

18-AMALIY MASHG'ULOT. KO'RSATKICHLAR. ADRES OLUVCHI O'ZGARUVCHILAR

NAZARIY QISM

Ko'rsatkichlar. Qiymatlari adres bo'lgan o'zgaruvchilarga *ko'rsatkich o'zgaruvchilar* deyiladi. Ko'rsatkich uch xil turda bo'lishi mumkin:

- 1) birorta obyektga xususan o'zgaruvchiga ko'rsatkich;
- 2) funksiyaga ko'rsatkich;
- 3) void ko'rsatkich;

Funksiyaga ko'rsatkich. Funksiyaga ko'rsatkich programma joylashgan xotiradagi funksiya kodining boshlang'ich adresini ko'rsatadi, ya'ni funksiya chaqirilganda boshqaruv ayni shu adresga uzatiladi. Funksiyani boshqa funksiyaga argument sifatida uzatish ham funksiya ko'rsatkichi orqali amalga oshiriladi. Funksiyaga ko'rsatkichning yozilish sintaksisi quyidagicha:

`<tur> (<*<nom>)(<parametrlar ro'yxati>);`

Masalan, `int *Funk(int a, int b)`

Obyektga ko'rsatkich. Obyektga ko'rsatkich quyidagicha e'lon qilinadi:

`<tur> *<nom>`

Misol:

`int *x, j, *y;`

`float *a, b, c;`

void ko'rsatkich. Bu ko'rsatkich obyekt turi oldindan noma'lum bo'lganda ishlatiladi. Void ko'rsatkichining muhim afzalliklaridan biri – unga har qanday turdagi ko'rsatkich qiymatini yuklash mumkinligidir. Void ko'rsatkich adresidagi qiymatni ishlatishdan oldin, uni aniq bir turga oshkor ravishda keltirish kerak bo'ladi. Void ko'rsatkichni e'lon qilish quyidagicha bo'ladi.

`void *<nom>`

Ko'rsatkichga boshlang'ich qiymat berish.

Boshlang'ich qiymatlar quyidagi usullar orqali berilishi mumkin:

1. *Ko'rsatkichga mavjud bo'lgan obyektning adresini berish:*
 - a) Adresni olish amali orqali:

`int a = 5, k = 4; //butun o'zgaruvchilar`

`int *p = &a; //p ko'rsatkichga a o'zgaruvchining adresi yoziladi`

```
int *p1(&k); //p1 ko'rsatkichga k o'zgaruvchining adresi yoziladi
```

b) Boshqa, initsializatsiyalangan ko'rsatkich qiymatini berish

```
int a = 5;  
int *p = &a; //p ko'rsatkichga a o'zgaruvchining adresi yoziladi  
int *r = p; // p oldin e'lon qilingan va qiymatga bo'lgan ko'rsatkich
```

c) Massiv yoki funksiya nomini berish

```
int A[10]; //Massivni e'lon qilish  
int *t = A; //Massivning boshlang'ich adresini berish;  
void A1(int a); //A1 nomli funksiyani aniqlash  
void *B(int) //Funksiyaga ko'rsatkichni e'lon qilish  
B = A; //Funksiya adresini ko'rsatkichga berish
```

2. Oshkor ravishda xotiraning absolut adresini berish

```
char *a = (char*)0xB8000000;
```

3. Bo'sh qiymat berish:

```
int *a = NULL;  
int *b = 0;
```

4. Dinamik xotirada new amali bilan joy ajratish va uni adresini ko'rsatkichga berish.

```
int *n = new int //ko'rsatkichning o'zi uchun joy kompilyatsiya vaqtida ajratiladi  
int *m = new int(10) // joy ajratishdan tashqari m adresiga boshlang'ich qiymat 10  
sonini joylashtiradi  
int *q = new int[5] //5 element uchun joy ajratilgan
```

Xotira *new* amali bilan ajratilgan bo'lsa, u *delete* amali bilan bo'shatilishi kerak. Yuqoridagi dinamik o'zgaruvchilar bilan bog'langan xotira quyidagicha bo'shatiladi:

```
delete n; delete m; delete[]q;
```

Adresni olish amali

Adresni olish quyidagicha e'lon qilinadi:

```
<tur>&<nom>
```

Ko'rsatkichlar va adres oluvchi o'zgaruvchilar funksiya parametri sifatida

Funksiya prototipida yoki aniqlanish sarlavhasida ko'rsatilgan parametrlar *formal parametrlar* deyiladi, funksiya chaqirishida ko'rsatilgan argumentlarga *faktik parametrlar* deyiladi. Faktik parametrlarni funksiyaga ikki xil usul bilan uzatish mumkin: qiymati yoki adresi bilan.

Agar parametr adres bilan uzatilsa, stekka adres nusxasi yoziladi va xuddi shu adres bo'yicha qiymatlar o'qiladi. Funksiya o'z ishini tugatgandan keyin shu adres bo'yicha qilingan o'zgarishlar saqlanib qolinadi va bu qiymatlarni boshqa funksiyalar ishlatishi mumkin.

Misol:

```
#include <iostream>
using namespace std;

void A(int a; int *b; int &c)
{
    a++;
    (*b)++;
    c++;
    *b = a + c;
    c = *b + a;
}

int main()
{
    int a = 1, b = 2, c = 3;
    cout << a << " " << b << " " << c << endl;
    A(a, &b, c);
    cout << a << " " << b << " " << c << endl;
    return (0);
}
```

5.1. O'zgaruvchining xotirada qaysi adresda joylashganini ko'rsating.

```
#include <iostream>

using namespace std;
```

```

void main()
{
    int n = 5;
    int *a;
    //& adres olish amali
    a = &n;
    cout << "n=" << n << endl;
    *a = 15;
    cout << "n=" << n << endl;
    cout << "Ko'rsatkich qiymati, ya'ni ko'rsatkich ko'rsatayotgan adres=" << a
<< endl;
    cout << "Ko'rsatkich ko'rsatayotgan adres qiymati=" << *a << endl;
    return;
}

```

5.2. Ko'rsatkichlar bilan ishlashga doir yana misol.

```

#include <iostream>

using namespace std;
void main()
{
    double n = 5;
    double *kptr;
    //& adres olish amali
    kptr = &n;
    cout << "O'zgaruvchilarning qiymati:" << endl;
    cout << "n="; cin >> n;
    cout << "*kptr="; cin >> *kptr;
    cout << "Xotira adresi:" << endl;
    cout << "n o'zgaruvchi joylashgan adres: &n=" << &n << endl;
    cout << "Ko'rsatkich ko'rsatayotgan adres. kptr=" << kptr << endl;
    cout << "Ko'rsatkich joylashgan adres. &kptr=" << &kptr << endl;
    return;
}

```

5.3. n natural son berilgan. 2 ning n-darajasini hisoblovchi funksiya hosil qiling va undan foydalaning.

```

#include <iostream>

using namespace std;
void Daraja(int n, int*S) //S ko'rsatkich sifatida e'lon qilinmoqda
{
    for (int i = 1; i <= n; i++)
        *S *= 2;
}
void main()
{
    int n = 3, S = 1; //S ga bir qiymat ta'minlamoqda
    Daraja(n, &S); //Sning adresi Daraja nomli funksiyadan chaqirib olinmoqda
    cout << "Natija: S=" << S << endl;
    return;
}

```

**5.4. a va b sonlarning EKUBini funksiya yordamida tuzuvchi dastur tuzing.
Funksiya ko'rinishini 3 xil usulda ishlab chiqing**

```

#include <iostream>
using namespace std;
void EKUBT(int a, int b, int*EKUB) //1-usul
{
    while (a != b)
    {
        if (a > b)
        {
            a = a - b;
            *EKUB = a;
        }
        else
        {
            b = b - a;
            *EKUB = b;
        }
    }
}
void main()
{
    int a, b, EKUB;
}

```

```

    cout << "a="; cin >> a;
    cout << "b="; cin >> b;

    EKUBT(a, b, &EKUB);
    cout << "Natija: EKUB=" << EKUB << endl;
    return;
}

#include <iostream>
using namespace std;
int EKUBT(int *a, int *b) //2-usul
{
    int EKUB;
    while (*a != *b)
    {
        if (*a > *b)
        {
            *a = *a - *b;
            EKUB = *a;
        }
        else
        {
            *b = *b - *a;
            EKUB = *b;
        }
    }
    return EKUB;
}

void main()
{
    int a, b, EKUB;
    cout << "a="; cin >> a;
    cout << "b="; cin >> b;

    EKUB = EKUBT(&a, &b);
    cout << "Natija: EKUB=" << EKUB << endl;
    return;
}

```

5.6. Kiritilgan K butun musbat sonning chap tarafiga R raqamini qo'shuvchi funksiya hosil qiling.

```
#include <iostream>
using namespace std;
void AddLeft(int *K, int R)
{
    int r = 0;
    int K1 = *K;
    while (K1 != 0)
    {
        K1 = K1 / 10;
        r++;
    }
    *K += R*pow(10, r);
}
void main()
{
    int K = 10, R = 5;
    AddLeft(&K, R);
    cout << K << endl;
    return;
}
```