



Функции (продолжение)

Лекция 8 (дополнение)

Пример 1

Пример. Составить программу, которая в массивах $M[10]$ и $L[20]$ определяет максимальный нечетный элемент, используя функцию.

Пример 1 (продолжение)

```
#include "stdafx.h"
```

```
#include "locale.h"
```

```
#include "conio.h"
```

```
#include <iostream>
```

```
using namespace std;
```

```
//Объявление функции поиска максимума
```

```
int max(const int* mas, const int n);
```

Пример 1 (продолжение)

```
//Главная функция
int _tmain(int argc, _TCHAR* argv[])
{
    setlocale(LC_ALL, "");
    int M[20];
    int L[10];
```

Пример 1 (продолжение)

//Ввод массива M

```
cout<<"Введите массив M[20]\n";
for (int i=0; i<20; i++)
{
    cout<<"Введите M["<<i<<"]="";
    cin>>M[i];
}
```

//Ввод массива L

```
cout<<"Введите массив L[10]\n";
for (int i=0; i<10; i++)
{
    cout<<"Введите L["<<i<<"]="";
    cin>>L[i];
}
```

Пример 1 (продолжение)

```
//Вывод на экран массива M
```

```
cout<<"\nМассив M[20]\n";
```

```
for (int i=0; i<20; i++)
```

```
    cout<<M[i]<<" ";
```

```
//Вывод на экран максимального элемента среди
```

```
//нечетных путем вызова функции max
```

```
cout<<"\nМаксимальный элемент массива M[20]
```

```
среди нечетных = "<<max(M,20);
```

Пример 1 (продолжение)

```
//Вывод на экран массива L
```

```
cout<<"\nМассив L[10]\n";
```

```
for (int i=0; i<10; i++)
```

```
cout<<L[i]<<" ";
```

```
//Вывод на экран максимального элемента среди
```

```
//нечетных путем вызова функции max
```

```
cout<<"\nМаксимальный элемент массива L[10]
```

```
среди нечетных = "<<max(L,10);
```

```
getch();
```

```
return 0;
```

```
}
```

Пример 1 (продолжение)

```
//Описание функции max, которая находит
//максимальный среди нечетных элементов массива
int max(const int* mas, const int n)
{
    int k=0, maxn;
    for (int i=0; i<n; i++)
        if (mas[i]%2!=0)
            if (k==0)
            {
                k=1;
                maxn=mas[i];
            }
}
```


Пример 1 (продолжение)

```
    else
        if (maxn < mas[i])
            maxn = mas[i];
    return maxn;
}
```

Пример 2

Пример. Составить программу, которая в матрицах $A[2,3]$ и $B[3,4]$ определяет произведение и сумму кратных 7 элементов, используя функцию.

Пример 2 (продолжение)

```
#include "stdafx.h"  
#include "stdlib.h"  
#include "locale.h"  
#include "conio.h"  
#include <iostream>  
using namespace std;
```

Пример 2 (продолжение)

//Функция ввода элементов массива

```
void vvod(int *mas, const int nstr, const int nstb, const char sym);
```

//Функция вывода элементов массива на экран

```
void vivod(const int *mas, const int nstr, const int nstb, const char sym);
```

//Функция поиска в массиве произведения и суммы

//элементов кратных 7

```
void poisk(const int *mas, const int nstr, const int nstb, const char sym);
```

Пример 2 (продолжение)

```
//Главная функция main
int _tmain(int argc, _TCHAR* argv[])
{
    setlocale(LC_ALL, "");
//Объявление массивов А и В
    int A[2][3];
    int B[3][4];
```

Пример 2 (продолжение)

```
//Вызов функций ввода, поиска и вывода элементов
//массивов A и B
    vvod(&A[0][0],2,3,'A');
    vvod(&B[0][0],3,4,'B');
    vivod(&A[0][0],2,3,'A');
    poisk(&A[0][0],2,3,'A');
    vivod(&B[0][0],3,4,'B');
    poisk(&B[0][0],3,4,'B');
    getch();
    return 0;
}
```

Пример 2 (продолжение)

```
//Описание функции vvod
```

```
void vvod(int *mas, const int nstr, const int nstb, const  
char sym)  
{  
    cout<<"\nМассив  
"<<sym<<"["<<nstr<<"]["<<nstb<<"]\n";  
    for (int i=0; i<nstr; i++)  
        for (int j=0; j<nstb; j++)  
            {  
                cout<<"Введите "<<sym<<"["<<i<<"]["<<j<<"]="";  
                cin>>mas[i * nstb + j];  
            }  
}
```

Пример 2 (продолжение)

```
//Описание функции vivod
void vivod(const int *mas, const int nstr, const int nstb,
           const char sym)
{
    cout<<"\nМассив
    "<<sym<<"["<<nstr<<"]["<<nstb<<"]\n";
    for (int i=0; i<nstr; i++)
    {
        for (int j=0; j<nstb; j++)
            printf("%3d ",mas[i * nstb + j]);
        printf("\n");
    }
}
```


Пример 2 (продолжение)

//Описание функции poisk

```
void poisk(const int *mas, const int nstr, const int nstb,  
           const char sym)
```

```
{
```

```
    int sum=0, proizv=1;
```

```
    for (int i=0; i<nstr; i++)
```

```
        for (int j=0; j<nstb; j++)
```

```
            if (mas[i * nstb + j]%7==0)
```

```
                {
```

```
                    sum=sum + mas[i * nstb + j];
```

```
                    proizv=proizv * mas[i * nstb + j];
```

```
                }
```

Пример 2 (продолжение)

```
cout<<"\nСумма элементов массива "<<sum<<"  
кратных 7 = "<<sum;  
cout<<"\nПроизведение элементов массива  
<<sum<<" кратных 7 = "<<proizv<<"\n";  
}
```