

Пренесување на вредност на функција преку вредносен и референцен параметар

При повикување функција, аргументите во функцијата стандардно се предаваат по вредност – ваков начин на повикување се нарекува **повикување по вредност**. Бидејќи параметрите на функцијата претставуваат локални променливи (се креираат со повикување на функцијата и ги снесува кога ќе се изврши наредбата return), промените кои се прават врз параметрите не влијаат на промена на аргументите. Ова значи дека не постои начин на кој функцијата може да ги промени вредностите на аргументите. Така, на пр. не е можно да се направи функција за замена на два броја со повикување по вредност. Променливите ќе се заменат во функцијата, а во главната програма ќе останат исти. Во ваквите случаи се користи **повикување по референца** – во функцијата се пренесува адресата на променливата, а не нејзината вредност. На тој начин, со промена на вредноста на параметрите, директно се менуваат и вредностите на аргументите. Кога е потребно аргументите да се предадат по референца, во заглавјето на функцијата, пред имињата на параметрите се става знакот **& (ampersand)**. Повикување по референца се користи кога функцијата не враќа вредност или кога е потребно да се вратат повеќе вредности. Овие функции имаат тип **void**.

Разлика

1. пренос по вредност:

- локалните параметри на функцијата се копии на оригиналните (вистинските) параметри кои се пренесуваат
- измените направени во функцијата на локалните променливи немаат влијание (не ги менуваат) на вистинските параметри

2. пренос по референца

- локалните параметри на функцијата се референци (покажуваат) на мемориската локација на оригиналните (вистинските) параметри кои се пренесуваат
- измените на овие променливи во функцијата влијаат (ги менуваат) вистинските параметри
- Не се прави копија на променливите, но памети каде и кога е складиран податокот така што се заштедува на време и простор.

Пр. замена на вредности на две променливи

```
# include <iostream>
using namespace std;
// prototip na Функција за замена на вредностите на две променливи
void zameni(int &x, int&y);

int main()
{
int a,b;
cout<<"vnesi dva broja"<<endl;
cin>>a>>b;
zameni(a, b);
cout<<"novite vrednosti se "<<a<<" "<<b;
return 0;
}
void zameni(int &x, int &y) // x i y se parametri
{
```

```
int t;
t=x; // vrednosta na x se stava vo privremena promenliva t
x=y; // vrednosta na y se stava vo promenlivata x
y=t; // vrednosta na t se stava vo promenlivata y
}
```

Пр. подредување на два броја по големина

```
#include <iostream>
using namespace std;
void zamena(float &x, float &y);

int main()
{
    float a, b;

    cout << "внеси два броја " << endl;
    cin >> a >> b;
    if(a>b)
        zamena(a,b);
    cout << "podredeni broevite se : ";
    cout << a << " " << b << endl;
    return 0;
}
```

```
void zamena(float &x, float &y)
{
    float temp;

    temp = x;
    x = y;
    y = temp;
}
```

Пр. подредување на три броја по големина

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

void podredi(float &x, float &y)
{
    float temp;

    temp = x;
    x = y; y = temp;
}
```

```
int main()
{
    float a, b, c;
```

```

cout << "Vnesi tri broja : " << endl;
cin >> a >> b >> c;
if(a>b)
    podredi (a,b);
if(a>c)
    podredi (a,c);
if(b>c)
    podredi (b,c);
cout << "Подредени броевите се : ";
cout << a << " " << b << " " << c << endl;
return 0;
}

```

Пр. Напиши функција за наоѓање обратен број. Во главната програма да се најдат спротивни броеви на збирот и на разликата на броевите a и b.

```

#include <iostream>
using namespace std;
int obraten(int n)
{
    int obr=0, ost;
    while(n!=0)
    {
        ost=n%10;
        obr=obr*10+ost;
        n/=10;
    }
    return obr;
}
int main()
{
    int a,b,a1,b1,raz,z;
    cout << "Vnesi dva celi brja" << endl;
    cin>>a>>b;
    raz=a-b;
    z=a+b;
    a1=obraten(raz);
    b1=obraten(z);
    cout<<"obraten na razikata e "<<a1<<endl;
    cout<<"obraten na zbirot e "<<b1;
    return 0;
}

```

Пр. Напиши функција со која се одредува збирот на цифрите на двоцифрен природен број. Во главната програма да се отпечатат сите двоцифрени броеви на кои збирот на цифрите им е поголем или еднаков на 10.

```

#include <iostream>
using namespace std;
int zbir(int n);
int main()
{
    int i,z;
    for(i=10;i<=99;i++)
    {

```

```

        z=zbir(i);
        if (z>10) cout<<i<<" ";
    }
    return 0;
}
int zbir(int n)
{
    int z=0;
    z=n%10+n/10;
    return z;
}

```

Пр. Напиши функција со која се одредува збирот на цифрите на двоцифрен природен број. Во главната програма да се отпечатаат сите двоцифрени броеви на кои збирот на цифрите им е делив со 3.

```

#include <iostream>
using namespace std;
int zbir(int n);
int main()
{
    int i,z;
    for(i=10;i<=99;i++)
    {
        z=zbir(i);
        if (z%3==0) cout<<i<<" ";
    }
    return 0;
}
int zbir(int n)
{
    int z=0;
    z=n%10+n/10;
    return z;
}

```

Пр. Напиши функција со која се одредува бројот на цифрите на природен број. Во главната програма да се вчитаат два природни броја и да се провери дали тие имаат ист број на цифри.

```

#include <iostream>
using namespace std;
int broj(int n);
int main()
{
    int m,n,b1,b2;
    cout << "vnesi dva celi broja" << endl;
    cin>>m>>n;
    b1=broj(m);
    b2=broj(n);
    if (b1==b2) cout<<"da";
}

```

```

else cout<<"ne";
return 0;
}
int broj (int n)
{
int b=0, ost;
while(n!=0)
{
ost=n%10;
b++;
n/=10;
}
return b;
}

```

Пр. Напиши функција со која се проверува дали еден број се дели со друг број. Во главната програма да се отпечатат сите двоцифрени броеви деливи со бројот k.

```

#include <iostream>
using namespace std;
int deliv(int m,int n)
{
int t=0,b;
t=m%n;
if (t==0) b=0;
else b=1;
return b;
}
int main()
{
int k,i,p;
cout << "Vnesi go k" << endl;
cin>>k;
for(i=10;i<=99;i++)
{
p=deliv(i,k);
if(p==0) cout<<i<<" ";
}
return 0;
}

```

Пр. Напиши функција со која се проверува дали еден број се дели со друг број. Во главната програма да се одредат сите делители на дадениот природен број.

```

#include <iostream>
using namespace std;
int deliv(int m,int n)
{
int t=0,b;

```

```

    t=m%n;
    if (t==0) b=0;
    else b=1;
    return b;
}
int main()
{
    int k,i,p;
    cout << "Vnesi go k" << endl;
    cin>>k;
    for(i=1;i<=k;i++)
    {
        p=deliv(k,i);
        if(p==0) cout<<i<<" ";
    }
    return 0;
}

```

Пр. *Дадениот природен број да се разложи на прости множители со користење функција со која се проверува дали еден број се дели со друг број.

```

#include <iostream>
using namespace std;
int deliv(int m,int n)
{
    int t=0,b;
    t=m%n;
    if (t==0) b=0;
    else b=1;
    return b;
}
int main()
{
    int k,i=2,p;
    cout << "Vnesi go k" << endl;
    cin>>k;
    while(k!=1)
    {
        p=deliv(k,i);
        if(p==0)
        {
            cout<<i<<" ";
            k=k/i;
        }
        else i++;
    }
    return 0;
}

```

```
}
```

пр. *Да се отпечатат сите прости броеви од 1 до n со користење на функцијата prost.

```
#include <iostream>
using namespace std;
bool prost(int n)
{
    int b=0,j;
    for(j=2;j<=n/2;j++)
    {
        if(n%j==0) b++;
    }
    if(b==0) return true;
    else return false;
}
int main()
{
    int n,i;
    bool t;
    cout << "vnesi go n" << endl;
    cin>>n;
    for(i=2;i<=n;i++)
    {
        t=prost(i);
        if (t) cout<<i<<" e prost " <<endl;
    }
    return 0;
}
```

Пр. Напиши функција (од типот void) izmeni која како аргументи има референци на 3 цели броеви. Во процедурата, првиот аргумент се зголемува за 1, вториот аргумент се множи по 2, и третиот се намалува за 1. Потоа во главната програма да се декларираат 3 целобројни променливи x, y и z и да се внесат нивните вредности преку тастатура. Да се повика процедурата со нив како аргументи. На крајот во главната програма да се отпечатат нивните нови вредности.

Пример: x=3, y=2, z=4;
Излез: 4, 4, 3; (3+1=4, 2*2=4, 4-1=3)

```
#include <iostream>
using namespace std;
void izmeni(int &x, int &y, int &z)
{
    x++;
    y*=2;
    z--;
}
int main()
{
    int x,y,z;
    cout << "vnesi tri celi broja" << endl;
```

```
cin>>x>>y>>z;  
izmeni(x,y,z);  
cout<<x<<" "<<y<<" "<<z;  
return 0;  
}
```