифр><.><цифр>[<E+><цифр>] эсвэл

<цифр><.><цифр>[<E-><цифр>] болно.

Жишээ: 0.2, 2.4, 4.3E-1, 3.2E+12 гэх мэт.

Хэмжээнээс нь хамааруулж double, float, long double гэсэн төрлүүдэд хуваана.

төрөл	урт	хэмжээ
float	32 бит	$3.4 * (10^{**-38}) - 3.4 * (10^{**+38})$
double	64 бит	$1.7 * (10^{**-308}) - 1.7 * (10^{**+308})$
long double	80 бит	$3.4 * (10^{**-4932}) - 1.1 * (10^{**+4932})$

### Хувьсагчийн зарлалт

Бичигдэх хэлбэр:

```
double    <хувьсагчийн нэр>[<=утга олгох>];
```

```
float     <хувьсагчийн нэр>[<=утга олгох>];
```

```
long double <хувьсагчийн нэр>[<=утга олгох>];
```

Жишээ: double count=5.6, number =5.6E16, real\_num=3.2E-4;

```
float     count1, real_count =4289e-26;
```

```
long double big_num, big_number=3245.7845;
```

```
double    num=2.3; float num2=742.3E27;
```

### Жишээ программ: Prog1\_2.cpp

```
#include<conio.h>
```

```
#include<stdio.h>
```

## Программчлалын “Си” хэл

```
#include<stdlib.h>
main()
{
double num=4.78E2;    // Бодит тоон хувьсагч зарлах хэсэг
float num_f=3.5;
clrscr();
printf("%e\t%f",2.4E+5,2.4E+5);    // Бодит тоо хэвлэх
printf("\n%3.3e\t%3.3f",2.4E-5,2.4E-2);    // Бодит тоо хэвлэх
printf("\n%e",num);    // Бодит тоог E зэргээр бодит тоог илэрхийлж хэвлэх
printf("  %f",num);    // Бодит тоог таслалтай байдлаар илэрхийлж хэвлэх
printf("  %3.3f",num);    // Таслалаас өмнө 3, хойно 3 оронтойгоор хэвлэх
printf("  %3.3e",num);
printf("\n**Print num_f value**\n");
printf("%e\t%f",num_f,num_f);
printf("\n%2.2e\t%2.2f",num_f,num_f);
getch();
}
```

Дэлгэцэнд:

2.400000e+05            240000.000000

2.400e-05            0.024

4.780000e+02   478.000000   478.000        4.780e+02

\*\*Print num\_f value\*\*

3.500000e00    3.500000 // таслалтайгаар мөн E зэрэгтээгээр хэвлэх

3.50e00        3.50        // E ба таслалаас өмнө 2, хойш 2 оронгоор

### с. Тэмдэгтэн тогтмол.

Тэмдэгтэн тогтмол дан хашилт ( ‘ ’ ) дотор бичигдэх ба үсэг, цифр, тусгай тэмдэгтүүдийг хэрэглэнэ. Тэмдэгтэн тогтмол нь char, unsigned char төрлийн байна.

төрөл	урт	хэмжээ
unsigned char	8 бит	0 – 255
Char	8 бит	-128 - 127

*Хувьсагчийн зарлалт*

## Программчлалын “Си” хэл

---

Бичигдэх хэлбэр:

```
char <хувьсагчийн нэр>[<=утга олгох>];  
unsigned char <хувьсагчийн нэр>[<=утга олгох>];
```

Жишээ:

```
char          char_value, single_char='a';  
char          value1='I';  
unsigned char ch='\n';
```

*Жишээ программ: Prog1\_3.cpp*

```
#include<conio.h>  
#include<stdio.h>  
#include<stdlib.h>  
main()  
{  
char ch='H';          // Тэмдэгтэн хувьсагч зарлах хэсэг  
unsigned char ch1;  
clrscr();            // Дэлгэц цэвэрлэх функц  
ch1='e';             // ch1 хувьсагчид тэмдэгт оноох  
printf("\n%c%c ",ch,ch1);  
ch='i'; ch1='s';     // ch1 хувьсагчийн утгыг өөрчлөх  
printf("%c%c",ch,ch1); printf(" Bob.");  
getch();  
}  
  
    Дэлгэцэнд:  
    He is Bob.          // Дэлгэцэнд гарах өгүүлбэр
```

### d. Тэмдэгтэн мөрөн тогтмол.

Тэмдэгтэн мөрөн тогтмол нь давхар хашилт (“ ”) дотор бичигдэх ба үсэг, цифр, тусгай тэмдэгтүүдийн дараалал байна. Үүнийг төрлийнхөө хувьд тэмдэгтэн массив байна гэж ойлгож болно. “ABC” тэмдэгтэн тогтмол байна гэж үзвэл утгууд нь санах ойн дараалсан хаягт байрлах ба эцэст нь ‘\0’ тэмдэгт автомаатаар бичигдэнэ. Өөрөөр хэлбэл санах ойд ABC\0 гэсэн дарааллаар байрлана. Олон тэмдэгттэй тэмдэгтэн мөр байгаа тохиолдолд \ тэмдэгтээр тасалж дараагийн мөрөнд бичиж болно. Жишээ нь: p=”ABC”;



## Программчлалын “Си” хэл

---

p="AB"\"C"; p="AB"\"C"; -эдгээр нь бүгд ижил утгатай. Тэмдэгтэн мөрөн тогтмол нь тэмдэгтэн тогтмолтой ижил төрөлтэй байна.

### *Хувьсагчийн зарлалт*

Бичигдэх хэлбэр:

```
char *<хувьсагчийн нэр>[<=утга олгох>];  
char <хувьсагчийн нэр>[<индекс>][<=утга олгох>]; гэх мэт.
```

Жишээ:

```
char *my_string, *str[]={“My”,“name”,“is”,“Bob”}  
char mas[10], string_mas[100]=“Programm”;  
char *value1=“I am Bob”;  
char mas1[]=“Proramming language C”;
```

### *Жишээ программ: Prog1\_4.cpp*

```
#include<conio.h>  
#include<stdio.h>  
#include<stdlib.h>  
main()  
{  
char *sh=“I am a worker.”; // Тэмдэгтэн мөрөн хувьсагчуудын зарлалтууд  
char sh1[20]=“I am a"" worker.”;  
char sh2[20]=“I am a\ worker.”;  
char *sh3=“Borlandc”;  
char *sh4=“Bor\  
landc”;  
clrscr();  
printf(“\n%s\t%s\t%s”,sh,sh1,sh2); // Тэмдэгтэн мөрөн хувьсагчуудыг хэвлэх  
printf(“\n%s\t%s”,sh3,sh4);  
getch();  
}
```

Дэлгэцэнд:

```
I am a worker. I am a worker. I am a worker.  
Borlandc Borlandc
```

# Программчлалын “Си” хэл

## Идентификаторууд

Программд ашиглагдаж байгаа хувьсагч болон тэмдэгтүүдийн нэрийг идентификатор гэнэ. Энэ нь үсгээр эхэлсэн нэг буюу түүнээс дээш тэмдэгтүүдийн дараалал юм. Өөрөөр хэлбэл үсгүүдийн дарааллыг үг буюу идентификатор гэж нэрлэх буюу идентификатор гэдэг нь программ дахь тогтмол, хувьсагч, функцуудын нэр юм.

Идентификатор дотор хоосон зай авч болохгүй. ТС 1.5 хувилбараас эхлээд \$ тэмдэг орохыг зөвшөөрсөн. Мөн доогуур зураасаар (`_bb`) эхэлсэн байж болно. "Си" хэл бусад хэлнээс ялгагдах ялгаа нь том, жижиг үсгийг хоёр өөр идентификатор гэж ойлгодог. Идентификатор нь түлхүүр үгнүүдтэй давхцахгүй байх ёстой.

## Түлхүүр үг

Энэ нь "Си" хэлний компиляторт урьдчилан тодорхойлогдсон идентификаторууд юм. Эдгээр түлхүүр үгсийг хэрэглэх заавар байдаг.

auto	extern	short	sizeof	sm
-cs	-ss	break	float	signed
while	cdecl	-ds	-es	char
for	static	do	far	-BX
-CL	case	goto	struct	huge
-CH	-CX	continue	if	switch
interrupt	-AH	-DH	default	int
typedef	near	-AL	-DL	double
long	union	pascal	-AX	-DL
else	register	unsigned	const	-BH
-DX	enum	return	void	volatile
-BL	-SI			

Программыг ашиглахад хялбар эмх цэгцтэй болгоход түлхүүр үгнүүдийн нэрийг сольж болно.

## Тайлбар

Энэ нь хоосон мөр мэтээр ойлгогдох тэмдэгтүүдийн дараалал юм. Тайлбарын эхлэл төгсгөлийг `/* */` тэмдгүүдээр таньдаг. Иймээс тайлбарыг

## Программчлалын “Си” хэл

---

программын аль ч хэсэгт хэрэглэж болно. Энэ нь программын аль нэг хэсгийг ойлгомжтой болгох үүднээс тайлбарлах зорилготой.

Бичигдэх хэлбэр нь:

```
/*<Тэмдэгтүүдийн дараалал>*/
```

Тайлбар дотор тайлбар оруулж болохгүй.

Мөн сүүлийн үеийн хувилбарт тайлбарыг // гэж тэмдэглэж болно. Энэ нь бичигдсэн газраасаа хойшоо тухайн мөрийг дуустал талбар болгож байна гэсэн үг.

*Жишээ программ: Prog1\_5.cpp*

```
/* Тайлбарыг хэрхэн хэрэглэж болох жишээ программ*/  
#include<conio.h>  
#include<stdio.h>  
#include<stdlib.h>  
main()  
{  
/* Тэмдэгтэн мөр зарлаж байна.*/  
char *sh="Today is Monday. I am a worker."  
//Дэлгэц цэвэрлэх функц.  
clrscr();  
//Дэлгэцэнд sh тэмдэгтэн мөрийг хэвлэж байгаа функц.  
printf("%s",sh);  
/*Гараас товчлууур дартал үр дүнг дэлгэцэн дээр харуулах функц.*/  
getch();  
}
```

Дэлгэцэнд:

```
Today is Monday. I am a worker. // Дэлгэцэнд гарах өгүүлбэр
```

# Программчлалын “Си” хэл

---

## II бүлэг

### Программын бүтэц

Программын бүтцэд үндсэн программ, программын файлууд, программын биелэлт, үйлчлэх хүрээ ба амьдрах хугацаа гэх мэт орно.

### Үндсэн программ

Үндсэн программ нь дараахь элементүүдийг агуулна.

1. Препротессорын удирдлагууд.
2. Компиляторын заалтууд.
3. Зарлалтууд.
4. Программын бие.

Препротессорын удирдлагууд гэдэг нь компиляци хийхийн өмнө программын текстийг өөрчлөх удирдамжууд юм.

Компиляторын заалтууд. Компиляци хийх явцад компиляцийн ажиллагааг удирдах тусгай удирдамжууд байна.

Зарлалтууд гэдэг нь хувьсагч, тэмдэгт, функцуудын зарлалт байна.

Хувьсагчийн зарлалтаар ямар төрлийн хувьсагч байгаагаас хамаараад тухайн хувьсагчид зориулсан санах ой нөөцлөгдөнө. Зарлах үед тухайн хувьсагчийн анхны утга нь тодорхой эсвэл тодорхой бус байна. Хувьсагчийн ангиллаас хамаараад автомат, регистр, статик, гадаад хувьсагч гэж хуваана.

Функцын зарлалтаар түүний нэр, функцын буцаах утгын төрөл тодорхойлогдох ба түүнд дамжигдах параметруудийг тодорхойлж болно. Программ дотор хичнээн ч функц хэрэглэж болно. Энэ нь программын үүргээс хамаарна. "Си" дээр бичигдсэн программ ажиллахын тулд хамгийн багадаа нэг функцтэй байна. Хэрэв "Си" нь ганцхан функцтэй бол тэр нь үндсэн функц (main) байна.

"Си" дээр бичигдсэн ямар ч программ main() функцээс эхэлж ажилладаг ба энэ чанараараа бусад функцуудээсээ ялгагдана. Дээрхээс бусдаараа функцууд нь адил тэгш эрхтэй байдаг.

```
main() // Үндсэн функцын нэр
{      // Үндсэн функцын буюу программын эхлэл
}      // Үндсэн функцын буюу программын төгсгөл
```

# Программчлалын “Си” хэл

---

## Программын файл

Энэ нь нэг ба хэд хэдэн файлаас тогтоно. Хэд хэдэн файлаас программын файл бүрдэж байвал эдгээрийг холбох хоёр арга байна.

1. Файлуудыг тус тусад нь компиляци хийж сүүлд нь холболт хийнэ. (LINK.EXE-холбох)
2. Компиляци хийхийн өмнө файлуудыг препроцессорын #include удирдлагаар нэг файлд нэгтгэж компиляци хийж дараа нь холболт хийнэ.

## Программын биелэлт

"Си" программ бүр нь үндсэн функцтэй байна. Бүх функцүүд тэр тусмаа main() функц нь аргументтэй байж болно. Аргументийг гараас өгч болно. main() функц нь хоёр аргументтэй.

1. int argc
2. char \*argv[]

Эхний аргумент нь бүхэл тоо, хоёр дахь нь тэмдэгтэн мөрийн массив байна. Мөн аргументгүй тохиолдолд main() гэж бичиж болно.

Функцын аргументийг бичих 2 арга байдаг.

- a. main(argc,argv)  
int argc;  
char \*argv[];  
{  
}

*Жишээ программ: Prog2\_1.cpp*

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
// Аргументуудын төрлүүд нь тодорхой бус байна.
void main(argc,argv)
// Аргументуудын төрлүүдийг зарлаж байгаа хэсэг.
int argc;           // Бүхэл тоон аргумент
char *argv[];      // Тэмдэгтэн мөрөн аргумент
{
int i;
```

## Программчлалын “Си” хэл

---

```
clrscr();
printf("Arguments count= %i",argc);    // Аргументийн тоо.
// Аргументийн тоогоор давталт хийж хэвлэж байгаа хэсэг.
for(i=0;i<argc;i++)
printf("\n%i Argument name= %s",i,argv[i]);
getch();
}
```

b. main(int argc, char \*argv[])

```
{
}
```

*Жишээ программ: Prog2\_2.cpp*

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
// Аргументуудын төрлүүд нь тодорхой байна.
void main(int argc, char *argv[])
{
int i;
clrscr();
printf("Arguments count= %i",argc);    // Аргументийн тоо
// Аргументийн тоогоор давталт хийж хэвлэж байгаа хэсэг.
for(i=0;i<argc;i++)
printf("\n%i Argument name= %s",i,argv[i]);
getch();
}
```

Программын аргументуудыг хоёр янзаар тодорхойлж өгч болно. (Bat, Bold, Baatar-аргументууд) Дээрх жишээнүүдэд гурван аргумент өгөгдөнө. argc хувьсагчид программын аргументуудын тоо, argv хувьсагчид аргументуудын тэмдэгтүүд өгөгдөнө.

- a. Компиляци хийсний дараа Prog2\_1.exe файл үүснэ. Энэ файлыг ажиллуулахдаа MS-DOS-н командын мөрөнд Prog2\_1.exe Bat Bold Baatar гэж бичиж ажиллуулна.
- b. Компиляци хийхийн өмнө буюу Borlandc-ийн редакторын Run цэсний Arguments... хэсэгт Bat Bold Baatar гэж аргументийг тодорхойлж өгнө.

## Программчлалын “Си” хэл

---

Жич: Prog2\_1.exe файлын нэр (түүний агуулагдаж байгаа зам) нэг аргумент болж ордог.

Дэлгэцэнд:

```
Arguments = 4           // Гараас өгсөн аргументийн тоо
Argument 0=c:\borlandc\out\Prog2_1.exe
Argument 1=Bat          // Гараас өгсөн эхний аргумент
Argument 2=Bold         // Гараас өгсөн 2 дахь аргумент
Argument 3=Baatar       // Гараас өгсөн 3 дахь аргумент
```

### Амьдрах хугацаа үйлчлэх хүрээ.

Хувьсагчууд нь гадаад буюу ерөнхий (global), дотоод (local) хувьсагчууд гэсэн хоёр хэсэгт хуваагдана. Гадаад хувьсагч нь main() функцын өмнө нь зарлагдана. Гадаад хувьсагч нь бүхэл программд хүчин төгөлдөр.

"Си" хэлний функцүүд нь ерөнхий хэлбэрээр зарлагдана. Функц дотор функц зарлана гэсэн ойлголт байхгүй.

Дотоод хувьсагчууд нь тухайн блокт хүчинтэй байна. Блок нь нийлмэл оператор байж болох ба мөн функцийн биеийг блок гэж ойлгож болно. ( Блок-  
{ } )

Блокын дотор хийгдэх зарлалтыг дотоод зарлалт гэнэ.

### Жишээ программ: Prog2\_3.cpp

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
int global_num;           // Гадаад хувьсагч num-г зарлах
int number=9;           // Гадаад хувьсагч number-д утга оноож зарлах
void main()
{
int i=8;                 // Дотоод хувьсагч i-г зарласан хэсэг
num=10;                 // Гадаад хувьсагч num-д утга оноож байгаа хэсэг
clrscr();
printf("\n\n***Without the block***\n");
printf("\nGlobal value num=%i\tnumber=%i",num,number);
printf("\nLocal value i=%i",i);
{ int i; // Энэ блок дотор зарлагдсан хувьсагчууд дотоод хувьсагчууд болно.
```

## Программчлалын “Си” хэл

---

```
int num;
i=5;
num=6;
number=7;    // number гадаад хувьсагч
printf("\n\n***In the block***\n");
printf("\nGlobal value number=%i",number);
printf("\nLocal value num=%i\ti=%i",num,i);
}
printf("\n\n***Without the block***\n");
printf("\nGlobal value num=%i\tnumber=%i",num,number);
printf("\nLocal value i=%i",i);
getch();
}
```

Дэлгэцэнд:

```
***Without the block***
```

```
Global value num=10      number=9
```

```
Local value i=8
```

```
***In the block***
```

```
//number хувьсагч шинээр зарлагдаагүй, утга нь өөрчлөгдсөн.
```

```
Global value number=7    // Иймд гадаад хувьсагч хэвэндээ
```

```
//Блок дотор шинээр зарлагдаж байгаа учраас num, i дотоод
```

```
Local value num=6      i=5          // хувьсагчууд
```

```
/*Блокоос гарсны дараахь тохиолдолд хувьсагчуудын авах утга*/
```

```
***Without thr block***
```

```
Global value num=10      number=7
```

```
Local value i=8          /* i, num хувьсагчууд блокоос гараад
```

```
өмнөх утгаа авна.*/
```



# Программчлалын “Си” хэл

---

## III бүлэг

### Илэрхийлэл

Илэрхийлэл гэдэг нь үйлдлийн гишүүд ба үйлдлийн зөв хослолыг хэлнэ.  
Жишээ нь:  $a+b*c$

Дээрх тохиолдолд  $a, b, c$ - хувьсагчууд нь операндууд буюу үйлдлийн гишүүд,  $+, *$ - нь операц буюу үйлдлүүд болно. Илэрхийлэл нь тодорхой тооцоологдох дараалалтай.

Операндууд нь цааш илэрхийлэл болж задарч болно. "Си" хэлний бусад хэлнээс ялгагдах зүйл нь утга оноох үйлдэл нь илэрхийлэл болж чаддагт оршино.

### Операндууд

Илэрхийллийн операнд нь идентификаторууд, тогтмолууд, тэмдэгтэн мөрүүд байж болно. Эдгээр нь анхны хэлбэрээр мөн давхараар орж болно. Операндууд өөрийн төрөлтэй байх ба гарсан үр дүн нь мөн өөрийн төрөлтэй байна.

#### а. Идентификаторууд:

1. Бүхэл тоон ба хөвөгч, таслалтай тооны төрлүүдийн идентификатор.

Тухайн төрлийн идентификатор өгнө.

2. Тоочигдох төрлийн идентификатор.

Түүний элементийн төрлийг өгнө. Тоочигдох төрөл нь нэг төрлийн элементүүд дээр хийгдэнэ.

3. Бүтэц ба нэгдлийн идентификатор.

Тухайн бүтцээр эсвэл нэгдлээр тодорхойлогдсон төрлийг өгнө.

```
struct st {  
    int b;  
    char c;  
} a;
```

$a$  нь бүтцийг илэрхийлнэ.  $int\ b$ ,  $char\ c$  хувьсагчуудыг хамтатган тодорхойлж байна.

4. Заагчийн идентификатор.

## Программчлалын “Си” хэл

---

Энэ нь тухайн заагчаар тодорхойлогдож буй объектын хаяг байна.

```
int *a; char *sh; double *pac;
```

a нь бүхэл тоон заагч, sh нь тэмдэгтэн мөрөн заагч, pac нь бодит тоон заагч юм. Энэ нь илэрхийлэлд орж ирэхэд энэ заагчаар заагдсан хаяг орж ирнэ.

5. Массивын идентификатор.

```
int a[5]; char sh[100]="Red Book";
```

a, sh нь илэрхийлэлд орж ирэхэд массивын 0-р элементүүдийн хаягийг төлөөлнө. sh[4]- гэж орсон бол массивын 4-р элементийн элемент буюу 'B' гэсэн тэмдэгтийг авна.

6. Функцын идентификатор.

Тухайн функцэд шилжих хаягийг тодорхойлно.

Функцын идентификаторын өгөх төрөл нь тухайн функцын төрөл байна.

b. Тогтмолууд. Өмнөх бүлгүүдэд үзсэн.

c. Тэмдэгтэн мөрүүд.

Энэ нь өөрөө заагч юм. Иймд илэрхийлэлд заагчуудтай хамт орж болох юм.

```
printf("%s", "look");
```

 Эндээс "look" нь заагч болж байгаа ба тодорхой хаягийг л заана.

### Үйлдлүүд

Энэ хэсэгт арифметик ба бит үйлдлүүдийг тайлбарлана.

### Арифметик үйлдлүүд

+, -, \*, /, % -Эдгээр үйлдлүүд нь 2 операндыг нэмэх, хасах, үржүүлэх, хуваах, үлдэгдлийг нь авах үйлдлийг гүйцэтгэдэг бол ++ нь нэгээр нэмэгдүүлэх, -- нь нэгээр хорогдуулах үйлдлүүдийг хийнэ.

*Жишээ программ:* **Prog3\_1.cpp**

```
#include<conio.h>
```

```
#include<stdio.h>
```

## Программчлалын “Си” хэл

---

```
#include<stdlib.h>
main()
{
int num,number;
clrscr();
num=125; number=-4;
printf("num=%i\n",num);
printf("number=%i\n",number);
printf("\nOperation\n");          // Үйлдлүүдийн үр дүнг хэвлэх хэсэг.
printf("\nnum+number=%i",num+number); // Хувьсагчийн утга өөрлөгдөхгүй.
printf("\nnum-number=%i",num-number); // Хувьсагчийн утга өөрлөгдөхгүй.
printf("\nnum*number=%i",num*number); // Хувьсагчийн утга өөрлөгдөхгүй.
printf("\nnum/number=%i",num/number); // Хувьсагчийн утга өөрлөгдөхгүй.
printf("\nnum number=%i",num%number); // Хувьсагчийн утга өөрлөгдөхгүй.
printf("\nnum+1=%i",++num);          // num хувьсагчийн утга өөрчлөгдөнө.
printf("\nnumber-1=%i",--number); // number хувьсагчийн утга өөрчлөгдөнө.
getch();
}
```

Дэлгэцэнд:

```
num=125      number=-4
***Operation*** // Доор арифметик үйлдлүүдийн үр дүнг харуулсан.
num+number=121
num-number=129
num*number=-500
num/number=-31
num%number=1
num+1=126
number-1=-5
```

$+=$ ,  $-=$ ,  $*=$ ,  $/=$ ,  $\%=$  -Эдгээр нь нэмэгдүүлж, хорогдуулж, үржүүлж, хувааж, үлдэгдэл тооцож утга оноодог үйлдлүүд юм.

*Жишээ программ: Prog3\_2.cpp*

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
```

## Программчлалын “Си” хэл

---

```
{
int num=125,number=-4;
clrscr();
printf("num=%i\t",num);
printf("number=%i\n",number);
printf("***Operation***");
/* num хувьсагчийн утга дээр number хувьсагчийн утгыг нэмээд num
хувьсагчид өгнө. Өөрөөр хэлбэл num=num+number үйлдэлтэй ижил болно.*/
num+=number;
printf("\nnum+=number=%i",num);
num-=number; // num=num-number
printf("\nnum-=number=%i",num);
num*=number; // num=num*number
printf("\nnum*=number=%i",num);
num/=number; // num=num/number
printf("\nnum/=number=%i",num);
num%=number; // num=num%number
printf("\nnum %number=%i",num);
getch();
}
```

Дэлгэцэнд:

```
num=125      number=-4
***Operation*** // Үйлдлүүдийн үр дүнг харуулсан.
num+number=121
num-number=125
num*number=-500
num/number=125
num number=1
```

### Бит дээр хийгдэх үйлдлүүд

"Си" хэл өгөгдлийн бит үйлдэл хийдэг зургаан операторыг хэрэглэдэг. Эдгээр нь char, short int, long, unsigned төрөлтэй операндад хэрэглэгдэнэ. &, |, ^- Эдгээр нь тоог бит бүрээр нь AND, OR, XOR хийдэг үйлдлүүд юм. Доорх хүснэгтэнд хийх үйлдлүүдийг нь харуулав.

## Программчлалын “Си” хэл

x1 операнд	x2 операнд	& AND	 OR	^ XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Жишээ:  $k=20$ ;  $j=12$  бүхэл тоонууд гэвэл  $l=k\&j$  нь

k: 0000 0000 0001 0100

j: 0000 0000 0000 1100

l: 0000 0000 0000 0100 буюу 10-тын тооллын системд  $l=4$  болно.

Жишээ программ: **Prog3\_3.cpp**

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
{
int k=23,j=13,l;
clrscr();
printf("k=%i\tj=%i\n",k,j);
printf("***Bit operation***");
l=k&j; printf("\nk AND j=\t%i",l); // & үйлдэл хийж дэлгэцэн дээр хэвлэх.
l=k|j; printf("\nk OR j=\t%i",l); // | үйлдэл хийж дэлгэцэн дээр хэвлэх.
l=k^j; printf("\nk XOR j=\t%i",l); // ^ үйлдэл хийж дэлгэцэн дээр хэвлэх.
getch();
}
```

Дэлгэцэнд:

```
k=23    j=13    // k=0000 0000 0001 0111  j=0000 0000 0000 1101
```

```
***Bit operation***
```

```
k AND j=    5    // 0001 0111 & 0000 1101= 0000 0101
```

```
k OR  j=    31   // 0001 0111 | 0000 1101= 0001 1111
```

```
k XOR j=    26   // 0001 0111 ^ 0000 1101= 0001 1010
```

Дээрх жишээ программын дэлгэцэнд гарах үр дүнг 2-тын тооллын системд 0 ба 1-р илэрхийлэн бичив.

## Программчлалын “Си” хэл

~- Энэ нь урвуу код буюу инверс хийнэ. (Доорх хүснэгтэнд үзүүлэв.)

X1 операнд	~ урвуу код
0	1
1	0

*Жишээ программ: Prog3\_4.cpp*

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
{
int num=35; unsigned int k=29; // k-г int-ээр биш unsigned-аар зарлах ёстой.
clrscr();
printf("num=%i\tk=%i\n",num,k);
k=~num+k; // num хувьсагчид инверс хийгээд k хувьсагчийн утгыг нэмнэ.
printf("***Bit operation***");
printf("\n~num=%u",~num); // num хувьсагчийн инверсийг хэвлэх
printf("\n~num+k=\t%u",k);
getch();
}
```

Дэлгэцэнд:

num=35 k=29 // num хувьсагч=0000 0000 0010 0011

Bit operation

~num=65500 // 0000 0000 0010 0011 ~ 1111 1111 1101 1100

~num+k=65529 //num хувьсагчийн инверсийг аваад k-нутгыг нэмэх

<<, >> - Зүүн ба баруун тийш нэг битээр шилжүүлэх үйлдэл хийнэ.

Жишээ: i: 0010 1110 1101 1101 үед  
i<<2 1011 1011 0111 0100  
i>>2 0000 1011 1011 0111

*Жишээ программ: Prog3\_5.cpp*

```
#include<conio.h>
#include<stdio.h>
```

## Программчлалын “Си” хэл

```
#include<stdlib.h>
main()
{
    unsigned int num=35,number=29;    // Бит үйлдлийг бүхэл тоон дээр хийнэ.
    clrscr();
    printf("num=%i\tnumber=%i\n",num,number);
    printf("***Bit operation***");
    printf("\nnum<<2=\t%u",num<<2); // num-н утгыг зүүн тийш 2 шилжүүлэх
    printf("\nnum>>4=\t%u",num>>4); // num-н утгыг баруун тийш шилжүүлэх
    printf("\nnumber<<3=\t%u",number<<3);
    printf("\nnumber>>4=\t%u",number>>4);
    getch();
}
```

Дэлгэцэнд:

```
num=35 number=29
***Bit operation***    // Бит үйлдлүүд.
num<<2= 140            //0000 0000 0010 0011<<2=0000 0000 1000 1100
num>>4= 2              //0000 0000 0010 0011>>4=0000 0000 0000 0010
number<<3=232         //0000 0000 0001 1101<<3=0000 0000 1110 1000
number>>4=1           //0000 0000 0001 1101>>4=0000 0000 0000 0001
```

Мөн  $\gg=$ ,  $\ll=$ ,  $\&=$ ,  $|=$ ,  $\wedge=$  эдгээр нь дээрхтэй адил үүрэг гүйцэтгэх боловч утга оноодогоороо ялгаатай. Өөрөөр хэлбэл баруун тийш битээр шилжүүлж утга онооно, зүүн тийш битээр шилжүүлж утга онооно, бүхэл тоонд бит бүрээр AND хийж утга онооно, бүхэл тоонд бит бүрээр OR хийж утга онооно, бүхэл тоонд бит бүрээр XOR хийж утга онооно.

Жишээ: `int k=14;`

k: 0000 0000 0000 1110

`k>>=2;` үүний дараа

k: 0000 0000 0011 1000 буюу k=56 болно.

`k|=25;` үүний дараа

k: 0000 0000 0011 1000

25:0000 0000 0001 1001

k: 0000 0000 0011 1001 буюу k=57 болно.

## IV бүлэг

### "Си" хэлний операторууд

"Си" хэлний үндсэн операторуудад дараахь операторууд орно.

1. Хоосон оператор ; .
2. Нийлмэл оператор {} .
3. Оператор илэрхийлэл  $a=a+b$  .
4. Нөхцөлт оператор `if()` .
5. Алхамт ба параметртэй давталтын оператор `for()` .
6. Өмнө нөхцөл шалгах оператор `while()` .
7. Төгсгөлдөө нөхцөл шалгах оператор `do while()` .
8. Үргэлжлүүлэх оператор `continue` .
9. Сонголтын оператор `switch()` .
10. Зогсоогч буюу таслагч оператор `break` .
11. Удирдлага шилжүүлэгч оператор `goto` .
12. Дэд функцээс буцах оператор `return`.

#### 1. Хоосон оператор

Бичигдэх хэлбэр: `<;>`

Үйлдэл хийх шаардлагагүй тохиолдолд хоосон операторыг хэрэглэнэ. Хоосон операторыг хаана хэрэгтэй байна тэр газар, өөрөөр хэлбэл программын дурын хэсэгт хэрэглэж болно. Хоосон операторыг операторын төгсгөл гэж ойлгож болно. Энэ нь ихэнхдээ `do while`, `for`, `if`, `while` операторуудад ашиглагдана.

*Жишээ программ: Prog4\_1.cpp*

```
/*A,B хоёр тооны A тоо их бол дэлгэцэнд max=A гэж хэвлэ.*/
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
{
int A,B;
clrscr();
```



## Программчлалын “Си” хэл

---

```
printf("A="); scanf("%i",&A);      // A хувьсагчид утга олгох
printf("B="); scanf("%i",&B);      // B хувьсагчид утга олгох
if (B>A) ; // Хэрэв B тоо их бол ямар ч үйлдэл хийгдэхгүй. Хоосон оператор.
else      // Хэрэв A тоо их бол дэлгэцэнд max=A гэж хэвлэнэ.
    printf("\nmax=A");
getch();
}
```

Дэлгэцэнд:

```
A=      // Гараас дурын тоо өгнө. Жишээ нь: 45
B=      // Гараас дурын тоо өгнө. Жишээ нь: 14
max=A
```

### 2. Нийлмэл оператор

Бичигдэх хэлбэр: <{}>

Үүнийг блок буюу их хаалт гэж нэрлэж болно.

Нийлмэл операторыг нэг цогц оператор гэж ойлгож болно.

Нийлмэл операторт гаднаас шилжилт хийж болно (goto). Давталт болон нөхцөл шалгах операторыг хэрэглэж байх үед хоёр ба түүнээс дээш үйлдлүүд заавал хийгдэх тохиолдолд блок буюу их хаалтыг (нийлмэл операторыг) хэрэглэнэ. Харин давталт болон нөхцөл шалгах операторыг хэрэглэж байх үед ганцхан үйлдэл хийгдэх бол их хаалтыг хэрэглэх шаардлагагүй.

Мөн функцын эх биеийг блок гэж ойлгож болно.

```
Жишээ: main()
    {      // main() функцын эхлэл
    }      // main() функцын төгсгөл
```

Нийлмэл оператор нь тухайн программаас хамаараад хэдэн ч удаа хэрэглэгдэж болно.

*Жишээ программ: Prog4\_2.cpp*

```
/*1-ээс N хүртэлх тооны нийлбэрийг ол.*/
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
```

## Программчлалын “Си” хэл

---

```
{ //Нийлмэл операторын эхлэл1
int N,i,sum;
clrscr();
sum=0; // Нийлбэрийг хадгалах sum хувьсагчийн утга 0 байх ёстой.
printf("Input N=");
scanf("%i",&N); // Хэд хүртэлх тооны нийлбэр олохоо тодорхойлох хэрэгтэй.
i=1; // Давталт хийх эхний утга.
while(i<N)
    { printf("%i",sum); //Нийлмэл операторын эхлэл2
      sum=sum+i; // Нийлбэрийг тус бүрд нь олж хэвлэх.
      printf("+%i=%i\n",i,sum);
      i++; // i хувьсагчийн утгыг N хүртэл нэг нэгээр нэмэгдүүлэх.
    } //Нийлмэл операторын төгсгөл2
printf("All sum=%i",sum); // Нийт олдсон нийлбэрийг хэвлэх.
getch();
} //Нийлмэл операторын төгсгөл1
```

Дэлгэцэнд:

```
Input N= // Гараас N хувьсагчид дурын тоо өгнө. Жишээ нь: 5
0+1=1 // 1 үед нийлбэрийг олж хэвлэсэн.
1+2=3 // 2 үед нийлбэрийг олж хэвлэсэн.
3+3=6 // 3 үед нийлбэрийг олж хэвлэсэн.
6+4=10 // 4 үед нийлбэрийг олж хэвлэсэн.
All sum=10 // Нийт нийлбэрийг олж хэвлэсэн.
```

### 3. Оператор илэрхийлэл

Бичигдэх хэлбэр: <илэрхийлэл>;

Оператор хэрэгтэй газар бүр илэрхийллийг бичиж болно.

Жишээ: a=a+b; j++; гэх мэт.

*Жишээ программ: Prog4\_3.cpp*

```
/*Хоёр тооны ихээс нь багыг нь хасах программ бич.*/
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
```

## Программчлалын “Си” хэл

---

```
main()
{
int count1,count2;
clrscr();
printf("Input count1="); scanf("%i",&count1);
printf("Input count2="); scanf("%i",&count2);
if (count1>count2)           // 2 тооны ихийг олж байгаа хэсэг.
    printf("Sub=%i-%i=%i",count1,count2,count1-count2);
else
    printf("Sub=%i-%i=%i",count2,count1,count2-count1);
getch();
}
```

Дэлгэцэнд:

```
Input count1=           // Гараас дурын тоо өгнө. Жишээ нь:125
Input count2=           // Гараас дурын тоо өгнө. Жишээ нь: 45
Sub=125-45=80           // Их тооноос бага тоог нь хасаад хэвлэх.
```

### 4. Нөхцөлт оператор

```
Бичигдэх хэлбэр: if(<нөхцөл1>)
                    <оператор1>;
                    [else
                    <оператор2>];
```

<нөхцөл1>- нь нөхцөлт илэрхийлэл байж болно.

<нөхцөл1> нөхцөл биелж байвал <оператор1> гүйцэтгэгдэнэ. Эсрэг тохиолдолд буюу <нөхцөл1> биелээгүй үед <оператор2>-г гүйцэтгэнэ.

Бичигдэх хэлбэрийн дөрвөлжин хаалт ( [] ) дотор бичигдсэн бичлэгүүд нь заавал бичигдэх албагүй. Өөрөөр хэлбэл хэрэглэгдэж болно эсвэл хэрэглэгдэхгүй орхигдож болно гэсэн үг юм. Энэ нь тухайн программаас шалтгаална.

Иймд: [else <оператор2>]; нь бичигдээгүй үед

```
Бичигдэх хэлбэр: if(<нөхцөл1>)
                    <оператор1>; ийм болно.
```

Энэ нь <нөхцөл1> нөхцөл биелж байвал <оператор1> гүйцэтгэгдэнэ. Эсрэг тохиолдолд юу ч хийгдэхгүй. (Программын дараагийн мөрөнд шилжинэ.)

# Программчлалын “Си” хэл

Жишээ программ: Prog4\_4a.cpp

```
/*a,b хоёр тооны ихийг нь хэвлэх программ*/
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
{
int a,b;
clrscr();
/*Гараас a, b хувьсагчуудад бүхэл тоон утга авах*/
printf("Input a="); scanf("%i",&a);
printf("Input b="); scanf("%i",&b);
if(a>b) // a тоо b тооноос их бол
printf("max value=%i",a);
else // эсрэг тохиолдол буюу b тоо a тооноос их бол
printf("max value=%i",b);
getch();
}
```

Дэлгэцэнд:

```
Input a= // Гараас дурын тоо өгнө. Жишээ нь: 10
Input b= // Гараас дурын тоо өгнө. Жишээ нь: 8
max value=10
```

Нөхцөлт оператор нь давхарласан байдлаар орсон байж болно.

```
if(<нөхцөл1>
if(<нөхцөл2>
<оператор1>;
[else
<оператор2>;]
[else
<оператор3>;]
```

<нөхцөл1> биелж байвал <нөхцөл2>-г шалгана. <нөхцөл2> биелж байвал <оператор1>-г гүйцэтгэнэ. Эсрэг тохиолдолд (<нөхцөл2> биелээгүй бол) <оператор2>-г гүйцэтгэнэ <нөхцөл1> биелээгүй бол шууд <оператор3>-г гүйцэтгэнэ.

[] хаалтанд байгаа учраас

```
[else
```

## Программчлалын “Си” хэл

---

```
<оператор2>;] бичигдээгүй гэж үзвэл дээрх бичлэг
if(<нөхцөл1>
    if(<нөхцөл2>)
        <оператор1>;
    else
        <оператор2>; ийм хэлбэртэй болно.
```

<нөхцөл1> биелж байвал <нөхцөл2>-г шалгана. <нөхцөл2> биелж байвал <оператор1>-г гүйцэтгэнэ. Эсрэг тохиолдолд (<нөхцөл2> биелээгүй бол) <оператор2>-г гүйцэтгэнэ.

```
Мөн           [else
                <оператор2>;]
                [else
                <оператор3>;]
```

хоёулаа бичигдээгүй гэж үзвэл дээрх бичлэг нь дараахь хэлбэртэй болно.

```
if(<нөхцөл1>
    if(<нөхцөл2>)
        <оператор1>;
```

<нөхцөл1> биелж байвал <нөхцөл2>-г шалгана. <нөхцөл2> биелж байвал <оператор1>-г гүйцэтгэнэ. Өөрөөр хэлбэл <нөхцөл1>, <нөхцөл2> зэрэг биелсэн үед <оператор1> гүйцэтгэгдэнэ.

### *Жишээ программ: Prog4\_4b.cpp*

```
/*a,b,c гурван тооны ихийг нь олох программ бич*/
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
{
    int a,b,c;
    clrscr();
    printf("Input a="); scanf("%i",&a);
    printf("Input b="); scanf("%i",&b);
    c=30;                // с хувьсагчийн утгыг тодорхойлж байна.
    printf("Input c=%i\n",c);
    if(a>b)              // а тоо b тооноос их
        if(a>c)         // а тоо b, c тоонуудаас их
```

## Программчлалын “Си” хэл

---

```
        printf("max value=%i",a);
    else                // a тоо b тооноос их ба c тоо a тооноос их
        printf("max value=%i",c);
else                // b тоо a тооноос их
    if(b>c)          // b тоо a, c тоонуудаас их
        printf("max value=%i",b);
    else                // b тоо a тооноос их ба c тоо b тооноос их
        printf("max value=%i",c);
getch();
}
```

Дэлгэцэнд:

```
Input a=           // Гараас дурын тоо өгнө. Жишээ нь: 40
Input b=           // Гараас дурын тоо өгнө. Жишээ нь: 53
Input c=30
max value=53
```

Дээрх программыг (Prog4\_4b.cpp) өөрчилж бичье.

*Жишээ программ: Prog4\_4c.cpp*

```
/*a,b,c гурван тооны ихийг нь олох программ бич*/
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
{
int a,b,c;
clrscr();
/*Гараас тоонуудыг авсаны дараа их, багыг нь тодорхойлох хэрэгтэй.*
printf("Input a="); scanf("%i",&a);
printf("Input b="); scanf("%i",&b);
c=30; printf("Input c=%i\n",c);
if(a>b && a>c)                // a тоо b, c тоонуудаас их
    printf("max value=%i",a);
if(a<b && b>c )                // b тоо a, c тоонуудаас их
    printf("max value=%i",b);
if(c>b && a<c )                // c тоо b, a тоонуудаас их
    printf("max value=%i",c);
```

## Программчлалын “Си” хэл

---

```
getch();  
}
```

Дэлгэцэнд:

```
Input a=      // Гараас дурын тоо өгнө. Жишээ нь: 50  
Input b=      // Гараас дурын тоо өгнө. Жишээ нь: 3  
Input c=30  
max value=50
```

### 5. Алхамт ба параметртэй давталтын оператор

Бичигдэх хэлбэр: for([<эхний илэрхийлэл>;  
[<нөхцөлт илэрхийлэл>;  
[<илэрхийлэл буюу алхам>])  
<оператор>

<нөхцөлт илэрхийлэл> үнэн үед <оператор> биелнэ. <нөхцөлт илэрхийлэл> худал болтол <оператор> давтагдан хэрэгжинэ. <эхний илэрхийлэл> нь оператор доторх параметрын анхны утгыг тодорхойлох болно. <илэрхийлэл ба алхам> нь <эхний илэрхийлэл> -н утгыг нэмэгдүүлэх ба хорогдуулах үйлдлийг гүйцэтгэх замаар давталтаас гарах боломжийг олгодог байна.

for буюу алхамт ба параметртэй давталтын оператороос <нөхцөлт илэрхийлэл> үнэн үед ажиллагааг нь тасалж гарахдаа return, break, goto операторуудаар гарч болно.

Алхамт ба параметртэй давталтын операторыг тооцоолох дараалал байдаг.

1. <эхний илэрхийлэл> тооцоологдоно.
2. <нөхцөлт илэрхийлэл>-н утга үнэн бол <оператор> биелэгдэнэ. (Эсрэг тохиолдолд давталтаас гарна.)
3. <илэрхийлэл буюу алхам> тооцоологдоно.

Үүний дараа дээрх үйл ажиллагаа давтагдах ба энэ нь давталтаас гартал давтагдана.

Хэрэв нөхцөлт илэрхийлэл байхгүй бол энэ илэрхийллийг хэзээд үнэн гээд <оператор> биелж, <илэрхийлэл буюу алхам> тооцоологдоно. Энэ үед давталт хязгааргүй үргэлжилнэ. Иймд break, goto, return операторуудаар гарна. Доорх жишээг дискэн дээрээс нь дуудаж ажиллуулж үзэхэд энэ талаар илүү сайн ойлголт авах болно.

## Программчлалын “Си” хэл

---

Жишээ программ: Prog4\_5.cpp

```
/*Өгөгдсөн тэмдэгт мөрөнд хоосон зай орсон бол хоосон зайг хасаж бичих
программ бичье.*/
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
{
char str[100]="Bor land c "; // str хувьсагчид тэмдэгтэн мөрөн утгыг олгох
int i,j,space; // Давталт хийх i, j ба хоосон зайг тоолох хувьсагчууд зарлах.
clrscr();
printf("str=\"%s\"",str);
/*i, j, space хувьсагчуудын утгыг эхлээд 0 гэж өгнө. str[i]!='\0' нь мөрийн
төгсгөл хүрсэн эсэхийг шалгана. i++ нь нэг нэгээр нэмэгдсээр байгаад мөрийн
төгсгөл хүрэх ёстой.*/
for(i=j=space=0;str[i]!='\0';i++)
    { if(str[i]==' ') // Хоосон зай байвал
        space++; // space хувьсагч хоосон зайг тоолно.
      else // Хоосон зай биш байвал өөрөөр хэлбэл үсэг байвал
        {
          str[j]=str[i]; //Хоосон зайнаас бусдыг нь массивт хадгална..
          j++; // Хоосон зай биш бол массивын индекс нэгээр нэмэгдэнэ.
        }
    }
str[j]='\0'; // Мөрийн төгсгөл гэдгийг тодорхойлж өгнө.
printf("\nSpace count=%i\n**New form**\nstr=\"%s\"",space,str);
getch();
}
```

Дэлгэцэнд:

```
str="Bor land c " // Өгөгдсөн тэмдэгтэн мөр
Space count=4 // Хоосон зайг тоолсон тоолуур
**New form**
str="Borlandc" // Хоосон зайг хасаж үгийг шахсан байдал.
```



# Программчлалын “Си” хэл

## 6. Өмнө нөхцөл шалгах оператор

Бичигдэх хэлбэр: while(<нөхцөлт илэрхийлэл>)

<оператор>

<нөхцөлт илэрхийлэл>-н утга үнэн үед <оператор> хэрэгжих ба худал болох хүртэл давтагдан биелэгдэнэ. <илэрхийлэл> худал үед <оператор> биелэхгүй ба удирдлага нь түүний дараахь операторт шилжинэ. while операторын давталтыг түүний дотор бичигдсэн break, goto, return операторуудаар дуусгаж болно.

### Жишээ программ: Prog4\_6.cpp

```
/*Өгөгдсөн 10-тын тооллын системийн тоог 2-тын тооллын систем рүү
шилжүүлэх программ.*/
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
{
int N,ul,i=0;
clrscr();
printf("Decimal count N="); scanf("%i",&N); //10-тын системд гараас тоо өгөх
printf("***Binary count***");
while(N>0)
    { ul=N%2; // 2-тын тооллын систем дэх утгуудыг олох хэсэг
      gotoxy(12-i,4); // Курсорыг шилжүүлжүүлэн байрлуулагч функц
      printf("%i",ul);
      N=N/2; // Давталтаас гаргах үйлдэл
      i++; //Курсорын байрлалыг шилжүүлэхэд хэрэглэх хувьсагч
    }
getch();
}
```

Дэлгэцэнд:

```
Decimal count N=124 // 10-тын тооллын системд байгаа тооны ul-г
***Binary count*** // дэлгэцэнд утгуудыг хойноос урагш хэвлэж
    1111100 // 2-тын тооллын системд шилжүүлсэн байдал.
```

# Программчлалын “Си” хэл

---

## 7. Төгсгөлдөө нөхцөл шалгах оператор

Бичигдэх хэлбэр: do

<оператор>

while(<нөхцөлт илэрхийлэл>);

<оператор> гүйцэтгэгдсэний дараа <нөхцөлт илэрхийлэл>-н утгыг шалгана. <нөхцөлт илэрхийлэл> худал болох хүртэл энэ үйл ажиллагаа давтагдан биелэгдэнэ. break, goto, return операторуудаар давталтын ажиллагааг дуусгаж болно.

*Жишээ программ: Prog4\_7.cpp*

```
/*Esc товчлуур дарагдтал өгөгдсөн товчлууруудын ASCII кодыг хэвлэ.*/
```

```
#include<conio.h>
```

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
main()
```

```
{
```

```
int key; // Гараас товчлуурын утгыг авах хувьсагч.
```

```
clrscr();
```

```
do {printf("Input key=");
```

```
key=getch(); // Гараас товчлуур авах функц.
```

```
printf("%i\n",key);// Өгөгдсөн товчлуурын утгыг бүхэл тоогоор хэвлэх.
```

```
} while(key!=27); // Гараас өгсөн товчлуурыг ESC мөн эсэхийг шалгах
```

```
}
```

Дэлгэцэнд:

```
Input key= // Товчлуур дар. Жишээ нь: Enter=13
```

```
Input key=13
```

```
Input key=27
```

Гараас өгөгдсөн товчлууруудын ASCII кодууд хэвлэгдэнэ. Esc (27) товчлуур дарахад программ дуусах болно.

## 8. Үргэлжлүүлэх оператор

Бичигдэх хэлбэр: continue;

Энэ оператор нь do while, for, while давталтын операторуудтой цуг хэрэглэгдэнэ.

## Программчлалын “Си” хэл

---

Энэ оператороор давталтын операторууд нь дараагийн алхамуудад шилжинэ.

do while, while давталтын операторт continue операторыг ашиглах үед continue операторын дараагийн алхам нь нөхцөлт илэрхийллийг бодохоос эхэлнэ. Өөрөөр хэлбэл удирдлага while операторт шилжинэ гэсэн үг.

for операторт continue операторыг хэрэглэх үед continue операторын дараагийн алхам нь for операторын хувьсагчийн утгыг өөрчилж дараа нь давталт төгсгөлдөө хүрсэн эсэхийг шалгана.

### Жишээ программ: Prog4\_8.cpp

```
/* N тооны пактрелийг олох программ*/
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
main()
{
int i,N,рас=1;          // Пактрелийг олох хувьсагчийн утга 1-ээс эхлэх ёстой.
clrscr();
printf("INPUT N="); scanf("%i",&N); // Гараас N хувьсагчид тоо авна.
for(i=N;i>=0;i--)      // N-ээс 0 хүртэл буурах давталт хийнэ
{ if (i==1) continue; // i-н утга 1 бол рас-г 1-ээр үржүүлэх шаардлагагүй.
  if (i==0) continue; // i-н утга 0 бол рас-г 1-ээр үржүүлэх шаардлагагүй.
  рас=рас*i;
}
printf("N!=%i",рас);  // N! олсоноо хэвлэх хэсэг.
getch();
}
```

Дэлгэцэнд:

INPUT N= // Гараас дурын тоо өгнө. Жишээ нь: 5

N!=120 // Дэлгэцэнд хэвлэгдэх үр дүн

## 9. Сонголтын оператор

Бичигдэх хэлбэр: switch(<илэрхийлэл>)

{

[Зарлалтын хэсэг]

[case <тогтмол илэрхийлэл1>:]

## Программчлалын “Си” хэл

```
    <оператор1>:]
[case <тогтмол илэрхийлэл2>:]
    <оператор2>:]
.....
[default:
    <оператор>:]
}
```

switch сонголтын операторыг илэрхийллийн утгаас хамаарч боломжит хувилбарын аль нэгийг сонгоход хэрэглэнэ. Өөрөөр хэлбэл эхлээд <илэрхийлэл>-н утга тооцоологдоод түүний утгаас хамаарч аль нэг сонголбор ажиллуулахад хэрэглэнэ гэсэн үг юм. <тогтмол илэрхийлэл...>-үүдийг сонголборууд гэнэ.

<илэрхийлэл>-н утгаас хамаарч switch оператор сонгогдсон сонголборын <оператор...> ыг гүйцэтгэнэ. Үүний дараа switch блокын төгсгөл хүртэлх бүх сонголборын <оператор...>-г автоматаар гүйцэтгэж дуусаад гарна. Үүнийг таслахын тулд, өөрөөр хэлбэл тухайн сонголборыг гүйцэтгээд сонголтын оператороос гаргахын тулд break операторыг хэрэглэдэг.

<Илэрхийлэл>-н утга нь сонголборуудын алинтай ч тохироогүй үед default оператороор орж <оператор>-г гүйцэтгээд switch оператороос гарна.

<Зарлалтын хэсэг> байгаа тохиолдолд энэ нь хийгдэхгүйгээр тухайн сонгогдсон сонголборт шилжинэ. (Доорх жишээнүүдээс харагдана.)

Жишээ: 

```
char c='A';
switch(c)
{ case 'A': printf("A"); // Үүнийг гүйцэтгэсний дараа
  case 'B': printf("B"); // мөн энэ мөрийг гүйцэтгэнэ
  default: printf("Not A and B"); // мөн энэ мөрийг гүйцэтгэнэ.
}
```

гэж өгөгдсөн тохиолдолд дэлгэцэнд “ABNot A and B” гэсэн тэмдэгтэн мөр хэвлэгдэнэ.

```
char c='B';
switch(c)
{ case 'A': printf("A");
  case 'B': printf("B"); // Үүнийг гүйцэтгэсний дараа
  default: printf("Not A and B"); // мөн энэ мөрийг гүйцэтгэнэ.
}
```

гэж өгөгдсөн тохиолдолд дэлгэцэнд “BNot A and B” тэмдэгтэн мөр хэвлэгдэнэ.

## Программчлалын “Си” хэл

```
char c='C';
switch(c)
{ case 'A': printf("A");
  case 'B': printf("B");
  default: printf("Not A and B"); } // Үүнийг гүйцэтгэнэ.
```

гэж өгөгдсөн тохиолдолд дэлгэцэнд “Not A and B” гэсэн тэмдэгтэн мөр хэвлэгдэнэ.

Одоо break операторыг ашиглаж үзье.

Жишээ: 

```
char c='A';
switch(c)
{ case 'A': printf("A"); break;
  case 'B': printf("B"); break;
  default: printf("Not A and B"); } // Үүнийг гүйцэтгэсний дараа // энэ мөрийг гүйцэтгэхгүй // энэ мөрийг гүйцэтгэхгүй.
```

гэж өгөгдсөн тохиолдолд дэлгэцэнд “A” гэсэн тэмдэгтэн мөр хэвлэгдэнэ.

```
char c='B';
switch(c)
{ case 'A': printf("A"); break;
  case 'B': printf("B"); break;
  default: printf("Not A and B"); } // Үүнийг гүйцэтгэсний дараа // энэ мөрийг гүйцэтгэхгүй.
```

гэж өгөгдсөн тохиолдолд дэлгэцэнд “B” гэсэн тэмдэгтэн мөр хэвлэгдэнэ.

```
char c='C';
switch(c)
{ case 'A': printf("A"); break;
  case 'B': printf("B"); break;
  default: printf("Not A and B"); } // Үүнийг гүйцэтгээд switch // оператороос гарна. (break-г хэрэглэхгүй байж болно)
```

## Программчлалын “Си” хэл

---

гэж өгөгдсөн тохиолдолд дэлгэцэнд “Not A and B” гэсэн тэмдэгтэн мөр хэвлэгдэнэ.

*Жишээ программ: Prog4\_9.cpp*

```
/*countA, countB хоёр тоо өгөгдсөн. Энгийн илэрхийлэл бодох программ.*/
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
main()
{
int countA=12,countB=6,result;          // Хувьсагчуудыг зарлаж байна.
char operation_symbol;
clrscr();
printf("Input operation symbol=");
switch(operation_symbol=getch())
    { printf("It isn't in display.");      // Ажиллахгүй алгасах мөр
      printf("%c",operation_symbol);     // Ажиллахгүй алгасах мөр
    case '+':      // operation_symbol хувьсагчийн утга '+' тэмдэг байвал
        result=countA+countB;          // нэмэх үйлдэл хийнэ.
        break;      // switch оператороос гарах
    case '-':      // operation_symbol хувьсагчийн утга '-' тэмдэг байвал
        result=countA-countB;          // хасах үйлдэл хийнэ
        break;
    case '*':      // operation_symbol хувьсагчийн утга '*' тэмдэг байвал
        result=countA*countB;          // үржих үйлдэл хийнэ.
        break;
    case '/':      // operation_symbol хувьсагчийн утга '/' тэмдэг байвал
        result=countA/countB;          // хуваах үйлдэл хийнэ.
        break;
    default:      // operation_symbol нь дээрхийн алинтай нь ч ижил биш
        printf("\nYou must input operation symbol.");      // байвал бүх
        result=0;countA=0;countB=0;operation_symbol=' '; // хувьсагчийн
    }
    printf("\n%i %c %i= %i",countA,operation_symbol,countB,result);
    getch();
}
```

## Программчлалын “Си” хэл

---

Дэлгэцэнд:

```
Input operation symbol= // Гараас тэмдэгт өгөх. Жишээ нь: +
12 + 6= 18             // Гараас өгөгдсөн үйлдлийг хийж хэвлэнэ
```

### 10. Зогсоогч буюу таслагч оператор

Бичигдэх хэлбэр: break;

Энэ оператор нь do while, for, while, switch операторуудын биелэлтийг таслан тэдгээрийн дараагийн операторт удирдлагыг шилжүүлнэ. break нь эдгээр операторуудын гадна бичигдвэл компилятор түүнийг алдаанд тооцно.

#### Жишээ программ: Prog4\_10.cpp

```
/* Өгөгдсөн өгүүлбэрийн эхний үгийг хэвлэх программ.*/
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
{
char *var_string;
clrscr();
printf("Input string=");
gets(var_string);           // Гараас тэмдэгтэн мөр авдаг функц.
printf(" *First word* =");
while(*var_string!='\0')    // Тэмдэгтэн мөрийн төгсгөл хүртэл давтана.
{ printf("%c",*var_string); // Эхний үгийг хэвлэж байгаа хэсэг.
  if (*var_string==' '||*var_string==',') // Үг нь хоосон зай, таслалаар
    break;                               // тусгаарлагддаг учир давталтаас гарна.
  var_string++;                          // Дараагийн тэмдэгтийн хаягийг авна.
}
getch();
}
```

Дэлгэцэнд:

```
Input string= // Гараас тэмдэгтэн мөр оруулах. “This is my book.”
*First word* =This // Эхний үг.
```

# Программчлалын “Си” хэл

## 11. Удирдлага шилжүүлэгч оператор

Бичигдэх хэлбэр: goto <тэмдэгт>;

```
.....  
<тэмдэгт>: <оператор>
```

goto оператор нь <тэмдэгт>-ээр тэмдэглэгдсэн операторт удирдлагыг шилжүүлнэ. Тэмдэглэгдсэн <оператор> байхгүй эсвэл ийм тэмдэгтэй оператор олон байвал компилятор түүнийг алдаанд тооцно. Операторын тэмдэг нь зөвхөн goto оператор байгаа үед хүчинтэй.

goto оператор байхгүй, тэмдэгт оператор байгаа бол тэмдэгт оператор нь тооцоологдохгүй ба программын дэс дараалал хэвийн үргэлжилнэ.

### Жишээ программ: Prog4\_11.cpp

```
/* Компьютерийн санасан тоог таах программ. */  
#include<conio.h>  
#include<stdio.h>  
#include<stdlib.h>  
main()  
{  
int mem_count,my_count;           // Гараас өгөх тоог хадгалах хувьсагч.  
clrscr(); randomize();             // Санамсаргүй утгыг давтахгүй байх.  
mem_count=random(10);              // Санамсаргүй тоог гаргана.  
rep: clrscr();                      // Удирдлага шилжих мөр  
printf("Input your count (0-9) ="); scanf("%i",&my_count);  
if (mem_count!=my_count)           // Санасан тоог таагаагүй бол  
    goto rep;                       // удирдлага rep-д шилжинэ.  
printf("It is OK.");               // Таасан үед дэлгэцэнд хэвлэх.  
getch();  
}
```

Дэлгэцэнд:

Input your count (0-9) = // Гараас 0-9 хооронд тоо өгнө.

It is OK. // Таасан тохиолдолд программ дуусна.

## 12. Дэд функцээс буцах оператор

Бичигдэх хэлбэр: return[<илэрхийлэл>]



## Программчлалын “Си” хэл

---

Энэ оператор нь функц утга буцаахад хэрэглэгдэнэ. Функц дуудагдаад ажиллаж дуусны дараа утга буцааж, дуудсан хэсгийн дараагийн операторт удирдлагыг шилжүүлнэ. Хэрэв <илэрхийлэл> өгөгдсөн байвал түүнийг бодож функцын тодорхойлсон төрөл бүхий утгыг буцаана. <илэрхийлэл> байхгүй бол буцах утга байхгүй.

### *Жишээ программ: Prog4\_12.cpp*

```
/*N хүртэлх сондгой тоонуудын үржвэрийг олох программ бич*/
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
long Multiply(int N); // Дэд функцыг зарлаж байгаа хэсэг.
main()
{
long Mul; // Функцээс буцах утгыг авах Mul хувьсагч.
int N=20; // Хэд хүртэл давтагдахыг тодорхойлж байна.
clrscr();
printf("N=%i",N);
Mul=Multiply(N); // N параметртэй дэд функцыг дуудаж байна.
printf("\nMULTIPLY=%li",Mul);
getch();
}
/*Үржвэр олдог дэд функцыг тодорхойлж байгаа хэсэг.*/
long Multiply(int N)
{int i; long s=1;
for(i=1;i<N;i+=2) // i хувьсагч 2 алхамтайгаар нэмэгдэх тул утга нь 1,3,5 г.м
s=s*i; // Үржвэр олж байгаа хэсэг.
return s; // Нийт олсон үржвэрийг үндсэн функцэд буцаах.
}
```

Дэлгэцэнд:

N=20

MULTIPLY=654729075 // 1-ээс 20 хүртэлх сондгой тооны үржвэр.

# Программчлалын “Си” хэл

## V бүлэг Динамик ой

### Массив

Нэг ижил төрөлтэй элементүүдийг индекслэн нэг цогц болгохыг массив гэнэ. Массивын зарлалуудыг доор үзүүлье.

Бичигдэх хэлбэр:

<төрөл> <хувьсагчийн нэр>[<индекс>;

<төрөл> <хувьсагчийн нэр>[<индекс1>][<индекс2>;

<төрөл>- Массивыг төлөөлөх хувьсагчийн төрөл буюу өөрөөр хэлбэл массивын элементүүдийн төрөл байна.

<хувьсагчийн нэр>- Массивын нэр байна.

<индекс>- Массивт байж болох элементийн тоо. (1 хэмжээст)

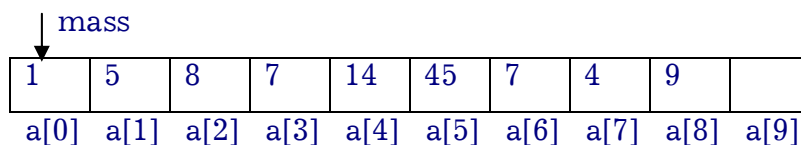
<индекс1>, <индекс2>- 2 хэмжээст массивын мөр ба баганын дугаар болно.

Жишээ: `int mass[10];` // 1 хэмжээст `mass` нэртэй бүхэл тоон массив.

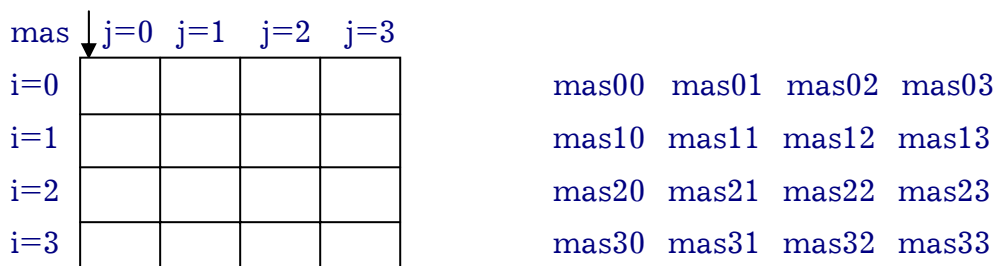
`float mas[4][4];` // 2 хэмжээст `mas` нэртэй бодит тоон массив.

Эхний жишээний хувьд `mass` массив нь бүхэл тоон 10 элемент агуулж болно.

Зардагдсан элементийн тооноос илүү олон тооны элемент хэрэглэж болохгүй. Харин дутуу тооны элемент хэрэглэж болно. Өөрөөр хэлбэл, 10 элемент агуулсан гэж зарлаад 11 элемент авах үйлдэл хийж болохгүй ба 9 элемент авч болно гэсэн үг юм. Энэ тохиолдолд үлдсэн 1 зай нь хэрэглэгдэхгүй үлдэнэ. Массивын индекс нь 0 эхэлж дугаарлагддаг.



2 дахь жишээний хувьд 4,4 хэмжээстэй бодит тоон `mas` нэртэй массив байна гэж ойлгож болно.



## Программчлалын “Си” хэл

Массивын элементүүдийн анхны утгыг дараахь байдлаар олгож болно.

```
Жишээ: int i_mass[]={1,2,3,4,5};           // Массивын хэмжээг 5 гэж авна.
        double d_mass2[4]={2.1,4.5,8.3,1.6}; // Массивын хэмжээ 4 байна.
        float f_mass[3][3]={{1.1,1.2,1.3},{2.1,2.2,2.3},{3.1,3.2,3.3}};
        int mass1[][4]={{1,2,3,4},{5,6,7,8},{1,2,3,4}}; // Массивын хэмжээ 3,4 юм.
mass1 нь дараахь байдалтай болно.
```

```
mas00=1    mas01=2    mas02=3    mas03=4
mas10=5    mas11=6    mas12=7    mas13=8
mas20=1    mas21=2    mas22=3    mas23=4
```

Массивын хэмжээг тодорхойлж өгөхгүйгээр утга олгосон үед хэдэн утга оноосон байна түүний тоогоор массивын элементийн хэмжээг зарласан гэж үзнэ. Үүнийг дээр өгсөн жишээнээс харж болно.

### Жишээ программ: Prog5\_1.cpp

```
/*Массивт санамсаргүйгээр 0-5 хүртэлх утгуудыг оруулж хэвлэх программ.*/
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
{
int i,j,N=4,i_mas1[10],i_mas2[10][10]; // Массив ба хувьсагч зарлах хэсэг.
clrscr();
printf("i_mas2\t   i_mas1\n\n");
for(i=0;i<N;i++)           // 2 хэмжээст массивын хувьд мөрийн индекс болно.
    { for(j=0;j<N;j++) // 2 хэмжээст массивын хувьд баганын индекс болно
        { i_mas2[i][j]=random(5); // 2 хэмжээст массивт санамсаргүй утга
          printf("%i ",i_mas2[i][j]); // авч хэвлэх хэсэг
        }
        i_mas1[i]=random(5); // 1 хэмжээст массивт санамсаргүй утга авна.
        printf("\t%i\n",i_mas1[i]); // i_mas1 массивыг хэвлэж байгаа хэсэг.
    }
getch();
}
```

Дэлгэцэнд:

## Программчлалын “Си” хэл

```
i_mas2      i_mas1      // i_mas2 массив нь 2 хэмжээст массив.
0 0 1 0      1          // i_mas1 массив нь 1 хэмжээст массив.
1 2 0 1      4          // Дэлгэцэнд массивуудын утгыг
3 1 0 4      3          // санамсаргүйгээр авч хэвлэж үзүүлэв.
4 0 1 2      0
```

Массивыг зарлах үед түүнийг санах ойн тодорхой заагдсан зайд байрлуулдаггүй ба динамик санах ойн хэрэглэгдээгүй чөлөөтэй хэсгийн хаягийг нэрээр нь заалгана. Иймд массивын нэр нь массивын эхний элементийг зааж байгаа хаяг байдаг. Өөрөөр хэлбэл 1 хэмжээст массивын нэр нь массивын 0-р элементийн заагч, 2 хэмжээст массивын нэр нь [0][0]-р элементийн заагч болно.

Тэмдэгтэн массив зарлах нь тоон тогтмолон массив зарласантай адил байна.

```
Жишээ: char a_char[]={ 'a','b','c','d'};
        char a_char1[4][3]={{ 'b','i','g'},{'b','a','d'},{'b','u','t'},{'b','e','g'}};
        char a_char2[4][3]={"abc","def","abc","asd"}; гэх мэт.
```

Тэмдэгтэн массивыг тэмдэгтэн мөр гэж ойлгож болно. Тэгвэл a\_char массив “abcd” гэсэн тэмдэгтэн мөр болно. Харин a\_char1 массив маань “big”, “bad”, “but”, “beg” гэсэн үгнүүдийг агуулж байна. Тэмдэгтэн массивын хойно тэмдэгтэн мөрийн төгсгөлийн тэмдэг ‘\0’ автоматаар тавигдана.

### Жишээ программ: Prog5\_2.cpp

```
/*Гараас өгөгдсөн үг өгүүлбэрт байгаа эсэхийг шалгах программ.*/
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
{
char input_word[20]; // 20 тэмдэгт агуулах тэмдэгтэн массив зарлах хэсэг.
char old_sentence[100]; // 100 тэмдэгт агуулж болох тэмдэгтэн массив зарлах
int cmp_symbol,j=0,i; // Массивуудын индексүүдийг зарласан байдал.
clrscr();
printf("Input your sentence."); gets(old_sentence); // Гараас өгөх өгүүлбэр.
printf("Input word="); gets(input_word); // Гараас өгөгдөх үг.
i=0, cmp_symbol=0;
while(old_sentence[i]!='\0') // Өгөгдсөн өгүүлбэрийн төгсгөл мөн эсэх.
```

## Программчлалын “Си” хэл

```
/*Өгөгдсөн өгүүлбэрийн үг бүрийг таслан авч байгаа хэсэг*/
if( (old_sentence[i]==' ' || old_sentence[i]=='.' || old_sentence[i]==' ' ||
    old_sentence[i+1]=='\0' || input_word[j]==' ' || input_word[i]=='.' ||
    input_word[i]==',' )
    { if (!cmp_symbol&&j!=0) printf("This sentence has <%s>\n",input_word);
      /*Өгөгдсөн үг байвал хэвлээд тоолуур болон индексг 0 болгно.*/
      cmp_symbol=0,j=-1;
    }
else /* Үсгүүд нь ялгаатай бол cmp_symbol хувьсагчийн утга нэмэгдэнэ.*/
    if(old_sentence[i]!=input_word[j]) cmp_symbol++;
    i++; j++; // Индексүүдийг нэмэгдүүлнэ.
}
getch();
}
```

Дэлгэцэнд:

Input your sentence. // Гараас өгүүлбэрээ оруул. Жишээ:” This is.”

Input word= // Гараас үгээ оруул. Жишээ нь: is

This sentence has <is>.

### Заагч ба хаяг

Заагч гэдэг бол объектыг заах хувьсагч. Заагч төрлийн хувьсагчийг зарлахдаа өмнө нь (\*) од тавина. Жишээ нь: Int \*a, \*I; гэх мэт.

&a-Үүний тусламжтай операндын хаяг өгөгдөнө.

(\*)- Үүний тусламжтайгаар тухайн хаяг дээрх өгөгдлийг авч болно. Энэ үед операнд нь заагч төрөлтэй байх ёстой.

Жишээ: int y; int \*p=8;

Энэ үед y=\*p утга оноох үйлдэл нь y хувьсагчид p заагчийн зааж байгаа санах ойн үүрийн утгыг авна. Үйлдлийн үр дүн нь операндын зааж байгаа утга болно. Заагчийн төрөл нь тэр заагчаар заалгаж байгаа төрөл болно.

“&” тэмдгээр хувьсагчийн хаягийг авна. Тэгвэл p=&y үйлдэл нь y-ийн хаягийг p-д олгоно. Харин &(y+1) эсвэл &3 үйлдлүүд байж болохгүй. Мөн 2 заагчийг нэмэх, үржүүлэх, хуваах зэрэг үйлдлүүд хийгддэггүй. Заагч нь нэгээр нэмэгдүүлэх (++), нэгээр хорогдуулах (--), нэмэх (+), хасах (-) үйлдэлд ашиглагдана. Заагч нь нэгээр нэмэгдэж байгаа тохиолдолд заагч нь

## Программчлалын “Си” хэл

объектын хэмжээгээр нэмэгддэг , нэгээр хорогдож байгаа тохиолдолд заагч нь объектын хэмжээгээр хорогддог.

Жишээ:     `int *count_point=4;`  
              `count_point++;`  
              `*count_point=19;`

The diagram illustrates pointer arithmetic. It shows a variable named 'count' with an arrow pointing to a box containing the number '4'. Below it, a pointer variable '\*count\_point' has an arrow pointing to a box containing '19'. To the left of the '19' box, there is another box containing '4', representing the value stored at the address pointed to by '\*count\_point'.

*Жишээ программ: Prog5\_3cpp*

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
void main()
{
int *a; int x; // Хувьсагч а нь бүхэл тоон заагч, х нь бүхэл тоон хувьсагч.
clrscr();
*a=2; printf("%i ",*a);    // а заагчийн утгыг нь 2 болгох.
a++; *a=5; // а заагчийн хаягийг 1-р нэмэгдүүлж утгыг нь 5 болгох
printf("%i ",*(--a)); // а заагчийн хаягийг 1-р хорогдуулж, утгыг нь хэвлэх.
a++; *a=6; // а заагчийн хаягийг 1-р нэмэгдүүлээд, утгыг нь 6 болгох.
x=*a; printf("%i ",x);    // а заагчийн утгыг х хувьсагчид олгох.
getch();
}
Дэлгэцэнд:
2 2 6    // Дэлгэцэнд гарах үр дүн
```

Заагчийн төрөл нь тухайн заагчаар заалгаж байгаа төрөл болно. Нэг заагчийг тухайн заагчийн төрөлтэй ижил төрлийн өөр заагчид олгох боломжтой. Төрөл нь тодорхой бус төрөлт биш заагч гэж байна. Үүнийг зарлахдаа өмнө нь void түлхүүр үгийг хэрэглэнэ. Void төрлийн заагч нь дурын төрлийн заагчид ашиглагдах боломжтой.

Жишээ: `int x=45, *point_i;`  
          `double d_point=45.2;`  
          `void *v_point;`  
          `*point_i=123;`

## Программчлалын “Си” хэл

---

Дээр зарлагдсан заагч, хувьсагчуудын хувьд `v_point=&x;v_point=point_i;`  
`v_point=&d_point;` эдгээр үйлдэл хийгдэж болно. Харин `point_i=&d_point;`  
`point_i=v_point;` үйлдлүүд хийгдэж болохгүй. Учир нь төрөл нь өөр юм.

Жишээ программ: Prog5\_4cpp

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
main()
{
int *pa, x, a=4, *d;    // Заагч болон хувьсагчуудыг зарлах хэсэг.
clrscr();
pa=&a;                // а хувьсагчийн хаягийг pa заагчид олгож байна.
x=*pa;               // pa заагчийн зааж байгаа утгыг x хувьсагчид олгох.
if (x==*&x) printf("True");
else printf("False");
d=&x;
printf("\nAddress=%u\tAddress++=%u",d,d+1); // Заагчийн утгуудыг хэвлэх.
getch();
}
```

Дэлгэцэнд:

True

/\* int төрөл 2 байт учир d+1-н хаяг нь мөн 2 байтаар нэмэгдэнэ.\*/

Address=65524 Address+1=65526

### Массив ба заагч

Заагчийн операнд нь массивын идентификатор байж болно.

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
void main(){
    int a[]={1,3,3,4,35,6};
    int *p;
    int n=6,i;
    clrscr();
```

## Программчлалын “Си” хэл

---

```
p=a;
for(i=0;i<n;i++){
    printf("%i ",*p);
    p++;
}

getch();
}

#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
void main(){
    int a[3][3]={{1,3,3},{4,35,6},{1,2,3}};
    int *p;  int n=3,i;
    clrscr();
    p=a[0];
    for(i=0;i<n;i++)
        printf("%i ",*p++);
    getch();
}
```

### Заагчийн заагч

Заагчийн хаягийг заах үйлдлийг заагчийн заагч гэнэ. Үүнийг заагчид хамаарах массив ба мөр тэмдэгтийн хувьд хэрэглэхэд тохиромжтой. Доорхи жишээнүүдийг хар.

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
void main(){
    int a[3]={1,3,5};
    int *p[3]={a,a+1,a+2};
    int **pp=p;
    int n=3,i;
    clrscr();
```



## Программчлалын “Си” хэл

---

```
        for(i=0;i<n;i++){
            printf("%i ",**pp);
            pp++;
        }
        getch();
}

#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
void main(){
    char *a[]{"File","edit","search"};
    char **p[3]={a,a+1,a+2}; char ***pp=p;
    clrscr();
    printf("%c",***pp);
    pp=pp+1;
    **pp=**pp+2;
    printf("%c",***pp);
    pp=pp-1;
    **pp=**pp+2;
    printf("%c",***pp);
    pp=pp+2;
    **pp=**pp+1;
    printf("%c",***pp);
    getch();
}

#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
void main(){
    char *a[]{"File","edit","search"};
    char **p[3]={a,a+1,a+2}; char ***pp=p;
    clrscr();
    printf("%c%c%c%c%c",***pp,*(**(pp+1)+2),*(**p+2),*(**(pp+2)+1));
    getch();
}
```

# Программчлалын “Си” хэл

## VI бүлэг

**Нийлмэл төрлийн хувьсагч: тоочигдох төрөл,  
бүтэц төрөл, битэн талбар, нэгдэл**

### **Тоочигдох төрөл.**

Зарим тохиолдолд олон дахин өөрчлөгдөхгүй хувьсагчдад тоон утга оноон тоочих хэлбэрээр зарлаж болно. Тоочих төрлийг программд тодорхойлохдоо “enum” түлхүүр үгийг ашиглана.

Бичигдэх хэлбэр:

```
enum [<те_нэр>] {<тогт_нэр>[=<элемент>], ...} [хувьсагч_нэр];
```

Энэ зарлалтаар тоочигдох төрлийн хувьсагч тодорхойлогдож, түүний утгын муж тодорхой болно.

<те\_нэр>- Тоочигдох төрлийн нэр.

<тогт\_нэр>- <элемент>-үүдийг агуулж байгаа тогтмолын нэр юм.

<элемент>- Үүнд бүхэл тоо харгалзана. Элементүүд нь бүхэл тоон төрлийнх байна. Үүнээс үндэслээд тоочигдох төрлийг бүхэл тоон төрлийн нэг гэж үзэж болох ба түүн дээр бүхэл тоон төрөл дээр хийгдэх бүх үйлдэл хийгдэнэ.

Жич: Тоочигдох төрлийн <элемент>-үүдэд бүхэл тоон төрлүүд харгалзана.

<хувьсагч\_нэр>- Зарлагдсан тоочигдох төрлийг төлөөлж чадах хувьсагчийн нэр.

Жишээ: `enum day {January=1,February,March,April,May,June,July, August, September,October,November,December};`

Дээрх жишээний хувьд: `day` нь тоочигдох төрлийн нэр.

“January”, “February”, “March”, “April” гэх мэт тогтмолын нэр.

`January=1, February=2` буюу `January+1, March=3` буюу `February+1, April=4` гэсэн утгуудыг авна.

### *Жишээ программ: Prog6\_1.cpp*

```
/*Товчлууруудын ASCII кодуудыг тоочигдох төрөл ашиглан тодорхойлох*/  
#include<conio.h>  
#include<stdio.h>  
#include<stdlib.h>  
enum keys {Esc=27,Enter=13,F1=59,F2,F3,F4  
           } hit_key; // Тоочигдох төрөл зарлах хэсэг
```

## Программчлалын “Си” хэл

---

```
main()
{
enum keys key;    // Тоочигдох төрлийн key хувьсагчийг зарласан байдал.
key=Enter;       // key хувьсагч нь Enter тогтмолын утгыг авна.
hit_key=F4;      // Тоочигдох төрлийн hit_key хувьсагч F4-н утгыг авна.
clrscr();
printf("key=%i\thit_key=%i",key,hit_key);
getch();
}
```

Дэлгэцэнд:

```
key=13   hit_key=62 // Тоочигдох төрлийн хувьсагчуудын утгууд.
```

### Бүтэц төрөл.

Бүтцийн тусламжтайгаар нэг объектод хэд хэдэн төрлийг хамааруулж болно. Өөрөөр хэлбэл янз бүрийн төрөлтэй өгөгдлүүдийг хамтатган тодорхойлж хэрэглэгчид шинэ өгөгдлийн төрлийг бий болгож өгнө. “Си” хэлэнд бүтэц дээр цөөхөн төрлийн үйлдэл хийгддэг. Үүнд: функцийн аргумент хэлбэрээр дамжуулах, функцийн утга хэлбэрээр хүлээн авах, мөн хаяг тодорхойлох. Үүнээс гадна бүтэц дээр утга оноож болно.

Бичигдэх хэлбэр: struct [<бүтцийн нэр>]

```
{ [<төрөл> <хувьсагчийн нэр[>, <хувьсагчийн нэр1>...>];
  [<төрөл> <хувьсагчийн нэр[>, <хувьсагчийн нэр1>...>];
  .....
} [<бүтэц_хувьсагч>];
```

<бүтцийн нэр>- Энэ нь тухайн бүтцийг төлөөлж чадах бүтцийн ерөнхий нэр.

Бүтэц доторх элементүүд нь өөрийн төрөл ба хувьсагчуудтай байна.

Жишээ: struct book

```
{ char *name, writer[20];
  int page_number;
}
```

<бүтэц\_хувьсагч> Энэ нь бүтэц төрлийн хувьсагч эсвэл бүтэц төрлийн заагч байж болно.

Жишээ: struct book

```
{ char *name, writer[20];
  int page_number;
```

## Программчлалын “Си” хэл

---

```
} var_book;
```

Программын явцад book төрлийн хувьсагч, заагч хэрэгтэй боллоо гэвэл

Жишээ: struct book \*point\_book, new\_book; гэж бичиж өгнө.

Дээрх хоёр хувьсагчууд (<бүтцийн нэр>, <бүтэц\_хувьсагч>) нь [] хаалтанд байгаа учраас сайн дурын буюу өөрөөр хэлбэл бичигдэж болно эсвэл бичигдэхгүй орхигдож болох хувьсагчууд. Гэвч эдгээр хувьсагчуудын аль нэг нь заавал зарлагдах ёстой. 2-уулаа зарлагдаагүй тохиолдолд тухайн бүтцийг дуудах боломжгүй болно. Бүтэц доторх элементэд хандахдаа “.”, “->” ашиглана 2 янзаар хандаж болно. Бүтэц төрлийн хувьсагч байх үед “.”, бүтэц төрлийн заагч байх үед “->” гэж бүтцийн элементэд хандана.

Жишээ: struct student

```
{ char name[20], *class, *address;
  int age, credit;
} anket, *new_anket;
```

.....

```
anket.name="Bat"; anket.age=20;
```

```
new_anket->name="Ganaa"; new_anket->age=22; гэж хандана.
```

Хэрэв бүтцийн тодорхойлолтыг функцийн дотор зарлавал энэ бүтэц нь зөвхөн тухайн функц дотор хүчин төгөлдөр ба үндсэн программд зарлагдсан байвал түүнээс хойш орших бүх функцэд хүчин төгөлдөр байна.

### *Жишээ программ: Prog6\_2.cpp*

```
#include<conio.h>
```

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
struct Student // Student бүтэц зарлаж байгаа хэсэг
{ char name[25]; // name талбар нь 20 тэмдэгт авна.
  int age; // age талбар нь бүхэл тоон утга авна.
};
```

```
main()
```

```
{
```

```
Student *Stud; // Stud гэсэн бүтцийг заагч хувьсагч зарлаж байна.
```

```
int n,i;
```

```
clrscr();
```

```
printf("Input n=");scanf("%i",&n); // Оюутны тоогоо гараас оруулна.
```

```
for(i=0;i<n;i++) // Оюутны тоогоор давталх хийж бичлэгээ оруулна.
```

## Программчлалын “Си” хэл

```
{ /* Бүтцэд нэр, насыг гараас оруулж өгч байна.*/
printf("%i-name: ",i+1); scanf("%s",(Stud+i)->name);
/* Stud+i нь бүтцийн заагчийн хаягийг i-ээр нэмэгдүүлж байна.*/
printf("%i-age: ",i+1); scanf("%i",&(Stud+i)->age);
}
printf("\nPrinted\n"); i=0;
while(i<n)
{ printf("%i-name: %s",i+1,Stud->name);
printf("\tage:\t%i\n",Stud->age);
i++; Stud++; // Бүтцийн заагчийн хаягийг нэгээр нэмэгдүүлж байна.
}
getch();
}
```

Дэлгэцэнд:

```
Input n= // Хэдэн оюутны бичлэг оруулах тоогоо өгнө. Жишээ 2.
1-name= // 1-р оюутны нэрийг өгнө. Жишээ нь: “Bat”.
1-age= // 1-р оюутны насыг өгнө. Жишээ нь: 20
2-name= // 2-р оюутны нэрийг өгнө. Жишээ нь: “Bold”.
2-age= // 1-р оюутны насыг өгнө. Жишээ нь: 25
***Printed structure*** // Бүтэц доторх утгуудаа хэвлэсэн байдал.
1-name: Bat age: 20
2-name: Bold age: 25
```

### Битэн талбар.

Битэн төрлийг ихэнхдээ 2 зорилгоор ашиглана.

1. Санах ойг хэмнэх зорилгоор. хувьсагчийг шахахад
2. Битүүд нь өөр функцийг агуулж байгаа гадаад төхөөрөмжүүдийн региструудэд хандах ажиллагааг хөнгөвчлөх зорилгоор.

Битээр тодорхойлогдсон бүтцийн элементийг битэн талбар гэнэ. Бүтэц төрөлтэй адилхан зарладаг боловч бүтцийн элемент нь 1-ээс 16 бит хүртэлх урттай битийг агуулна.

Бичигдэх хэлбэр: struct [<битэн талбарын нэр>]  
{[<төрөл> <хувьсагч>:<тогтмол илэрхийлэл>];  
[<төрөл> <хувьсагч>:<тогтмол илэрхийлэл>];

## Программчлалын “Си” хэл

---

```
}
```

Бичлэгийн <тогтмол илэрхийлэл1> -үүд нь сөрөг тоо байж болохгүй.

Жишээ: struct

```
{ unsigned icon:8;
  unsigned color:4;
  unsigned underline:1;
  unsigned blink:1;
} screen[25][80];
```

Энэ жишээ нь дэлгэцэн дээрх тэмдэгтийн атрибутын тодорхойлолт юм.

### *Жишээ программ: Prog6\_3.cpp*

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
struct bit_field
{
    // Битэн талбар зарлаж байгаа хэсэг.
    int bit_1      : 1; // Эхний 1 тэмдэгтийг агуулах хувьсагч.
    int bits_2_to_5 : 4; // Дараагийн 4 тэмдэгтийг агуулах хувьсагч.
    int bit_6      : 1; // Дараагийн 1 тэмдэгтийг агуулах хувьсагч.
    int bits_7_to_16 : 10; // Дараагийн 10 тэмдэгтийг агуулах хувьсагч.
} bit_var;
// Бит талбарын хувьсагч.

main()
{
    clrscr();
    bit_var.bit_1=1;           // 0000 0000 0000 0001
    bit_var.bits_2_to_5=10;    // 0000 0000 0001 0101
    bit_var.bit_6=1;          // 0000 0000 0010 0001
    bit_var.bits_7_to_16=1020; // 1111 1111 0000 0001
    printf("%u",bit_var);      // bit_var хувьсагчийн утга=1111 1111 0011 0101
    getch();
}

Дэлгэцэнд:
bit_var=65333 // bit_var-н утгыг 10-тын системд хэвлэсэн.
```

# Программчлалын “Си” хэл

## Нэгдэл.

Объектийг нэгдлийн тусламжтайгаар хугацааны янз бүрийн агшинд нэг объектод өөр өөр төрлийн утга хадгалах бололцоотой. Нэгдлийг зарлах явцад түүнд хадгалж болох төрлүүдийн холбоо бий болно. Хугацааны нэг агшинд нэгдэл нь тодорхойлогдсон төрлүүдийн зөвхөн нэгнийх нь утгыг авна. Нэгдэлд шаардлагатай санах ойн хэмжээ нь түүний элементүүдийн хамгийн уртынх нь санах ойд эзлэх хэмжээгээр тодорхойлогдоно. Иймд нэгдлийг санах ойг нөөцлөхөд хэрэглэж болно. Нэгдлийн элемент рүү хандах нь бүтэц төрөлтэй ижил.

```
Бичигдэх хэлбэр: union [<нэгдлийн нэр>]
                    { <төрөл> <хувьсагчийн нэр>;
                    .....
                    } <нэгдлийн хувьсагч>;
```

```
Жишээ: union sign
        { char ch;
          char * mvar[20];
        } number;   Санах ойд 20 тэмдэгтийн хэмжээгээр зай авна.
```

### Жишээ программ: Prog6\_4.cpp

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
union int_or_long { int    i;
                   long   l;
                   } a_number;

main()
{ int i=342; long l=4635;
  a_number.i=i; a_number.l=l;    // Нэгдэлд i, j хувьсагчийн утгыг өгнө.
  printf("i=%i",a_number.i);    // Нэгдэлийн i элементийн утгыг хэвлэх.
  printf("l=%ld ",a_number.l);  // Нэгдэлийн j элементийн утгыг хэвлэх.
  getch();
}
```

Дэлгэцэнд:

```
i=4635      l=4635    // Хамгийн сүүлд нэгдэлд 4635 утга орсон.
              // Иймд i, l хувьсагчуудын утга ижил.
```

## Программчлалын “Си” хэл

---

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
struct anket{      char  name[10];
                  int   age;
                  int   clss;
                  }students[5];
void main(){
    int i,n=5;
    clrscr();
    for(i=0;i<n;i++){
        printf("%i-student name=",i+1); scanf("%s",students[i].name);
        printf("%i-student age=",i+1); scanf("%i",&students[i].age);
        printf("%i-student class=",i+1); scanf("%i",&students[i].clss);
    }
    printf("\n");

    for(i=0;i<n;i++){
        printf("%s  ",students[i].name);
        printf("%i  ",students[i].age);
        printf("%i  \n",students[i].clss);
    }
    getch();
}
```



# Программчлалын “Си” хэл

## VII бүлэг

### Функц

Тодорхой үйлдлийг гүйцэтгэж тодорхой утгыг өгдөг программын хэсэг. "Си" программчлалын хэл нь дор хаяж нэг функцтэй байна. "Си" -ийн функцууд тэгш эрхтэй байдаг ба программ эхлэхдээ main() буюу үндсэн функцээс эхэлнэ. "Си" хэлэнд функц нь голлох үүрэгтэй бөгөөд функцыг хэрэглэгчийн функц, стандарт функц гэж хоёр ангилна. Стандарт функцууд "Си" хэлний толгой файлд (\*.h) хадаглагддаг. Хэрэглэгчийн функцыг программ зохиогч олон дахин давтагддаг үйлдлийг багасгах зорилгоор өөрөөр хэлбэл тодорхой нэг бодлогонд программын тухайн нэг хэсэг хэд дахин ашиглагдах шаардлагатай болдог ба шаардлагатай тэр модулийг функц болгон бичээд олон дахин дуудаж ашиглана. Хэрэглэгчийн функцыг зорилгоос нь хамааруулж аргументтай, аргументгүй, буцах утгатай, буцах утгагүйгээр бичиж болно. Функцыг зарлах үед түүний буцах утгыг тодорхойлно. Функцыг эхлээд зарлаад тодорхойлолтыг нь өөр газар зарлаж болно. Хэрэв функц нь бүхэл тоон утга өгдөг бол түүнийг бичихгүй байж болно.

Бичигдэх хэлбэр:

```
[<санах ойн ангилал>][<төрөл>]<функцын нэр>(идентификаторууд)
    {<функцын эх бие>}
```

<санах ойн модель>- Функцыг тодорхойлоход түүний санах ойн ангилалыг оруулж өгч болно. extern, static гэх мэт. Эсвэл санах ойн ангиллыг тодорхойлоогүй байж болно.

<төрөл> - Функцын өгөх утгын төрөл. Функц нь массив, функцээс өөр бүх төрлийн утга олгож болно. Функц нь бүх төрлийн заагчийг буцаана. Мөн массивын, функцын заагчийг буцаана. Функцээс буцааж буй утга нь функцын зарлагдсан төрөлтэй ижил байна. Хэрэв функц нь return командыг агуулаагүй эсвэл return командад ямар нэг утга бичээгүй бол функцын утга тодорхойгүй гэж үзнэ.

Функцэд дамжиж ирэх аргумент нь хэд хэд байж болох ба функц нь зөвхөн нэг төрлийн утгыг буцаана. Хэрэв функцийн өгөх төрлөөс өөр төрлийн утгыг буцаах болбол төрөл шилжүүлэх үйлдлийг ашиглана.

Жишээ нь:

```
#include<conio.h>
```

## Программчлалын “Си” хэл

---

```
#include<stdio.h>
#include<stdlib.h>
/*Өгөгдсөн координатад тэмдэгт хэвлэх хэрэглэгчийн функц*/
print(int x, int y,char ch) // print хэрэглэгчийн функц нь 3 аргументтай.
{gotoxy(x,y); // x,y аргументуудыг курсор шилжүүлэх координат болгох.
 printf("%c",ch); // ch аргументийг дэлгэцэнд хэвлэх
}
main()
{
char ch[5]="Hello";
int i,n=5;
clrscr();
for(i=0;i<n;i++) // “Hello” тэмдэгтэн мөрийг print() функцэд дамжуулах
print(40+i,12,ch[i]);
getch();
}
```

Аргументтай хэрэглэгчийн функцыг бичихдээ функцэд дамжигдаж ирж буй аргументууд нь функцээс буцахдаа утгаа өрчлөх, өөрчлөхгүй эсэхээс нь хамааруулж функцэд тодорхойлж өгөх ба функцийг дуудах нь өөр өөр байна. Функцэд массивыг дамжуулахдаа массивын нэрийг дамжуулахад хангалттай. Учир нь массивын нэр нь массивын эхний элементийн хаягийг зааж байдаг. Жишээ нь: Аргумент нь утгаа өөрчлөхгүй, int төрлийн утга буцаадаг функц.

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
int sum(int a[],int n);
main(){
int i,n,a[100]; clrscr();
printf("How many element=");scanf("%i",&n);
for(i=0;i<n;i++){
printf("a[%i]=",i);scanf("%i",&a[i]);
}
printf("\nSummary = %i",sum(a,n));
getch();
return 0;
}
```

## Программчлалын “Си” хэл

---

```
int sum(int a[],int n){
int i,s=0;
for(i=0;i<n;i++)
s+=a[i];
return s;
}
```

Жишээ нь: Аргумент нь утгаа өөрчилдөг буцах утгагүй функц.

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
sum(int a[],int n,int *s);
main(){
int i,n,a[100],s=0; clrscr();
printf("How many element=");scanf("%i",&n);
for(i=0;i<n;i++){
printf("a[%i]=",i);scanf("%i",&a[i]);
}
sum(a,n,&s);
printf("\nSummary = %i",s);
getch();
return 0;
}
sum(int a[],int n,int *s){
int i;
for(i=0;i<n;i++)
*s+=a[i];
}
```

Хэрэглэгчийн функц рүү мөн бүтцийг дамжуулж болно.

Жишээ нь:

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
int n;
struct Tsalin{
char name[10];
int info;
```

## Программчлалын “Си” хэл

---

```
};  
int Sum(Tsalin temp[]){  
int s=0,i;  
for(i=0;i<n;i++)  
s+=temp[i].info;  
return s;  
}  
main()  
{Tsalin names[100];  
int i; clrscr();  
printf("How many element="); scanf("%i",&n);  
for(i=0;i<n;i++){  
    printf("%i-name: ",i+1); scanf("%s",names[i].name);  
    printf("%i-info: ",i+1); scanf("%i",&names[i].info);  
}  
printf("\nSummary = %i", Sum(names));  
getch();  
}
```

# Программчлалын “Си” хэл

---

## VIII бүлэг

### Препроцессор

include удирдлагаар нэр бүхий файлын агуулгыг компиляци хийгдэж байгаа файлд оруулна. Препроцессор нь оруулж буй файлыг тухайн файльтай нэг ижил болгоно. Орж ирж буй файл нь дотроо препроцессорыг агуулж болно. Энэ нь сүүлд үндсэн файлдаа боловсруулалт хийнэ. include удирдлага нь давхарласан байж болно.

Хэрэв файлын нэр нь <> хаалтанд бичигдсэн байвал препроцессор нь тэрхүү файлыг эхлээд компиляторын удирдлагад тодорхойлсон замаар дараа нь стандарт тодорхойлсон замуудаар хайна.

Стандарт тодорхойлсон замууд гэдэг нь DOS-н PATH.

Компиляторын тодорхойлсон замууд гэдэг нь "Си" хэлний OPTIONS менүний DIRECTORIES дотор зарлагдсан замууд юм. <> хаалтанд зарлагдсан үед тухайн зам дээр байхгүй бол хэрэглэгдэж байгаа идэвхтэй каталог дотор байсан ч олохгүй.

Хэрэв файлын нэр нь “ ” дотор бичигдсэн байвал түүнийг идэвхтэй зам дээр, DOS-н PATH дээр эцэст нь компиляторын тодорхойлсон замууд болох OPTIONS менүний DIRECTORIES дотор зарлагдсан замуудаар үргэлжлүүлэн хайна.

Жишээ: #include<stdio.h>  
#include<My\_first\_file.cpp>  
#include”c:\appl\borlandc\bin\my\_file.c”  
#include”c:\appl\borlandc\bin\example\my\_header\_file.h” гэх мэт.

"Си" хэлэнд файлын нэрийг нэрлэгдсэн тогтмолоор орлуулах боломжтой.

Жишээ: #define My\_File\_Name “c:\appl\borlandc\bin\bodlogo.cpp”  
#include My\_File\_Name

### Нөхцөлт компиляци

Нөхцөлт компиляцид хэрэглэгдэх удирдлагууд:

#if            #elif            #else            #endif.

Бичигдэх хэлбэр: #if <илэрхийлэл>

                 <оператор1>

                 [#else <илэрхийлэл>] буюу [#elif <илэрхийлэл>]

## Программчлалын “Си” хэл

<оператор2>

#endif

Нөхцөлт компиляцийн удирдлагаар үндсэн файлын тодорхой хэсгүүдийг орхиж компиляци хийх боломжтой. Өгүүлбэр зүйн хувьд:

#if

----

#endif ийм байх ёстой. Программд <оператор> нь нэг буюу хэд хэдэн мөр байж болно. Препроцессор нь эдгээрээс аль нэгийг нь сонгож компиляци хийнэ. Сонгогдоогүй хэсэг нь компиляцид орохгүй.

Жишээ:

```
#if DLEVEL>5// DLEVEL хувьсагч өмнө зарлагдсан байх ёстой.
#define SIGNAL 1// SIGNAL хувьсагчид
  #if STACKUSE==1
    #define STACK 200
  #else
    #define STACK 100
  #endif
  #else
    #define SIGNAL 0
    #if STACKUSE==1
      #define STACK 100
    #else
      #define STACK 50
    #endif
  #endif
#endif
```

*Жишээ программ: Prog8\_1.cpp*

```
#include<conio.h>
```

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#define DEBIT 10 // DEBIT хувьсагч тогтмолоор зарлагдаж байна.
```

```
#define CREDIT 100 // CREDIT хувьсагч тогтмолоор зарлагдаж байна.
```

```
int debit() // int төрлийн debit() функц зарлагдаж байгаа хэсэг.
```

```
{ return 200; // Бүхэл тоон утга буцаана.
```

```
}
```

```
char credit() // char төрлийн credit() функц зарлагдаж байгаа хэсэг.
```

```
{ return 'c'; // Тэмдэг буцаана.
```

## Программчлалын “Си” хэл

---

```
}
char *print_word() // char * төрлийн print_word() функц зарлагдаж байгаа
хэсэг
{ return "batbat"; // Тэмдэгтэн мөр буцаана.
}
main()
{
clrscr();
#if defined(CREDIT) // CREDIT хувьсагч тогтмолоор зарлагдсан бол
printf("CREDIT value defined= %i",debit());
#elif defined(DEBIT) // DEBIT хувьсагч тогтмолоор зарлагдсан бол
printf("CREDIT value didn't define, DEBIT value defined= %c",credit());
#else // Дээрх 2 хувьсагч хоёулаа зарлагдаагүй бол
printf("DEBIT and CREDIT values didn't defined= %s",printerror());
#endif
getch();
}
```

Дэлгэцэнд:

CREDIT value defined= 200

# Программчлалын “Си” хэл

---

## IX бүлэг

Урсгал ба директор

Durectory-н удирдлагын функцууд

<DIR.H>

Эдгээр тэмдгүүд нь урсгалын тэмдэгтүүдийн хамгийн дугаарыг тоог тодорхойлдог.

Нэр	Гүйцэтгэх үүрэг
MAXPATH	Урсгалын зам (бүтэн)
MAXDRIVE	Дискны төхөөрөмж ("A:","C:" гэх мэт)
MAXDIR	Урсгалын subdirectory тодорхойлолт
MAXFILE	Өргөтгөлгүй урсгалын нэр
MAXEXT	Урсгалын өргөтгөл

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <dir.h>
```

```
char old_dir[MAXDIR];
```

```
char new_dir[MAXDIR];
```

```
int main(void)
```

```
{
```

```
    if (getcurdir(0, old_dir))
```

```
    {
```

```
        perror("getcurdir()");
```

```
        exit(1);
```

```
    }
```

```
    printf("Current directory is: \\%s\n", old_dir);
```

```
    if (chdir("\\\\"))
```

```
    {
```

```
        perror("chdir()");
```

```
        exit(1);
```

```
    }
```

Тодорхойлогдсон төхөөрөмжөөс идэвхтэй каталогийг авахдаа дараахь функцийг хэрэглэнэ.

Бичигдэх хэлбэр:



## Программчлалын “Си” хэл

---

```
getcurdir(<төхөөрөмж>, <каталоги>);
```

<drive>- Бүхэл тоон хувьсагч. 0 байвал default, 1 байвал A гэх мэт төхөөрөмжүүдийг заана.

<каталоги>- Тэмдэгтэн мөрөн хувьсагч байна. Идэвхтэй каталогийг авна. Энэ функц нь амжилттай хэрэгживэл 0, алдаатай бол 1 гэсэн бүхэл тоон утгуудыг буцаана

```
if (getcurdir(0, new_dir))
{
    perror("getcurdir()");
    exit(1);
}
printf("Current directory is now: \\%s\n", new_dir);
printf("\nChanging back to original directory: \\%s\n", old_dir);
```

Идэвхтэй каталогийг өөрчлөхдөө дараахь функцыг хэрэглэнэ.

Бичигдэх хэлбэр:

```
int chdir(<path>);
```

<path>- Тэмдэгтэн мөр байна. Өөрчлөх замыг бүтнээр нь ("c:\april\BC" гэх мэт.) өгнө. Энэ нь өгөгдсөн каталогийн зам нь идэвхтэй каталоги болно.

Амжилттай хэрэгживэл 0, алдаатай бол -1 гэсэн бүхэл тоон утгуудыг буцаана

```
if (chdir(old_dir))
{
/* Системийн алдааны мэдээллийг хэвлэдэг.*/
    perror("chdir()");
    exit(1);
}

return 0;
}
/*Доорхи жишээнд каталоги үүсгэдэг mkdir(), устгадаг rmdir() функцуудыг ашигласан.*/
#include <stdio.h>
#include <conio.h>
#include <process.h>
```

## Программчлалын “Си” хэл

---

```
#include <dir.h>

#define DIRNAME "testdir.$$$"
int main(void)
{
    int stat;
    stat = mkdir(DIRNAME);
    if (!stat)
        printf("Directory created\n");
    else
    {
        printf("Unable to create directory\n");
        exit(1);
    }
    getch();
    system("dir/p");
    getch();
    stat = rmdir(DIRNAME);
    if (!stat)
        printf("\nDirectory deleted\n");
    else
    {
        perror("\nUnable to delete directory\n");
        exit(1);
    }
    return 0;
}
/* Файлуудын нэрийг хайж хэвлэдэг жишээ программ.*/
#include <stdio.h>
#include <dir.h>

int main(void)
{
    struct ffblk ffblk;
    int done;
    printf("Directory listing of *.*\n");
```

## Программчлалын “Си” хэл

---

```
done = findfirst("*.",&ffblk,0);
while (!done)
{
    printf(" %s\n", ffblk.ff_name);
    done = findnext(&ffblk);
}
return 0;
}
```

### Урсгалаас унших урсгалд бичих

Урсгалын байрлалыг тогттох fseek функц (SEEK\_SET- урсгалын эхэнд заагчийг байрлуулна. SEEK\_END-урсгалын төгсгөлд заагчийг байрлуулна. SEEK\_CUR-Урсгалын одоо байгаа байрлалд заагчийг байрлуулна.)

**Бүтэц:** <Бүтэц> гэж нэрлэгдэх өгөгдлийн энэ хэлбэр нь янз бүрийн төрөлтэй өгөгдлийг хамтатган тодорхойлоход хялбархан бөгөөд хэрэглэгчдэд шинэ өгөгдлийн хэлбэрийг үүсгэхэд хэрэглэгддэг. Бичигдэх хэлбэр:

```
struct [<бүтцийн нэр>]
{
    [<төрөл1> <хувьсагчийн нэр1>];
    [<төрөл2> <хувьсагчийн нэр2>];
    .....
    [<төрөл n> <хувьсагчийн нэр n>];
};
```

Бүтэц доторх элементүүд нь янз бүрийн төрөлтэй байх бөгөөд элементэд хандахдаа (<.>) цэгийг хэрэглэнэ.

Бичигдэх хэлбэр:

<хувьсагчийн нэр>=<бүтцийн нэр>[.<талбар1,талбар2, . . талбар n>

Жишээ нь:

```
struct {
    char name[5];
    int age;
} sum;
```

<Бүтэц төрлийн массив>

# Программчлалын “Си” хэл

---

Жишээ программ.

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
struct student {
    char name[15];
    int age;
};

main()
{
    struct student sum[20];
    int i,n;
    clrscr();
    printf("N=");
    scanf("%d",&n);

    /* Бүтэц төрлийн массивт өгөгдөл оруулах*/
    for(i=1;i<=n;i++)
    { printf("Sum[%d].name=",i);
      scanf("%s",sum[i].name);
      printf("Sum[%d].age=",i);
      scanf("%i",&sum[i].age);
    }
    printf("\n\nStudent with 16<Age<35\n");
    for(i=1;i<=n;i++)
    {

        if((sum[i].age>16) && (sum[i].age<25))
        { printf("\nnumber %i. %s ",i,sum[i].name);
          printf("%i\n",sum[i].age);
        }
    }
    getch();
}
```

## Программчлалын “Си” хэл

---

```
/*Урсгалаас бүтэц төрлийн хувьсагчийг унших, урсгалд бүтэц төрлийн
хувьсагчийг бичих жишээ*/

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<conio.h>
struct student {
    char name[15];
    int age;
} data;

main()
{
    struct student sum[20],D;
    int i,n;
    /* stream Урсгалын заагч */
    FILE *stream;
    char name_student[12];
    int age_student,size;
    stream=fopen("Student.dat","w+");
    clrscr();
    printf("N=");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    { printf("Name=");
      scanf("%s",name_student);
    /* Бүтцийн name талбарт тэмдэгтэн мөрийг хуулж байна.*/
      strcpy(D.name,name_student);
      printf(" Age=");
      scanf("%i",&age_student);
    /* Бүтцийн age талбарт бүхэл тоон утгыг оруулж байна.*/
      D.age=age_student;
      size=fwrite(&D,sizeof(data),1,stream);
    }
    i=0;
    /* Урсгалын байрлалыг тогттох fseek функц (SEEK_SET- урсгалын эхэнд
заагчийг байрлуулна. SEEK_END-урсгалын төгсгөлд заагчийг байрлуулна.
```

## Программчлалын “Си” хэл

---

```
SEEK_CUR-Урсгалын одоо байгаа байрлалд заагчийг байрлуулна.*/
    fseek(stream, 0, SEEK_SET);
/* feof() Файлын төгсөл мөн эсэхийг тодорхойлно. */
    while(!feof(stream))
        { fread(&sum[i],sizeof(data),1,stream);
          i++;
        }
    fclose(stream);
getch();
for(i=0;i<n;i++)
    {
        { printf("\n number %i. %s  ",i,sum[i].name);
          printf("%i\n",sum[i].age);
        }
    }
getch();
}
```

# Программчлалын “Си” хэл

---

## Х бүлэг

### Турбо Си хэлний санах ойн моделиуд

Турбо Си хэлний программд санах ойг дараахь зорилгоор ашиглана.

1. Программын кодыг байрлуулахад.
2. Программын өгөгдлийг байрлуулахад.
3. Динамик байдлаар ашиглахад.
4. Программын биелэлтэд компиляторт нөлөөлөх зорилгоор.

Санах ойг дараахь байдлаар хуваана.

1. Видео санах ой.
2. Ашиглагдаагүй шуурхай санах ой.
3. Стек.
4. Чөлөөт санах ой.
5. Неар (куча)
6. Статик өгөгдлүүд.
7. Программын код.
8. Тасалдлын векторууд.

Статик өгөгдлүүд ба программын кодын санах ойн хэсгүүд програмын биелэлтийн явцад өөрчлөгдөхгүй.

Неар нь программын биелэлтэнд шаардагдах динамик санах ой учраас програмын явцад өөрчлөгдөх бололцоотой.

Стек нь программд ашиглагдаж байгаа локаль хувьсагч , функц процедуруудаас шалтгаалж өөрчлөгдөнө.

Видео санах ой, ашиглагдаагүй шуурхай санах ой. Нь өөрчлөгдөхгүй.

### Си хэлний санах ойн модель

6 төрлийн модель ашиглагдана.

TINY- (маш бага) Энэ моделийг ашиглаж байгаа үед code segment, (CS) data segment(DS), stack segment(SS), extra segment(ES) үндсэн 4 сегмент нэг хаягт ачаалагдана. Программын код, статик өгөгдлүүд, hear, стек нийлээд 64к-аас хэтрэхгүй. Санах ойн энэ моделийг санах ой маш бага үед ашиглана. Энэ тохиолдолд заагч хувьсагчууд 2 байт хэмжээтэй байх учраас зөвхөн тухайн сегментийн хувьд заагч ашиглагдана.

## Программчлалын “Си” хэл

---

SMALL- (бага) Энэ моделийг ашиглаж байгаа үед программын кодоод 64К, стек, статик, динамик өгөгдлүүдэд 64к байт ашиглагдана. Программ 2 сегментэд байрлах бололцоотой. Энэ нь бага хэмжээний асуудлыг шийдвэрлэх бололцоотой. Заагч төрлийн хувьсагчууд нь 2 байт хэмжээтэй байна. Заагч нь 1 сегмент дотор үйлчилнэ.

MEDIUM- (дунд зэрэг) Энэ тохиолдолд программын кодын хэмжээ 1М байт байж болно. Ийм учраас программын кодоод зориулж зориулж алсын заагчийн ашиглах хэрэгтэй. (far) Стек, heap, статик өгөгдлүүдэд 64К байт ашиглагдана. Программын код их, стек, heap, статик өгөгдлүүд бага үед хэрэглэж болно. Өөрөөр хэлбэл ашиглаж байгаа хувьсагч цөөн гэсэн үг. Заагч хувьсагч нь 4 байт хэмжээтэй байна. Ахлах 2 байт нь сегментийн дугаар, бага 2 байт нь сегментийн доторхи шилжилтийг заана. Программ доторхи заагч нь far, хувьсагч, статик өгөгдлүүд нь near төрлийнх байна.

COMPACT- Программын код нь 64К, өгөгдлүүдэд нь 1М байт, стекийн хэмжээ нь 64К-гаас ихгүй байна. Энэ тохиолдолд статик өгөгдлийн хэмжээ нь 64К-гаас хэтрэхгүй байна. Томоохон хэмжээний өгөгдлүүдтэй (1М байт хүртэлх) код нь их зай эзлээгүй үед хэрэглэнэ. Программын дотоод заагчууд нь near буюу 2 байт, өгөгдлийн заагч нь 4 байт буюу far байна.

LARGE- Энэ тохиолдолд программын кодоод 1М байт, статик өгөгдлүүдэд 64К, heap нь 1М байт хүртэл байна. Үүнийг том бодлогуудад хэрэглэнэ. Ямарч өгөгдлийн хэмжээ 64К-гаас хэтрэхгүй. Үүнд 4 байт хэмжээтэй алсын заагч хэрэглэнэ.(бүгдэд нь)

HUGE- Large-тай төсөөтэй. Статик өгөгдлүүдийн нийт хэмжээ 64К-гаас хэтэрч болно.

Хувиргагчтай тодорхойлогч (описатель с модификатором (модиф))

+ Const  
+ Volatile  
Cdecl  
Pascal  
+ Near  
+ Far  
+ Huge  
Interrupt



## Программчлалын “Си” хэл

---

+ тэмдэгүүдтэй нь нэг бол идентификаторт үйлчилнэ үгүй бол тэдгээрийн дараа орших одонд үйлчилнэ. Хэрэв модиф-ын баруун талд идентификатор байгаа бол өөрөөр хэлбэл өөрөөр хэлбэл идентификаторт (иден) үйлчилж байгаа бол тухайн иден-аар нэрлэгдэж буй объектийн төрөл хувиргагдана. Хэрэв хувиргагч нь одонд үйлчилж байгаа бол ийнхүү одоор заагдаж байгаа объектийн төрөл хувиргагдана. Өөрөөр хэлбэл од нь хувиргагдсан төрлийг зааж байна. (Заагчийг биш хувиргагдсан төрлийг зааж байна.)

Жишээ: 1. `int const *p` – `int` тогтмол төрлийн заагч

2. `int * const p` – бүхэл төрлийн тогтмол заагч

`Const`, `volatile` хувиргагчууд нь төрлийн тодорхойлолтын өмнө бичигдэж болно. Жишээ нь: `const int volatile`

Turbo C-д `near`, `far`, `huge`-ын хэрэглээ хязгаарлагдмал тэдгээр нь зөвхөн функцын нэрийн өмнө эсвэл одоны өмнө бичигдэнэ. Нэг объектын хувьд хэд хэдэн хувиргагч байж болно. Гэхдээ тэдгээрийн дараалал нь тодорхой утга агуулахгүй.

Жишээ: `far pascal`

`pascal far` –энэ хоёр ялгаагүй.

`int far *pascal far func()`; – `func` нь бүхэл тоон утгатай. Түүний өгөх бүхэл тоон утга нь `far` заагчаар хувиргагдана. Үүнээс гадна хаалт хэрэглэж уншиж болно.

`char far *(far *getint) (int far *)` –`getint far` заагч байна.Түүний

7 6 2 1 3 4

аргумент нь `int` төрлийн `far` заагч. Буцаах утга нь `char` байна.

6 өмнө уншигдах учир `far` төрлийн заагч.

`const volatile`

`const` хувиргагчаар тодорхойлогдсон хувьсагчид утга оноох үйлдлийг хийж болохгүй. Мөн `++`, `--` үйлдлийг хийж болохгүй. `Const` хувиргагчаар заагдсан заагчийн утгыг өөрчилж болохгүй. `Const` хувиргагчтай хувиргагчууд байнгын санах ойд ачаалагдах боломжтой.

`volatile` энэ нь

`const` хувиргагчийн эсрэг утгатай.

`Volatile` –оор заагдсан хувьсагч нь гадны нөлөөгөөр өөрчлөгдөх бололцоотой. Тасалдлын аргаар өөрчилж болно. `Volatile`-оор зарлагдсан хувьсагчийн утгыг регистрт хадгалдаггүй. Хэзээд хувиргагдана гэж үзнэ. `Const volatile`-г хамт хэрэглэж болно. Энэ тохиолдолд тухай хувьсагчийг программын аргаар өөрчилж болохгүй ба гадны аргаар өөрчилж болно. Гадны төхөөрөмжөөс авах

## Программчлалын “Си” хэл

утгаар өөрчлөгдөнө. Хэрэв `const volatile` хувиргагчуудаар нийлмэл хувьсагчууд зарлагдвал энэ нь түүний элемент бүр хамаарна.

```
float const pi=3.1415
```

```
const maxint=32767
```

```
char *const str="Hello";
```

```
char const *str2="Good morning"; гэж зарлагдсан гэвэл
```

болохгүй үйлдлүүд:

```
pi=3.04;
```

```
i=maxint--;
```

```
str="Good"; strcpy(str,"Good"); str – хаягаар байт байтаар нь хуулна.
```

Си дээр pascal хувиргагчууд.

Turbo C нь бусад системүүдтэй холбогдож ашиглагдана.(Өөрийн өгөгдлийг дамжуулах өөрөө өгөгдөл авах)

Ингэхэд 2 төрлийн асуудал тулгарна.

1.Гадны нэрийг оруулах

2.Параметр дамжуулах

Си компиляторын ажиллагааны үр дүнд программын объектийн код агуулсан файл үүснэ. (obj өргөтгөлтэй файл) Бүх үндсэн файлуудыг компиляци хийсний дараа бий болсон объектийн буюу obj өргөтгөлтэй файлуудыг link.exe ба tlink.exe файлуудын тусламжтайгаар биелэгдэх (exe, com өргөтгөлтэй) файл болгоно. Ингэж холбох явцад үндсэн файлын заагчийг гадаад (глобал) объектод холбох үйл ажиллагаа явагдана. Компиляцийн үе бүх гадаад идентификаторууд өөрөөр хэлбэл функцийн нэрүүд ба гадаад хувьсагчууд объектийн файлд бичигдэнэ. Хэрэв тодорхой зааж өгөөгүй бол эдгээр идентификаторууд нь өөрийн анхны бичлэг дээр бичигдэнэ.

Жишээ: `Format b:\4\s` тодорхой заасан гэсэн үг.

obj

Үүнээс гадна идентификатор бүрийн өмнө `_` тэмдэглэх зураасыг хэрэглэнэ.

(`_bb`). Зориуд зааж өгөөгүй тохиолдолд холбогч нь том жижиг үсгүүдийг ялган танина.

Бусад компиляторуудын тусламжтай үүссэн объектийн модуль дахь гадаад идентификаторуудыг таньж ялгахад Pascal, cdecl –г хэрэглэнэ.

Pascal

Ингэж зарлагдсан Pascal `_a` (*a* гэж тэмдэглэнэ.) гэж тэмдэглэсэн идентификатор нь "Си" программчлалын хэлэнд ерөнхий идентификатораар хэрэглэгдэхээс гадна Pascal программд ойлгомжтой болно. Хэрэв Pascal

## Программчлалын “Си” хэл

---

хувиргагчийг функцийн идентификатортай хэрэглэвэл функцийн параметруудийг дамжуулахад бас нөлөөлнө.

Жишээ: `func(int a, float b)` нь стекэд хадгалагдахдаа

3. ердийн үед `b, a`

4. Pascal-д `a, b` гэж хадгалагдана.

Pascal мод-г ашиглаж байх үед функцийн аргументүүдийн тоо тогтмол байх ёстой.

### **cdecl- ("Си" –ийн зарлалт)**

"Си" компилятор бүх идентификаторуудыг (функц ба түүний заагчууд) Pascal хувиргагчтай болгодог. Компиляцийн тохируулга (OPTIONS меню) хийгдэнэ. Энэ тохиолдолд зарим функцууд ба түүний заагчийг "Си" зарлалтай үлдээхийн тулд `cdecl` хэрэглэнэ.

### **near, far, huge**

Эдгээр хувьсагчууд нь объектийн хаягтай харьцахад хэрэглэгдэнэ.

`near`- Энэ төрлийг авах заагчийн урт 16 бит. `near`-н тусламжтай тухайн сегментийн доторх шилжилтийг тэмдэглэнэ.

`far`- 32 бит. Сегментийн хаяг түүн доторх шилжилтийг тэмдэглэнэ. Мөн түүн доторх шилжилтүүдийн хаягийг тодорхойлно.

`huge`- 32 бит. Сегментийн ба шилжилтийн хаягийг тодорхойлно. Үүн дээр хийгдэх `==, !=, <, >, <=, >=` харьцуулах үйлдлүүд `far` заагч дээр хийгдэх харьцуулах үйлдлүүдээс илүү байна. `Huge` төрлийн заагчийг ашиглаж байхад нэмэлт хугацаа шаардана. Программын кодын хэмжээ ихсэнэ.

ТС 1.5 дээр 64 кбайтаас илүү хэмжээтэй массив ашиглахгүй. Иймд `huge` заагчийг хэрэглэнэ.

Жишээ: `int huge;`

`char *far *z;`      `z`- `char` төрөл

алсын заагчийн заагч `char` төрөл

`double near cdecl calc (double, double)`

"Си" зарлалт `calc` –2 параметртэй тухайн сегментэд байна.

`char far *nextchar, far *prevchar, far *currentchar`

Олон `far` зарлахдаа тус тусад нь бичиж өгнө.

## **Interrupt**

Interrupt хувьсагчийг 8086, 8088 процессоруудын тасалдлын векторуудтай ажиллах функцуудыг зарлахад хэрэглэнэ. Interrupt мод-той зарлагдсан функцуудын хувьд компиляцийн хувьд нэмэлт кодын генераци хийнэ. Нэмэлт кодонд үндсэн региструудийн утга хадгалагдана.

# Программчлалын “Си” хэл

---

## Хэвлэх функц printf();

Бичигдэх хэлбэр:

```
printf(“%[<тод>][<тоо1>][<.тоо2>]<төрөл>”, <хувьсагч>);
```

Тод:

Бичигдээгүй курсорыг баруун талд тавина.

<тоо1>: Хэвлэх тооны урт (таслалаас өмнөх)

<тоо2>: Таслалаас хойш хэвлэх оронгийн тоо:

<хувьсагч>: Дэлгэцэнд хэвлэх өгөгдлийг агуулсан байна

<төрөл>: Хувьсагчийн зарласан төрөлтэй ижил байна.

d,l -бүхэл тоог 10-тын тооллын системд хэвлэх

u -бүхэл тоог тэмдэггүй 10-тын тооллын системд хэвлэх

o -бүхэл тоог 8-тын тооллын системд хэвлэх

x,X -бүхэл тоог 16-тын тооллын системд хэвлэх

f -бодит тоог дараахь байдлаар хэвлэх

[-] ddddd.ddddd

e,E -бодит тоог дараахь байдлаар хэвлэх

[-] d.dddddde[-][+]ddd

g,G - бодит тоог дараахь байдлаар хэвлэх (Утга оноогдсон хэлбэрээр хэвлэнэ)

[-] ddddd.ddddd

[-] d.dddddde[-][+]ddd

c -нэг тэмдэгт хэвлэх

s -тэмдэгтэн мөр хэвлэх

p -заагчийн хаягийг хэвлэх

Мөн printf() функцын дотор тайлбар үгнүүдийг хэвлэж болно.

## Гараас утга авах функц scanf();

Бичигдэх хэлбэр:

```
scanf(“%[урт]<төрөл>”, <хувьсагч>);
```

%c- Нэг тэмдэгт

%[W]c- Тэмдэгтэн массив

%s- Тэмдэгт мөр

## Программчлалын “Си” хэл

---

`%e`, `%E`, `%f`, `%g`, and `%G`- Хөвөгч-таслалтай тоо

`%d`, `%i`, `%o`, `%x`, `%D`, `%I`, `%O`, `%X`, `%c`, `%n`- Тэмдэггүй (unsigned)

Тоо бол

`d` - 10-тын тооллын системд бүхэл тоо - Pointer to int (int \*arg)

`D` - 10-тын тооллын системд бүхэл тоо - Pointer to long (long \*arg)

`e`, `E` - таслалтай тоо - Pointer to float (float \*arg)

`f` - таслалтай тоо - Pointer to float (float \*arg)

`g`, `G` - таслалтай тоо - Pointer to float (float \*arg)

`o` - 8-тын тооллын системд бүхэл тоо - Pointer to int (int \*arg)

`O` - 8-тын тооллын системд бүхэл тоо - Pointer to long (long \*arg)

`i` - 10, 8, 16-тын тооллын системд бүхэл тоо - Pointer to int (int \*arg)

`I` - 10, 8, 16-тын тооллын системд бүхэл тоо - Pointer to long (long \*arg)

`u` - 10-тын тооллын системд тэмдэгтгүй бүхэл тоо - Pointer to unsigned  
int (unsigned int \*arg)

`U` - 10-тын тооллын системд тэмдэгтгүй бүхэл тоо - Pointer to unsigned  
long (unsigned long \*arg)

`x` - 16-тын тооллын системд бүхэл тоо- Pointer to int (int \*arg)

`X` - 16-тын тооллын системд бүхэл тоо - Pointer to long (long \*arg)

Тэмдэгт бол

`s` - Тэмдэгтэн мөр - Тэмдэгтэн массивын заагч (char arg[])

`c` - Тэмдэгт - Тэмдэгтийн заагч (char \*arg)

Тэмдэгтийг тодорхой урттай авъя гэвэл. (such as `%5c`)

- Pointer to array of W chars (char arg[W])

**Текстэн цонх буюу дэлгэцэнд курсорын байрлалыг тогтоох функц**

**`gotoxy();`**

Идэвхтэй цонхонд өгөгдсөн координат дээр курсорыг шилжүүлнэ.

Бичигдэх хэлбэр:

`gotoxy(<x>,<y>);`

`<x>`- Бүхэл тоо. Хэвтээ координатыг тодорхойлно.

`<y>`- Бүхэл тоо. Босоо координатыг тодорхойлно.

# Программчлалын “Си” хэл

## Бие дааж бодох бодлогууд

1. Таны урьдчилан төлөвлөсөн 5 асуултанд хариулах программ бич. Хариулт буруу бол эхний тохиолдолд “Сайн бод”, дараагийн тохиолдолд “Буруу”, эцсийн тохиолдолд “Харамсалтай байна” гэсэн үг гарч зөв хариултыг мэдээлж байхаар бичигдэнэ. Эсрэг тохиолдолд сайшаасан үг гарч байхаар программа бич.
2. Таны урьдчилан төлөвлөсөн 5 асуултанд хариулах праграмм бич. Буруу хариултыг засаж программын эцэст хэд нь зөв болсныг мэдээлж байхаар бичигдэнэ. Зөв болж засагдсан хариу энэ тоонд орохгүй.
3.  $A(N,M)$  массивийн мөрийн элементүүдийн нийлбэр их байхын мөрийн элементүүдийг хэвлэ.
4.  $A(N,M)$  массивийн баганын элементүүдийн нийлбэр их байх баганын элементүүдийг хэвлэ
5.  $A(N,M)$  массивийн хамгийн бага элементийг агуулсан мөр ба баганын элементүүдийг хэвлэ. (мөрийг хэвтээ баганыг босоо хэвлэ.)
6.  $A(N,M)$  массивийн  $k$  дугаар мөрийн элементүүдийг массиваас хас
7.  $A(N,M)$  массивийн элементүүдийн гол диагналийн хувьд тэгш хэмтэй хувиргаж, шинээр үүссэн массивийг хэвлэ.
8.  $A(N,M)$  массивт өгөгдсөн утгуудаас хамгийн их элементийг олж байрлалыг тогтоо.
9. Санал асуулга явуулсан  $N$  асуудал дээр  $M$  хурлын гишүүд саналаа өгөх явцыг бүртгэх программ бич.
10.  $A(N,M)$  массивийн элементүүдийн сөрөг ба эерэг элементүүдийн нийлбэр тус бүрийг олох программ бич.
11.  $A(N,M)$  массивийн сөрөг ба эерэг элементүүдийн тоог олж хэрэв сөрөг элементүүдийн тоо их бол диагналаас доошхи элементүүдийг 0-р, эерэг тохиолдолд диагналаас дээшхи элементүүдийг 1-р соль.
12.  $y = (x^2 - 2x + 2) / (x - 1)$   $x \in [1, 4]$  үед “ $x$ ” хувьсагчийн  $h = 0.2$  алхмаар өөрчлөгдөх үеийн “ $y$ ” утгуудыг олж  $A(3, 5)$  массивт хадгал.
13.  $A(N)$  массивт байгаа ижил утгуудыг олж ижил утгууд бүрээс зөвхөн нэгийг үлдээ. Үлдсэн утгуудыг зүүн тийш шахаж үлдсэн хэсэгт 0-үүдийг бич
14.  $A(N, M)$  ба  $B(M, R)$  матрицын үржвэрийг ол.
15.  $A(N, M)$  ба  $B(N, M)$  матрицын нийлбэрийг ол.
16.  $[Z] = [A] + [B] * [C]$  матрицийн үржвэр нийлбэрийг ол.

## Программчлалын “Си” хэл

---

17. Өгөгдсөн квадрат матрицын гол диагналын элементүүдийг “1”-ээр, бусад элементүүдийг 0-ээр соль
18.  $A = \begin{bmatrix} 2 & 1 & 3 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \end{bmatrix}$  матриц өгөгджээ.  $[A]^3 - 2 * [A]^2 - 9 * [A]$  бодож ол.
19.  $A_1, A_2, A_3, \dots, A_n$  ба  $B_1, B_2, B_3, \dots, B_n$  тоон дарааллуудын элементүүдийн хамгийн их ялгаврыг ол.
20.  $A_1, A_2, A_3, \dots, A_n$  тоон дарааллын 0 биш элементүүдийг дарааллыг нь хадгалан дарааллын эхэнд, 0 элементүүдийг дарааллын төгсгөлд байрлуул.
21. Өгөгдсөн  $A$  квадрат матрицын гол диагналын дагуу байрлаж байгаа утгууд “1”-тэй тэнцүү бусад утгууд 0-тэй тэнцэж байхаар программ бич.
22.  $M$  тооноос хэтрэхгүй байх бүх анхны тоог ол.
23.  $A(N)$  массивт өгөгдсөн зэрэгцээ орших хос утгууд хоорондоо ялгаатай байна. Эдгээр зэрэгцээ орших хос утгуудын байрыг хооронд нь солих программ бич.
24.  $A(2, 15)$  массивт өгөгдсөн утгуудаас хоорондоо тэнцүү байх нөхцөлийг биелүүлсэн элементүүдийн байрлалыг тогтоох программ бич.
25. Дурын өгөгдсөн  $N$  натураль тоог квадратуудын нийлбэр байдлаар задалж бичих программ бич.
26.  $A(M)$  массивт өгөгдсөн тоонуудаас хоорондоо ялгаатай ямар нэг тоог хэд байгааг тодорхойлох программ бич. Жишээ нь: Массив дахь 5.7.5 элементүүдээс хоорондоо ялгаатай 5,7 гэсэн хоёр тоо байна.
27. Цифрүүдийн нийлбэр нь өгөгдсөн  $N$  тоотой тэнцэх гурван оронтой тоог олох программ бич.
28. Аравтын системд байгаа өгөгдсөн  $M$  натураль тоог хоёртын системд шилжүүлэх программ бич.
29.  $(N)$  массивт 0,1,2 гэсэн утгууд санамсаргүйгээр байрлана. Массив элементүүдийн байрлалыг бүх 0 утгууд, дараа нь бүх 1, сүүлд нь бүх 2 утга байхаар өөрчил.
31. Дурын өгөгдсөн  $A(N)$  массив дахь утгуудыг ихээс бага руу эрэмбэлэх програамм бич.
32. 10 тамирчдаас эхний 5 байранд орсон тамирчдын нэр, 5 нь Хоршооллынх болно.
33. отго гэх мэт өгөгдсөн дурын үгийг полиндром мөн бишийг тодорхойлох программ бич.



## Программчлалын “Си” хэл

---

34. Компьютрын дэлгэцийн нэг мөрөнд бичигдсэн өгүүлбэр дахь үгийн тоог тогтоох программ бич.
35. Өгөдсөн дурын мөрийн тэмдэгт бүрийн ASCII кодыг гаргах программ бич.
36.  $100 \times 100$  хэмжээтэй дөрвөлжин шугамтай цаасан дээр бие биенээ огтлоогүй, шүргээгүй, давхцаагүй тэгш өнцгүүд зурагдсан байна.  $A(100,100)$  массив өгөгдсөн бөгөөд энэ массивийн  $A(I,G)=1$  бол зурагдсан тэгш өнцөгтийн аль нэг нүдэнд байна. Эсрэг тохиолдолд  $A(I,G)=0$  гэж үзнэ. Тэгш өнцөгтүүдн тоог тодорхойлох программ бич.
37. 7 оронгоос хэтрэхгүй байх 0-1 хооронд орших бутархай тоонуудыг өсөх эрэмбээр хэвлэх программ бич.
38. Бүхэл тоон  $A(N)$  массив,  $M$  тоо өгөгджээ.  $A(1)+A(2)+A(3)\dots A(n)=M$  нөхцлийг биелүүлж байх  $A(N)$  массивийн элементүүдийг тодорхойлох программ бич.
39.  $A(N,M)$  массивт өгөгдсөн утгуудаас мөрөндөө хамгийн бага байх, баганадаа хамгийн их байх нөхцлийг биелүүлсэн элементүүдийн байршлыг олох программ бич. Хэрэв ийм утга байхгүй бол 0-ийг хэвлэ.
40. Өгөгдсөн натурал тоо  $M, N$ -ийн хувьд  $M/N=M1/N1$  нөхцлийг биелүүлэх  $B1, N1$  тоог олох программ бич.
41. Бүхлээр хуваадаг тоонуудынхаа нийлбэртэй тэнцдэг натураль тоог өгөгдсөн  $M$  тоо хүртэл олох программ бич. Жишээ нь:  $6=1+2+3$ .
42. Эрэмбэлэгдсэн  $A(N), B(M)$  массиваас эрэмбэлэгдсэн  $C(N+M)$  массив үүсгэ.
43.  $A(N,N)$  массив зөвхөн 0,1,5,11 гэсэн утгуудаас тогтоно. Элементүүд нь хоорондоо ялгаатай байх  $A(I,G), A(I,G+1), A(I+1,G+1)$  дөрвөлсөн утга уг массивт хэд байгааг тодорхойлох программ бич.
44.  $A(N)$  массивт өгөгдсөн утгуудаас хамгийн олон удаа тохиолдох цифрийг тодорхойлох программ бич.
45.  $A(N), B(N)$  массивт өгөгдсөн утгуудаас  $A$  массивт min байхаар,  $B$  массивт max байхаар хос тоог ол.  $A, B$  массивт олдох тоонуудын байрлалууд давхцах ёстой.
46. Өгөгдсөн  $M$  тоог бүхлээр хувааж чадах тоон олонлогийг олох программ бич.
47. Савлах цехийн  $N$  ажилчны ажлын 6 хоногт хийсэн ажлыг доорхи үзүүлэлтээр дүгнэх программ бич.
  - А. 6 хоногт нийт савласан бүтээгдэхүүний дундаж
  - Б. 6 хоногт хамгийн өндөр бүтээл гаргасан ажилчны нэр, савласан бүтээгдэхүүний тоо, гариг.
48. Өгөгдсөн өгүүлбэр дахь үгүүд нь хэдэн удаа давтагдсан болохыг тогтоо.
49. Өгөгдсөн 2 өгүүлбэрт зэрэг орсон хамгийн урт үгийг ол
50. Өгөгдсөн өгүүлбэрт тэгш тоон урттай тэгш хэмгүй үгүүд байгаа эсэхийг тогтоо.

## Программчлалын “Си” хэл

---

51. Өгөгдсөн өгүүлбэр дэх үг тус бүрийн эгшгүүдийг тоог олж хамгийн их эгшиг орсон үгийг хэвлэ.
52. Өгөгдсөн өгүүлбэр дээр үг тус бүрийн гийгүүлэгчдийн тоог олж хамгийн бага гийгүүлэгч орсон үгийг хэвлэ.
53. Өгөгдсөн элементүүдийн дараалалд пробелууд (хоосон зай ) орсон бол түүнийг тэмдэгтүүдийн дарааллаас хасаад зүүн тийш нь шахаж хэвлэ.
54. Өгөгдсөн өгүүлбэрт хамгийн их давтагдсан тэмдэгтийг ол.
55. Өгөгдсөн өгүүлбэрт давтагдах тэмдэгтүүдийн байрлалын дугаарыг хэвлэ.
56. Өгөгдсөн N тооноос бага 2-тын тооллын бичлэг нь тэгш хэмтэй байх бүх анхны тоонуудыг ол.
57. Өгөгдсөн n тоог анхны тоонуудын үржвэрт задал. Жишээ нь:  $8=2*2*2$
58. Өгөгдсөн натураль тоон олонлогоос 2-тын тооллын бичлэг нь хамгийн их нэг орсон байх бүх тоонуудыг ол.
59. Өгөгдсөн n тооноос бага натураль тоонуудын 2-тын тооллын бичлэг дахь 1-үүдийн байрлалын дугаар арифметик прогресс үүсгэх бүх натураль тоог ол.
60. Цифрүүдийн нийлбэр нь n-тэй тэнцэх бүх 3-н оронтой тоог тодорхойл.
61.  $Ax^2 + Bx + C = 0$  тэгшитгэлийг бод.
62. Гараас өгөгдсөн натураль тоог үсгээр бичих программ бич. (100 хүртлэх )
63. Өгөгдсөн өгүүлбэр дэх үгийн тоог тогтоох программ бич.
- 64 Гараас өгөгдсөн энгийн илэрхийллийн утгыг олох программ бич.  
Ж нь:  $12+40=52$ ,  $10*3=30$ ,  $99/9=11$ ,  $100-190=-90$ .
65. A(N) массивийг өсөх, буурахаар эрэмбэл.
66. A(N) массиваас нэг элемент үлдээ. n-р элементийг хас.  
Ж нь: (1,3,-1,2,1) n=3  
1. (1,3,2,1) 2. (1,3,1) 3. (3,2)
67. A(NM), B(MK) матрицийн үржвэрийг C(NK) массивт хадгал.

# Программчлалын “Си” хэл

---

## Бие дааж бодох бодлогуудын бодолт

### Бодлого № 1

```
#include<stdio.h>
#include<conio.h>
void main (void)
{
int i, j=0, ch, p=0;
char *a[]={ "Russian", "Japan", "England", "USA", "Mongolia"};
char *b[]={ "Moscow", "Tokyo", "London", "Washinton", "UB"};
clrscr();
for(i=0;i<5;i++)
    printf("%d.%s\n",i+1,b[i]);
for(i=0;i<5;i++)
{ printf("\nWhat is the capital's name %s \n number:=",a[i]);
while(j<3)
{ ch=getch();
if( (ch+1)%10==i)
{ printf("Good.\n");
j=3;
}
else
{ switch (j)
{ case 0: {printf("Think.\n number:="); j++;
} break;
case 1: { printf("Error.\n number:="); j++;
} break;
case 2: { printf("I'm sorry.\n\n"); j++; p=1;
} break;
}
}
}
if (p==1) printf("This country capital's name is %s.\n",b[i]);
p=0; j=0;
}
getch();
}
```

### Бодлого № 2

## Программчлалын “Си” хэл

---

```
#include<stdio.h>
#include<conio.h>
void main (void)
{
int i,j=0,ch,p,n=0;
char *a[]={ "Ингэ", "Гүү", "Үнээ", "Ямаа", "Хонь"};
char *b[]={ "Ботго", "Унага", "Тугал", "Ишиг", "Хурга"};
clrscr();
for(i=0;i<5;i++)
{ printf("%d.%s\n",i+1,b[i]);
}

for(i=0;i<5;i++)
{ printf("\n%сний төл юу вэ? \n number:=",a[i]);
ch=getch(); p=0;
while(j<3)
{ if((ch+1)%10==i)
{ printf("Сайн байна.\n");
j=3;
if(p==0) n++;
}
else
{ printf("Буруу байна.\n number:="); j+=2; p=1;
ch=getch();
}
} j=0;
if(p==1) printf("Энэ амьтны төл нь -- %s.\n",b[i]);
}
if(n==5) printf("ОК.");
printf("\n\nЗөв хариулт = %d",n);
getch();
}
```

### Бодлого № 3

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
```

## Программчлалын “Си” хэл

---

```
int a[100][100],min=100,s=0,i,j,n,save;
clrscr();randomize();
printf("n=");scanf("%i",&n);
for(i=0;i<n;i++)
    { for(j=0;j<n;j++)
        { a[j][i]=random(10)-5;
          if(a[j][i]>=0) printf(" ");
          printf(" %i",a[j][i]);
          s=s+a[j][i];
        }
        if (min>s)
            { save=i; min=s;
              }
            s=0;
            printf("\n");
        }
printf("\nsum min column %i\n",min);
for(i=0;i<n;i++)
    { if(a[i][save]>=0)
        printf(" ");
        printf(" %i",a[i][save]);
    }
printf("\ncolumn=%i\n",save+=1);
getch();
}
```

### Бодлого № 4

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
int a[100][100],s=0,i,j,n,max=0,save;
clrscr(); randomize();
printf("N=");scanf("%i",&n);
for(i=0;i<n;i++)
    { for(j=0;j<n;j++)
        { a[i][j]=random(10);
          printf(" %i",a[i][j]);
        }
    }
```

## Программчлалын “Си” хэл

---

```
    s=s+a[i][j];
}
if(max<s)
{ save=i; max=s;
}
s=0;
printf("\n");
}
printf("\nSum max=%i\n",max);
for(j=0;j<n;j++)
printf(" %d",a[save][j]);
printf("\nmor=%i",save+=1);
getch();
}
```

### Бодлого № 5

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
int a[100][100],min=1000,sx,sy,i,j,n;
clrscr(); randomize();
printf("n=");scanf("%i",&n);
for(i=0;i<n;i++)
{ for(j=0;j<n;j++)
{ a[i][j]=random(10);
printf(" %i",a[i][j]);
if(min>a[i][j])
{min=a[i][j]; sx=i; sy=j;
}
}
printf("\n");
}
printf("\n");
for(i=0;i<n;i++)
{ for(j=0;j<n;j++)
{ if(i==sx||j==sy) printf(" %i",a[i][j]);
else printf(" ");
}
```

## Программчлалын “Си” хэл

---

```
    }
    printf("\n");
}
getch();
}
```

### Бодлого № 6

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
void main (void)
{
int i,j,ch,p,n,m,a[100][100];
clrscr();
randomize();
printf("Input N=");scanf("%d",&n);
printf("Input M=");scanf("%d",&m);
for(i=0;i<n;i++)
{ for(j=0;j<m;j++)
  { a[j][i]=random(10);
    printf("%d ",a[j][i]);
  }
  printf("\n");
}
printf("Input %i>Number=",n);scanf("%d",&p);p--;
for(i=0;i<n;i++)
  { for(j=0;j<m;j++)
    { if(i!=p) printf("%d ",a[j][i]);
      }
    printf("\n");
  }
getch();
}
```

### Бодлого № 7

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
```

## Программчлалын “Си” хэл

---

```
{
int a[100][100],m,j,i,n;
clrscr(); randomize();
printf("n="); scanf("%i",&n);
for(i=0;i<n;i++)
    { for(j=0;j<n;j++)
      { a[i][j]=random(10);
        printf(" %i",a[i][j]);
      }
      printf("\n");
    }
printf("\n");
for(j=0;j<n;j++)
    { for(i=0;i<n;i++)
      { m=a[i][j];
        a[i][j]=a[j][i];
        a[j][i]=m;
      }
    }
for(j=0;j<n;j++)
    { for(i=0;i<n;i++)
      printf(" %i",a[i][j]);
      printf("\n");
    }
getch();
}
```

### Бодлого № 8

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
int a[100][100],max=-100,column,row,i,j,n;
clrscr(); randomize();
printf("n="); scanf("%i",&n);
for(i=0;i<n;i++)
    { for(j=0;j<n;j++)
      { a[i][j]=random(20)+10;
        printf(" %i",a[i][j]);
      }
    }
```



## Программчлалын “Си” хэл

---

```
        if(max<a[i][j])
        {   max=a[i][j];
            column=i;
            row=j;
        }
    }
    printf("\n");
}
printf("column=%i\nrow=%i",column+1,row+1);
getch();
}
```

### Бодлого № 9

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
void main ()
{
int j,i,n=4,m=3;
char a[4][3];
clrscr();
for(i=0;i<n;i++)
    {   printf("*** %i-r khun ***",i+1);
        for(j=0;j<m;j++)
            {   printf("\n%i-r sanal (y/n):",j+1);
                a[i][j]=getche();
            }
        printf("\n");
    }
for(i=0;i<n;i++)
    {   for(j=0;j<m;j++)
        printf("%c ", a[i][j]);
        printf("\n");
    }
getch();
}
```

### Бодлого № 10

```
#include<stdio.h>
```

## Программчлалын “Си” хэл

---

```
#include<stdlib.h>
#include<conio.h>
void main ()
{
int i,j,n,a[10][10];
int s=0,e=0;
randomize(); clrscr();
printf("N="); scanf("%i",&n);
for(i=0;i<n;i++)
    { for (j=0;j<n;j++)
        { a[i][j]=random(10)-5;
          if(a[i][j]>=0)
            { s+=a[i][j];
              printf(" ");
            }
          else e+=a[i][j];
            printf(" %i",a[i][j]);
          }
        printf("\n");
    }
printf("SUM(+)=%i \nSUM(-)=%i",s,e);
getch();
}
```

### Бодлого № 11

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
int a[100][100],n,i,j,s=0,e=0;
clrscr(); randomize();
printf("n="); scanf("%i",&n);
for(i=0;i<n;i++)
    { for(j=0;j<n;j++)
        { a[i][j]=random(10)-5;
          if(a[i][j]>=0)
            { e++; printf(" ");
            }
          }
        }
}
```

## Программчлалын “Си” хэл

---

```
        else s++;
        printf(" %i",a[i][j]);
    }
    printf("\n");
}
i=0;
printf("\nEl(+)=%i El(-)=%i\n",e,s);
if(e>s)
{ while(i<=n)
    { for(j=i;j<n;j++)
        a[i][j]=1;
        i++;
    }
}
else
{ while(i<=n)
    { for(j=0;j<=i;j++)
        a[i][j]=0;
        i++;
    }
}
for(i=0;i<n;i++)
    { for(j=0;j<n;j++)
        { if(a[i][j]>=0)
            printf(" ");
            printf(" %i",a[i][j]);
        }
        printf("\n");
    }
getch();
}
```

### Бодлого № 12

```
#include <stdio.h>
#include <conio.h>
void main()
{
float a[3][5],y,x=1.2;
int i,j;
```

## Программчлалын “Си” хэл

---

```
clrscr();
for(i=0;i<3;i++)
    for(j=0;j<5;j++)
        { y=(x*x-2*x+2)/(x-1);
          a[i][j]=y;
          x+=0.2;
        }
for(i=0;i<3;i++)
    { for(j=0;j<5;j++)
      printf("%f ",a[i][j]);
      printf("\n");
    }
getch();
}
```

### Бодлого № 13

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
int a[100],j=0,i,n,save;
clrscr(); randomize();
printf("n="); scanf("%i",&n);
for(i=0;i<n;i++)
    { a[i]=random(4);
      printf(" %i",a[i]);
    }
while(j<n)
    { for(i=j+1;i<n;i++)
      if(a[j]==a[i])
          { save=a[i];
            a[i]=0;
            a[j]=save;
          }
      j++;
    }
printf("\n");
for(j=0;j<n;j++)
```

## Программчлалын “Си” хэл

---

```
{ for(i=0;i<n;i++)
  if(a[i]==0)
  { save=a[i];
    a[i]=a[j];
    a[j]=save;
  }
}
printf("\n");
for(i=0;i<n ;i++)
printf(" %i",a[i]);
getch();
}
```

### Бодлого № 14

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
int main(void)
{
int i,j,l,m,n,r,p=0,a[100][100],b[100][100],c[100][100];
clrscr(); randomize();
printf("N="); scanf("%d",&n);
printf("M="); scanf("%d",&m);
printf("R="); scanf("%d",&r);
for (j=0;j<n;j++)
  { for (i=0;i<m;i++)
    { a[i][j]=random(10);
      printf("%d ",a[i][j]);
    }
    printf("\n");
  }
printf("\n");
for (i=0;i<m;i++)
  { for (j=0;j<r;j++)
    { b[j][i]=random(10);
      printf("%d ",b[j][i]);
    }
    printf("\n");
  }
}
```

## Программчлалын “Си” хэл

---

```
printf("\n");
for(l=0;l<r;l++)
    { for(j=0;j<n;j++)
        { for(i=0;i<m;i++)
            { p=p+a[i][j]*b[l][i];
              }
            c[l][j]=p;p=0;
            printf("%d ",c[l][j]);
          }
        printf("\n");
    }
getch();
return 0;
}
```

### Бодлого № 15

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
main()
{
int c[100][100],b[100][100],a[100][100],m,r,s=0,j,i,n,k=0,l;
clrscr(); randomize();
printf("n="); scanf("%i",&n);
printf("m="); scanf("%i",&m);
for(i=0;i<n;i++)
    { for(j=0;j<m;j++)
        { a[i][j]=random(10);
          b[i][j]=random(10);
          printf("%d ",a[i][j]);
        }
        printf("\n");
    }
printf("*****\n");
for(i=0;i<n;i++)
    { for(j=0;j<m;j++)
        printf("%d ",b[i][j]);
        printf("\n");
    }
}
```

## Программчлалын “Си” хэл

---

```
printf("*****\n");
for(i=0;i<n;i++)
  { for(j=0;j<m;j++)
    printf("%d ",b[i][j]+a[i][j]);
    printf("\n");
  }
getch();
}
```

### Бодлого № 16

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
int main(void)
{
int i,j,l,m,n,r,p=0,a[100][100],b[100][100],c[100][100],d[100][100];
clrscr(); randomize();
printf("N="); scanf("%d",&n);
printf("M="); scanf("%d",&m);
printf("R="); scanf("%d",&r);
for (j=0;j<n;j++)
  { for (i=0;i<m;i++)
    { a[i][j]=random(10);
      printf("%d ",a[i][j]);
    }
    printf("\n");
  }
printf("\n");
for (i=0;i<m;i++)
  { for (j=0;j<r;j++)
    { b[j][i]=random(10);
      printf("%d ",b[j][i]);
    }
    printf("\n");
  }
printf("\n");

for (i=0;i<r;i++)
{ for (j=0;j<n;j++)
```

## Программчлалын “Си” хэл

---

```
{ c[j][i]=random(10);
  printf("%d ",c[j][i]);
}
printf("\n");
}
printf("\n");
for(l=0;l<r;l++)
  { for(j=0;j<n;j++)
    { for(i=0;i<m;i++)
      p=p+a[i][j]*b[l][i];
      d[l][j]=p;p=0;
      printf("%d ",d[l][j]);
    }
    printf("\n");
  }
printf("\n");
for(i=0;i<r;i++)
  { for(j=0;j<n;j++)
    printf("%d ",c[j][i]+d[i][j]);
    printf("\n");
  }
getch();
return 0;
}
```

### Бодлого № 17

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
void main()
{
int a[100][100],j,i,n;
clrscr();randomize();
printf("n=");scanf("%i",&n);
for(i=0;i<n;i++)
  { for(j=0;j<n;j++)
    { a[i][j]=random(10);
      printf(" %i",a[i][j]);
    }
  }
```



## Программчлалын “Си” хэл

---

```
    printf("\n");
}
printf("\n");
for(i=0;i<n;i++)
{ for(j=0;j<n;j++)
  { if(i==j)
    a[i][j]=1;
    else
    a[i][j]=0;
    printf(" %i",a[i][j]);
  }
  printf("\n");
}
getch();
}
```

### Бодлого № 20

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
main()
{
int a[100],j,i,n,save;
clrscr();randomize();
printf("n=");scanf("%i",&n);
for(i=0;i<n;i++)
  { a[i]=random(5);
    printf(" %i",a[i]);
  }
for(j=0;j<n;j++)
  { for(i=0;i<n;i++)
    if(a[i]==0)
      {
        save=a[i];
        a[i]=a[j];
        a[j]=save;
      }
  }
printf("\n\n");
```

## Программчлалын “Си” хэл

---

```
for(i=0;i<n;i++)
    printf(" %i",a[i]);
getch();
return(0);
}
```

### Бодлого № 21

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
main()
{
int a[100][100],j,i,n;
clrscr(); randomize();
printf("n="); scanf("%i",&n);
for(i=0;i<n;i++)
    { for(j=0;j<n;j++)
        { if (i==j)
            a[i][j]=1;
          else
            a[i][j]=0;
          printf(" %i",a[i][j]);
        }
      printf("\n");
    }
getch();
return(0);
}
```

### Бодлого № 22

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
int m,j,i,k;
clrscr();
printf("m=");scanf("%d",&m);
for(i=2;i<=m;i++)
    { k=0;
```

## Программчлалын “Си” хэл

---

```
    for(j=2;j<i;j++)
        if(0==(i%j)) k++;
        if(k==0)
            printf("%d\n",i);
    }
getch();
}
```

### Бодлого № 23

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
main()
{
int a[100],j,i,n,k,m;
clrscr(); randomize();
printf("n="); scanf("%i",&n);
for(i=0;i<n;i++)
    { a[i]=random(10);
      printf(" %i",a[i]);
    }
for(i=0;i<n-1;i+=2)
    { m=a[i];
      a[i]=a[i+1];
      a[i+1]=m;
    }
printf("\n");
for(i=0;i<n;i++)
    printf(" %i",a[i]);
getch();
return(0);
}
```

### Бодлого № 26

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
main()
{
```

## Программчлалын “Си” хэл

---

```
int a[100],j,i,n,k,m,t=0,l=0;
clrscr(); randomize();
printf("n="); scanf("%i",&n);
for(i=0;i<n;i++)
    { a[i]=random(10);
      printf(" %i",a[i]);
    }
for(i=0;i<n;i++)
if (a[i]==0)
    t++;
for(j=0;j<n;j++)
    { k=a[j];
      for(i=0;i<n;i++)
          if (a[j]==a[i])
              { m=a[j];
                a[j]=0;
                a[i]=m;
              }
    }
for(j=0;j<n;j++)
    { k=a[j];
      for(i=0;i<n;i++)
          if (a[i]==0)
              { m=a[i];
                a[i]=a[j];
                a[j]=m;
              }
    }
for(i=0;i<n;i++)
    if (a[i]==0)
        l++;
if(t!=0)
    l=l-1;
printf("\n");
for(i=0;i<n-l;i++)
    printf(" %i",a[i]);
getch();
return(0);
}
```

# Программчлалын “Си” хэл

---

## Бодлого № 27

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
main()
{
int j,i,n,l,m;
clrscr(); randomize();
printf("n="); scanf("%i",&n);
for(i=1;i<10;i++)
    { for(j=0;j<10;j++)
      for(l=0;l<10;l++)
        { m=i+j+l;
          if (m==n)
            printf(" %i",100*i+10*j+l);
        }
    }
getch();
return(0);
}
```

## Бодлого № 28

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
main()
{
int a[100],j,i=0,n,k,m;
clrscr();
printf("n="); scanf("%i",&n);
while(n!=0)
    { k=n%2;
      a[i]=k;
      n=n/2;
      i++;
    }
j=i-1;
for(i=j;i>=0;i--)
```

## Программчлалын “Си” хэл

---

```
    printf(" %i",a[i]);
getch();
return(0);
}
```

### Бодлого № 29

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
main()
{
int a[100],j,i,n,k,m;
clrscr(); randomize();
printf("n="); scanf("%i",&n);
for(i=0;i<n;i++)
    { a[i]=random(3);
      printf(" %i",a[i]);
    }
for(j=0;j<n;j++)
    { k=a[j];
      for(i=0;i<n;i++)

          if (a[j]<=a[i])
              { m=a[i];
                a[i]=a[j];
                a[j]=m;
              }
    }
printf("\n");
for(i=0;i<n;i++)
printf(" %i",a[i]);
getch();
return(0);
}
```

### Бодлого № 31

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
```

## Программчлалын “Си” хэл

---

```
main()
{
int a[100],j,i,n,m;
clrscr();randomize();
printf("n=");scanf("%i",&n);
for(i=0;i<n;i++)
{ a[i]=random(10);
  printf(" %i",a[i]);
}
for(j=0;j<n-1;j++)
  for(i=j+1;i<n;i++)
    if (a[j]>=a[i]) {m=a[i];a[i]=a[j];a[j]=m;}

printf("\n");
for(i=0;i<n;i++)
printf(" %i",a[i]);
getch();
return(0);
}
```

### Бодлого № 33

```
# include <stdio.h>
# include <string.h>
void main()
{ int i,l,p=0;
char *s;
clrscr();
gets(s);
l=strlen(s);
for (i=0;i<=l/2;i++)
if (s[i]!=s[l-i-1]) p++;
if (!p) printf("\n this word is Palindrom");
else printf("\n this word is not palindrom");
  getch();
}
```

### Бодлого № 34

```
#include<stdio.h>
#include<conio.h>
```

## Программчлалын “Си” хэл

---

```
#include<string.h>
void main()
{ char str[80];
  int i,count=1;
  clrscr();
  printf("Input string: "); gets(str);
  for(i=0;i<strlen(str);i++)
  if(str[i]== ' ' || str[i]==',' || str[i]=='.')
  if (str[i+1]!=' ' && str[i+1]!=',' && str[i+1]!='.'&& str[i+1]!='\0')
  count++;
  printf("ygiin too %d\n",count);
  getch();
}
```

### Бодлого № 35

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
void main()
{ char *str;
  int i;
  clrscr();
  printf("Input string ");scanf("%s",str);
  for(i=0;i<strlen(str);i++)
  printf("%c = %d\n",str[i],str[i]);
  getch();
}
```

### Бодлого № 42

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
void main ()
{ int i,j,max=0,n,m,a[100],b[100],c[100];clrscr();
  printf("A(N)-бвж N=");scanf("%i",&n);
  for (i=0;i<n;i++)
  { gotoxy (i*3+3,2);scanf("%i",&a[i]);}
  printf("B(M)-бвж M=");scanf("%i",&m);
  for(i=0;i<m;i++)
```



## Программчлалын “Си” хэл

---

```
{ gotoxy (i*3+3,4);scanf("%i",&b[i]);}
for(i=0;i<n;i++)
{for (j=0;j<n;j++)
  {if(a[i]<a[j])
    {max=a[j];a[j]=a[i];a[i]=max;}
  }
} printf("A(N)");
printf("\n");
for (i=0;i<n;i++)
printf(" %i",a[i]);max=0;
for(i=0;i<m;i++)
{for (j=0;j<m;j++)
  {if(b[i]<b[j])
    {max=b[j];b[j]=b[i];b[i]=max;}
  }
}
printf("\nB(N)");
printf("\n");
for (i=0;i<m;i++)
printf(" %i",b[i]);
for(i=0;i<n;i++)
c[i]=a[i];
for (i=n;i<n+m;i++)
c[i]=b[i-n];max=0;
for(i=0;i<m+n;i++)
{for (j=0;j<m+n;j++)
  {if(c[i]<c[j])
    {max=c[j];c[j]=c[i];c[i]=max;}
  }max=0;
}
printf("\n C(N+M) "); printf("\n");
for (i=0;i<m+n;i++) printf(" %i",c[i]);
getch();
}
```

Бодлого № 44

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
```

## Программчлалын “Си” хэл

---

```
void main (void)
{ int e,j,n,max=0,a[100],b[10];
clrscr();
printf("Input N=");
scanf("%i",&n);
for (j=0;j<n;j++)
    {printf("A(%i)=",j);
    scanf("%i",&a[j]);
    }
printf("\n");
for (j=0;j<10;j++)
    b[j]=0;
for (j=0;j<n;j++)
while(a[j]){
    b[a[j]%10]++;
    a[j]=a[j]/10;
}
    max=b[0]; e=0;
for (j=0;j<10;j++)
if (max<b[j]) {max=b[j];e=j;}
    printf("\n");
    printf("\n max number=%i",e);
getch();
}
```

### Бодлого № 45

```
# include <stdio.h>
# include <conio.h>
# include <stdlib.h>
void main()
{ int i,max,min,e,s,m,n,a[100],b[100];
clrscr(); randomize();max=0;min=100;
printf("\n A(m)-M=");scanf("%i",&m);
printf("\n B(n)-N=");scanf("%i",&n);
for (i=0;i<m;i++)
    {a[i]=random(2);printf(" A(m) %i",a[i]);}
printf("\n");
for (i=0;i<n;i++)
    {b[i]=random(2);
```

## Программчлалын “Си” хэл

---

```
printf(" B(n) %i",b[i]);
for (i=0;i<m;i++)
{if (min>a[i]
  {min=a[i];e=i;}
  }
for (i=0;i<n;i++)
{if (max<b[i]
  {max=b[i];s=i;}
  }
if(e==s)
printf("\n max:min=%i,A(i)=%i,B(j)=%i",max,e,s);
else printf("\n No");
getch();
}
```

### Бодлого № 46

```
# include<stdio.h>
void main ()
{ int i,m; clrscr();
printf("\n Input=");
scanf("%i",&m);
for (i=1;i<=m;i++)
{if(0==m%i)
  printf("\n %i",i);}
getch();
}
```

### Бодлого № 48

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
char *a[100];
char sent[255]="";
char *p;
int i=0,j,count,b[100],c[100];
main()
{ clrscr();
printf("Input...");
gets(sent);
```

## Программчлалын “Си” хэл

---

```
p=strtok(sent,".,?!");
if (p)
{ a[i]=p;
  while (1)
  { p=strtok(NULL,".,?!");
    if (p) { i++; a[i]=p; c[i]=0; }
    else break;
  }
  count=i;
  for (i=0;i<=count;i++)
    for (j=i;j<=count;j++)
      if (strcmp(strupr(a[i]),strupr(a[j]))==0)
      { b[i]=b[i]+1;
        if (i!=j) c[j]=1;
      }
  printf("\n\n");
  printf("-----\n");
  for (i=0;i<=count;i++)
    if (c[i]!=1)
      if (i!=count) printf("%s %d \n",strupr(a[i]),b[i]);
      else printf("%s %d ",strupr(a[i]),b[i]);
  }
else printf("\n\n");
getch();
}
```

### Бодлого № 53

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main()
{
  char sent[255];
  int i,j=0,length;
  clrscr();
  printf("Input...");
  gets(sent);
  length=strlen(sent);
  for (i=0;i<length;i++)
    if (sent[i]!=' ')
```

## Программчлалын “Си” хэл

---

```
    sent[j++]=sent[i];
sent[j]='\0';
puts(sent);
getch();
}
```

### Бодлого № 54

```
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>
char sent[255];
int i,j,length,max=0,maxchar,count=0,chr;
main()
{ clrscr();
  printf("Input string..");
  gets(sent);
  length=strlen(sent);
  for (i=0;i<length-1;i++)
  { count=1;
    chr=sent[i];
    for (j=i+1;j<length;j++)
      if (sent[j]==chr) count++;
    if (count>max) { max=count;
                    maxchar=chr; }
  }
  printf("%c maximum character = %d",maxchar,max);
  getch();
}
```

### Бодлого № 61

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
main()
{
  int a,b,c,d;
  float x1,x2;
  clrscr();
```

## Программчлалын “Си” хэл

---

```
printf("A="); scanf("%d",&a);
printf("B="); scanf("%d",&b);
printf("C="); scanf("%d",&c);
    d=b*b-4*a*c;
    if (d>=0){
x1=(-b+sqrt(d))/(2*a);
x2=(-b-sqrt(d))/(2*a);
printf("x1=%f  x2=%f",x1,x2);
    }
else
printf("No result");
getch();
}
```

### Бодлого № 65

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
main()
{ int a[100],j,i,k,n,m;
clrscr();randomize();
printf("n=");scanf("%i",&n);
printf(" anhnii massib :");
printf("\n");
for(i=0;i<n;i++)
{ a[i]=random(10);
printf(" %i",a[i]);
}
printf("\n byyrahaar erembelsen :");
for(j=0;j<n;j++)
{ k=a[j];
for(i=0;i<n;i++)
if (a[j]>=a[i]) {m=a[i];a[i]=a[j];a[j]=m;}
}
printf("\n");
for(i=0;i<n;i++)
printf(" %i",a[i]);
printf("\n osohoor erembelsen :");
for(j=0;j<n;j++)
```

## Программчлалын “Си” хэл

---

```
{ k=a[j];
  for(i=0;i<n;i++)
    if (a[j]<=a[i]) {m=a[i];a[i]=a[j];a[j]=m;}
}
printf("\n");
for(i=0;i<n;i++)
printf(" %i",a[i]);
getch();
return(0);
}
```