

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Программирование»**  
**Тема: Условия, циклы, оператор switch**

Студент гр. 6304

\_\_\_\_\_

Курков Д. В.

Преподаватель

\_\_\_\_\_

Берленко. Т. А.

Санкт-Петербург

2016

## Цель работы.

Научиться использовать условия, циклы, оператор switch.

## Задание.

Напишите функцию-меню, на вход которой может подаваться число  $n$ , одно из значений 0, 1, 2, 3 и массив целых чисел размера  $n$ .

В зависимости от аргумента, функция должна выводить следующее:

0 : максимальное число в массиве

1 : минимальное число в массиве

2 : сумму всех четных элементов

3 : число повторов нулевого элемента в массиве (не включая в это количество сам элемент)

иначе необходимо вывести строку "Данные некорректны"

Для реализации меню используйте оператор множественного выбора switch.

## Ход работы

### 1. Создание функций для обработки вводимых данных

#### 1. Функция, возвращающая максимальное число в массиве

##### Get\_max.c

```
int get_max (int arr [], int n)
{
    int i = 0;
    int a = arr [i++];
    while (i < n)
    {
        if (arr[i] >= a)
            a = arr[i];
        ++i;
    }
    return a;
}
```

#### 2. Функция, возвращающая минимальное число в массиве

##### Get\_min.c

```
int get_min (int arr [], int n)
{
    int i = 0;
    int a = arr [i++];
    while (i < n)
    {
        if (arr[i] <= a)
            a = arr [i];
        ++i;
    }
    return a;
}
```

#### 3. Функция, возвращающая сумму всех четных элементов

##### Get\_sum.c

```
int get_sum (int arr[], int n)
{
    int i = 0, a = 0;
    while (i < n)
    {
        if ((arr[i]%2) == 0)
            a+=arr[i];
        ++i;
    }
    return a;
}
```

4. Функция, возвращающая число повторов нулевого элемента в массиве

**Get\_count\_first\_el.c**

```
int get_count_first_el (int arr[], int n)
{
    int i = 0, a = 0;
    int s = arr [i++];
    while (i < n)
    {
        if (arr [i] == s)
            ++a;
        ++i;
    }
    return a;
}
```

2. Создание функции-интерфейса, для работы с вышеописанными функциями.

**Menu.c**

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

int get_max (int arr [], int n);
int get_min (int arr [], int n);
int get_sum (int arr [], int n);
int get_count_first_el (int arr [], int n);

void menu (int ch, int arr[], int n)
{
    int a;
    switch (ch)
    {
        case 0:
            a = get_max (arr, n);
            break;
        case 1:
            a = get_min (arr, n);
            break;
        case 2:
            a = get_sum (arr, n);
            break;
        case 3:
            a = get_count_first_el (arr, n);
            break;
        default:
            printf ("Данные не корректны\n");
            exit (1);
    }
    printf ("%d\n", a);
}
```

### 3. Создание функции main.

#### Main.c

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

void menu (int ch, int arr[], int n);

int main ()
{
    int ch, n = 20;
    int arr [n];
    srand(time(NULL));
    for (int i=0;i<n;i++)
        arr[i] = (rand()%1000 + 1);
    printf("ARR - ");
    for (int i=0;i<n;i++)
        printf("%d ", arr[i]);
    printf ("\nChoice - ");
    scanf ("%d", &ch);
    menu (ch, arr, n);
    return 0;
}
```

### 4. Создание и описание makefile, призванного скомпоновать вышеперечисленные функции и создать исполняемый файл

#### Makefile

all: clean program

clean: program  
rm \*.o

program: main.o menu.o get\_max.o get\_min.o get\_sum.o get\_count\_first\_el.o  
gcc -o program main.o menu.o get\_max.o get\_min.o get\_sum.o get\_count\_first\_el.o

main.o: main.c  
gcc -c main.c

menu.o: menu.c  
gcc -c menu.c

get\_max.o: get\_max.c  
gcc -c get\_max.c

get\_min.o: get\_min.c  
gcc -c get\_min.c

get\_sum.o: get\_sum.c  
gcc -c get\_sum.c

## **Вывод**

В ходе лабораторной работы было изучены и применены на практике такие средства языка C, как условия, циклы и оператор switch.