

ЗАДАЧИ С ИЗПОЛЗВАНЕ НА ФУНКЦИИ

Пример 1

Да се напише програма, която чрез три функции:

- въвежда три положителни цели числа от клавиатура;
- намира най-голямото от тях (max);
- определя $\sqrt{\text{max}}/x_1$, където x_1 е първото въведено число.

```
#include<iostream.h>
#include<math.h>
int getnum(); //деклариране на функциите
int getmax(int n1,int n2,int n3);
double calcul(int m,int n);
int max_n; //глобална променлива
int main() //главна функция
{int num1,num2,num3; //локални променливи
  num1=getnum();
  num2=getnum();
  num3=getnum();
  max_n=getmax(num1,num2,num3);
  cout<<"\n Най-голямото число е: "<<max_n;
  cout<<"\n Резултат от изчисленията: "<<calcul(max_n,num1);
  return 0;
}

int getnum() //дефиниция на функциите
{
  int num;
  do
  {cout<<"\n Въведете число: ";
   cin>>num;
  }
  while (num<=0);
  return num;
}

int getmax(int n1,int n2,int n3)
{
  max_n=n1;
  if (n2>max_n) max_n=n2;
  if (n3>max_n) max_n=n3;
  return max_n;
}

double calcul(int m, int n)
{
  return sqrt(m)/n;
}
```

Пример 2

Да се изчисли с точност $\varepsilon=10^{-4}$ стойността на функцията $s=\cos(x)$, използвайки разложението на косинуса в ред:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots = \sum_{n=0}^{\infty} t_n(x), \quad t_n(x) = (-1)^n \frac{x^{2n}}{(2n)!}$$

```

#include <iostream.h>
#include <math.h>
const double EPS=1e-4;
int n;
double s=0.,t,f,x,y;
//изчисляване на np!
int factor(int np)
{ for (int fac=1,i=1;i<=np;i++)
    fac*=i;
  return fac;
}
//изчисляване на общия член t
double fun(int n,double x)
{ return pow(-1,n)*pow(x,2*n)/factor(2*n);
}
void main()
{ cout<<"\n Въведете число x = ";
  cin>>x;
  t=n=1;
  while (fabs(t)>EPS)
    { s+=t;
      t=fun(n,x);
      n++;
    }
  y=cos(x); //за проверка
  cout<<"\n Приблизителна стойност cos(x): "<<s;
  cout<<"\n Точна стойност cos(x): "<<y;
}

```

Пример 3

```

//Програма за определяне на периметъра и лицето на триъгълник
// по зададени три страни посредством една функция
#include <iostream.h>
#include <math.h>
void triang(float x,float y,float z,float*s1,float *p1)
{ float pt;
  *p1=x+y+z; //периметър
  pt=*p1/2.0; //полупериметър
  *s1=sqrt(pt*(pt-x)*(pt-y)*(pt-z)); //лице
} //край на функцията triang
void main()
{float a,b,c,s,p;
  err:cout<<"\n Въведете страните на триъгълника -> ";
    cout<<"\n Страна a = "; cin>>a;
    cout<<"\n Страна b = "; cin>>b;
    cout<<"\n Страна c = "; cin>>c;
    if((a<=0)|| (b<=0)|| (c<=0)|| (a+b<=c)|| (a+c<=b)|| (b+c<=a))
      { cout<<"\n Грешно въведена страна!";
        goto err;
      }
  triang(a,b,c,&s,&p);
  cout<<"\n Лицето на триъгълника е "<<s;
  cout<<"\n Периметъра на триъгълника е "<<p;
}

```

Пример 4

```
//Програма за размяна стойностите на две различни цели числа
#include <iostream.h>
void change(int& a,int& b)
{ int t;
  t=a; a=b; b=t; //няма * както при указателите
}
int main()
{ int x,y;
  cout<<"\n x = "; cin>>x;
  do {cout<<"\n y = ";
      cin>>y;}
  while (x==y);
  change(x,y);
  cout<<"\n Резултати след размяната:";
  cout<<"\n x = "<<x<<"\t y = "<<y;
  return 0;
}
```

Пример 5

Да се напише програма, която намира броя на отрицателните елементи на един едномерен масив.

```
#include<iostream.h>
int i;
int BROI(float b[],int b_size)
{ int br=0;
  for (i=0;i<b_size;i++)
    if (b[i]<0) br++;
  return br;
}
void main()
{ int n;
  float a[25];
  do
  { cout<<"\n Брой елементи на масива: ";
    cin>>n;
  } while (n<2||n>25);
  for (i=0;i<n;i++)
  { cout<<"\n Елемент a["<<i<<"]=" ";
    cin>>a[i];}
  cout<<"\n Брой отрицателни елементи: "<<BROI(a,n);
}
```

Пример 6

Да се състави програма, която с използване на две функции намира:

- сумата от всички нечетни елементи на един едномерен целочислен масив с положителни елементи;
- максималният елемент на същия масив и неговия индекс.

```
#include<iostream.h>
#define BR 20
int i;
int suma(int n,int *pa)
//pa - указател към началото на масива
//n - брой елементи на масива
```

```

{   int s=0;
    for (i=0;i<n;i++,pa++)
        if (*pa%2!=0)  s+=*pa;
    return s;
}
int max(int n,int *pa, int *index)
//pa - указател към началото на масива
//index - индекс на максималния елемент
{   int max_el=0;
    *index=0;
    for (i=0;i<n;i++,pa++)
        if (*pa>max_el) { max_el=*pa; *index=i;}
    return max_el;
}
void main()
{int na,B[BR],main_index;
  do
    {   cout<<"\n Въведете брой елементи: ";
        cin>>na;
    } while (na<2||na>BR);
  for (i=0;i<na;i++)
    {   cout<<"\n Елемент B["<<i<<" ] = ";
        cin>>B[i];
    }
  cout<<"\n Сума нечетни елементи: "<<suma(na,B);
  cout<<"\n Макс.елемент: "<<max(na,B,&main_index);
  //пред main_index се поставя &, т.к.стойността
  //се изчислява в max
  cout<<"\n Индекс на макс.елемент: "<<main_index;
}

```

Пример 7

```

#include <iostream.h>
//функция fact за изчисляване на n!
//типът е long с цел избягване на препълване
long fact(int n)
{   if (n==0) return 1;
    return n*fact(n-1);
}
void main()
{   int n;
    do
        {   cout<<"\n Въведете n = ";
            cin>>n;
        } while (n<0);
    cout<<"\n n! (n факториел) - "<<fact(n);
}

```