

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

ОТЧЕТ
по лабораторной работе №1304
по дисциплине «Программирование»
Тема: Условия, циклы, оператор switch

Студент гр.

Павлов Д.Р.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

Цель работы.

Исследование управляющих конструкция языка Си, изучение основных способом написания программы и начало изучения функционального программирования.

Задание.

II Вариант

Напишите программу, выделив каждую подзадачу в отдельную функцию.

Реализуйте программу, на вход которой подается одно из значений 0, 1, 2, 3 и массив целых чисел размера не больше 100. Числа разделены пробелами.

Строка заканчивается символом перевода строки.

В зависимости от значения, функция должна выводить следующее:

0 : максимальное число в массиве. (max)

1 : минимальное число в массиве. (min)

2 : разницу между максимальным и минимальным элементом. (diff)

3 : сумму элементов массива, расположенных до первого минимального элемента. (sum)

Иначе необходимо вывести строку "Данные не корректны".

Выполнение работы.

В данной лабораторной работе я использую стандартную библиотеку `stdio.h`. Для того, чтобы сделать свой код более читаемым, я обозначаю максимальный буффер в будущем массиве.

```
#define MAX_BUFFER 101
```

Далее, я объявляю 4 опциональные(*зависящие от нулевого элемента массива) функции: `min` (*нахождение минимального элемента массива);`max`(*нахождение максимального элемента массива);`diff`(*нахождение

разности между максимальный и минимальным элементами массива);sum(*сумма элементов до минимального элемента массива)

1) Функция min

```
int min(int array[], int I){
    int minimum = 1000000;
    for (int i = 1; i < I; i++){
        if (minimum > array[i]){
            minimum = array[i];
        }
    }
    return minimum;
}
```

сама функция имеет тип `int`, поскольку она возвращает число, и два аргумента — `int array[]` (*массив) и `int I` (*размер массива). В данной функции я инициализирую переменную `minimum` (*равную какому-то большому значению), которая будет означать минимального число. Потом я делаю цикл `for(int i = 1; i < I; i++)`, который будет пробегаться по всем элементам массива. Важно заметить, что я беру `i = 1`, вместо привычного `0` поскольку нулевой элемент мы не считаем. Далее я задаю условие, которое будет сравнивать переменную `minimum` и `i`-ый элемент массива; если `minimum` больше чем элемент, то тогда присваиваем `minimum` значение `i`-го элемента. В конце функции просто возвращаем значение `minimum`.

2) Функция max

функция `max` почти ничем не отличается от функции `min`. Единственные различия — переменную `minimum` сменяет переменная `maximum`, которая уже равна `0`; условие внутри цикла меняет свой знак на противоположный.

```

int max(int array[], int I){
    int maximum = 0;
    for (int i = 1; i < I; i++){
        if (maximum < array[i]){
            maximum = array[i];
        }
    }
    return maximum;
}

```

3) Функция diff

функция diff, так же как и предыдущие, имеет тип `int` и два такие же аргументы типа `int` — `array[]` и `I` (*массив и размер массива). Далее мы инициализируем переменные типа `int` — `difference`, `maximum`, `minimum`. Где последние две переменные являются возвратом уже известных нам функций(*`max` и `min`), беря за аргументы все те же `arr` и `I`, а `difference` — разностью `maximum` и `minimum`. Именно `difference` мы и будем возвращать.

```

int diff(int array[], int I){
    int difference, maximum, minimum;
    maximum = max(array, I);
    minimum = min(array, I);
    difference = maximum - minimum;
    return difference;
}

```

4) Функция sum

Далее идет последняя функция, зависящая от нулевого элемента всего массива чисел. Она так же как имеет тип `int` и аргументы `array[]` и `I`. Инициализируем переменную `minimum` типом `int`, которая будет равна вызываемому значению функции `min` от все тех же `array` и `I`. Далее инициализируем переменную `total` типом `int` равному нулю и делаем знакомый по функциям `min` и `max` цикл `for (int i = 1; i < I; i++)`, но в этот раз внутри цикла ставим условие сравнения `i`го элемента массива на неравенство с минимальным значением этого массива — если `i`ый элемент не равен минимальному, то мы

прибавляем к переменной `total` `i`-ый элемент; в противном случае завершаем цикл. Далее после завершения цикла мы возвращаем `total`

```
int sum(int array[], int I){
    int minimum = min(array, I);
    int total = 0;
    for (int i = 1; i < I; i++){
        if (array[i] != minimum){
            total += array[i];
        }
        else{
            break;
        }
    }
    return total;
}
```

Далее мы объявляем функцию `main` типа `int`. Она выполняет основные действия программы. В ней мы инициализируем массив целых чисел `arr`, размер которого будет максимальным(*таким образом пока в нашем массиве храниться `*MAX_BUFFER` мусора); далее мы инициализируем переменную `c` типа `char`(*она нам нужна чтобы остановить заполнение массива в будущем); и — переменную `size` типа `int` равному нулю. Далее пишем цикл `for(int i = 0; i < MAX_BUFFER; i++)` (*но в этот раз мы начинаем цикл с нулевого элемента, поскольку он нам будет нужен для вызова конкретной функции). Внутри цикла мы считываем элементы, которые вводятся пользователем и заносим их в массив `arr` и увеличиваем размер(`size`) на 1. Важно отметить, что мы считываем элементы как типа `int`, так и `char`. Это делается для того, чтобы мы смогли остановить ввод значений в условии `if (c == „\n“)`, которая означает, что если мы ввели символ перевода строки(Enter a.k.a `\n`), то цикл останавливается. Далее мы вводим оператор `switch`, который будет зависеть от нулевого элемента массива. В случае, если нулевой элемент равен числу от 0 до 3 включительно, мы вызываем одну из функций(0 — `max()`; 1 — `min`; 2 — `diff`; 3

— `sum()`), но если нулевой элемент не равен этим значениям, мы выводим «Данные некорректны». Прошу заметить, что в конце каждого `case` мы пишем `break`. Это нужно для того, чтобы после вызова функции оператор `switch` прекращал работу. Так же стоит отметить, что в функции `printf()` перед форматированием мы пишем `\n`, дабы избежать «слипание» строк. В конце программы мы пишем `return 0`

```
int main(){
    int arr[MAX_BUFFER];
    char c;
    int size = 0;
    for(int i = 0; i < MAX_BUFFER; i++){
        scanf("%d%c", &arr[i], &c);
        size++;
        if (c == '\n'){
            break;
        }
    }

    switch (arr[0]) {
        case 0:
            printf("\n%d", max(arr, size));
            break;
        case 1:
            printf("\n%d", min(arr, size));
            break;
        case 2:
            printf("\n%d", diff(arr, size));
            break;
        case 3:
            printf("\n%d", sum(arr, size));
            break;
        default:
            printf("\nДанные некорректны");
    }
```

```

    }
    return 0;
}

```

Обработка результатов эксперимента.

2 10 25 34 65 70	60
3 40 50 20 30 40 10 50 70 80 100	180
0 10 20 30 40 50 60 12 53 63	63
5 24 4355 635 43 64	Данные некорректны
1 645 5 54 34 6 46 7	5

Выводы.

Благодаря данной лабораторной работе я изучил управляющие конструкции языка Си. Так же были изучены простейшие арифметические операции, операторы, массивы и циклы.

В данной лабораторной работе я разработал программу считывания с данных и для обработки их я использовал оператор switch, простейшие арифметические операции и условные операторы if

ПРОТОКОЛ

В протоколе приводятся результаты измерений, полученные в ходе выполнения экспериментальной части лабораторной работы.

Объем данных, заносимых в протокол, определяется соответствующими методическими указаниями и преподавателем.

Недопустимо заполнение протокола карандашом.

На протоколе должна быть подпись студента, выполнившего лабораторную работу (в случае, если работа выполнялась в составе бригады – всех членов бригады) и дата выполнения работы.

По завершении выполнения лабораторной работы протокол

п

о

д

п

и

с

ы

в

а

е

т

с

я

п

р

е

п

о

д

а

в