МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ имени В. И. ВЕРНАДСКОГО»

(ФГАОУ ВО «КФУ им. В. И. Вернадского»)

Таврическая академия

Физико-технический институт

ОТЧЕТ

ПО УЧЕБНОЙ ПРАКТИКЕ, проектно-технологической

обучающегося _____Стецюк Даниила Романовича_____

(фамилия, имя, отчество)

Физико-технический институт

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность (профиль)

Системное программирование и информационные технологии

Курс 2 Группа ПМиИ-б-о-202

Руководитель практики от КФУ _____ А. В. Ильченко

(подпись)

Отчет защищен с оценкой _____

Оглавление

Общая информация

В соответствии с учебным планом подготовки бакалавров по направлению 01.03.02 Прикладная математика и информатика учебная практика проходила с 8 июля по 16 июля 2022 года.

Целью учебной практики является закрепление теоретических знаний по программированию, приобретение профессиональных навыков и компетенций в области современных информационных технологий, получение опыта профессиональной деятельности в производственном коллективе.

Основные задачи учебной практики: выполнение индивидуального задания на разработку программ в среде C++.

Индивидуальное задание

Файл «Условия задач».

Темы и номера заданий:

- Функции – №19
- Одномерные массивы – №12, №34
- Двумерные массивы – №15, №32
- Структурный тип данных – №2

Тема 1. Функции

Задача №19

1. Условие задачи

Задан упорядоченный по убыванию целочисленный массив X(N). Вставить в массив X некоторое число H, сохранив упорядоченность массива. Найти среднее арифметическое простых чисел в массиве до и после вставки числа. В программе написать функции: вставки элемента, вычисления среднего арифметического простых элементов массива.

2. Словесное описание алгоритма

Память для массива выделяется динамически, с использованием операции new. Данные для массива генерируются датчиком случайных чисел.

Далее при помощи цикла каждый элемент массива передаётся в функцию Average() для проверки данного числа на простоту, поиска суммы простых чисел и их количетсва.

Функция Sorting() сортирует массив по убыванию.

Далее выводится сгенерированный отсортированный по убыванию массив и среднее арифметическое простых чисел этого массива.

Функция addElemet() служит для вставки нового случайно сгенерированного числа в массив. После чего массив снова сортируется в функции Sorting().

Далее элементы массива вновь передаются в цикле в функцию Average(), для поиска обновлённого количества простых чисел и их суммы.

Затем массив вновь выводиться на печать (после вставки нового числа) и среднее арифметическое простых числе обновлённого массива.

3. Описание интерфейса пользователя программы

После запуска программы на экран отображается подсказка с просьбой ввести количество элементов массиве.

Укажите количество элементов массива:

После ввода числа программа выполняет расчёт и выводит на экран следующую информацию:

Сгенерированный массив.
Среднее арифметическое простых чисел до вставки числа.
Элемент сгенерированный для вставки.
Среднее арифметическое простых чисел после вставки числа.

4. Код программы

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

//Прототив функции сортировки
void Sorting(int X[], int N);

//Прототив функции вставки элемента
void addElement(int X[], int N);

//Прототип функции проверки числа на простоту
int Average(int num, int &summ, int &amount);

int main() {
    int N; // Количество элементов массива
    int summ = 0, amount = 0; /*summ = сумма простых чисел, amount = количество
простых чисел*/
    srand((unsigned int) time(NULL));
    printf("Укажите количество элементов массива: ");
    scanf("%d", &N);
    int *X = new int[N + 1];
    for (int i = 0; i < N; i++) X[i] = rand() % 10; /*Заполнение псевдо
случайными числами массива*/

    for (int i = 0; i < N; i++) { /*Переход в функцию нахождения суммы простых
чисел и их среднего арифметического*/
        Average(X[i], summ, amount);
    }

    Sorting(X, N); // Сортируем массив по убыванию

    for (int i = 0; i < N; i++) printf("%d\t", X[i]);

    printf("\n");
    printf("Среднее арифметическое простых чисел до вставки числа: %.2f\n",
(float) summ / amount);

    addElement(X, N); // Добавляем элемент
    Sorting(X, N); // Сортируем массив по убыванию
    summ = 0, amount = 0;
    for (int i = 0; i < N; i++) { /*Переход в функцию нахождения суммы простых
чисел и их среднего арифметического*/
        Average(X[i], summ, amount);
    }
    for (int i = 0; i < N + 1; i++) printf("%d\t", X[i]);
    printf("\n");
    printf("Среднее арифметическое простых чисел после вставки числа: %.2f\n",
(float) summ / amount);
}

void Sorting(int X[], int N) { //Функция сортировки массива по убыванию
    for (int i = 0; i < N; i++) {
        for (int j = i + 1; j < N + 1; j++) {
            if (X[i] < X[j]) {
                int k = X[i];
                X[i] = X[j];
                X[j] = k;
            }
        }
    }
}

void addElement(int X[], int N) { //Функция вставки элемента
    X[N] = rand() % 10 + 1;
```

```
    printf("Элемент для вставки:\t%d\n", X[N]);
}

int Average(int num, int &summ, int &amount) { //Функция нахождения простых чисел
    if (num > 1) {
        for (int i = 2; i < num; i++)
            if (num % i == 0)
                return false;
        summ += num;
        amount++;
    } else
        return false;
}
```

5. Результат работы программы

```
Укажите количество элементов массива: 9
9       9       8       4       3       3       2       1       1
Среднее арифметическое простых чисел до вставки числа: 2.67
Элемент для вставки:    7
9       9       8       7       4       3       3       2       1       1
Среднее арифметическое простых чисел после вставки числа: 3.75
```

Тема 2. Одномерные массивы

Задача №12

1. Условие задачи

Дан массив целых чисел.
а) Все элементы, оканчивающиеся цифрой 4, уменьшить вдвое.
б) Все четные элементы заменить на их квадраты, а нечетные удвоить.
в) Четные элементы увеличить на a, а из элементов с четными номерами вычесть b.

2. Словесное описание алгоритма

Массив моделируется контейнером вектор.

Данные для массива генерируются датчиком случайных чисел в функции CreateArray(), после чего массив выводится на экран.

В функции TaskA() выполняются действия над исходным массивом установленные условием задачи а) – Уменьшение элементов заканчивающихся цифрой 4 вдвое. После чего результат выводится на экран.

В функции TaskB() выполняются действия над исходным массивом установленные условием задачи б) – Чётные элементы возводятся в квадрат, а нечетные удваиваются. После чего результат выводится на экран.

В функции TaskC() выполняются действия над исходным массивом установленные условием задачи в) – Чётные элементы увеличиваются на A, а из элементов с чётными номерами вычитается B. После чего результат выводится на экран.

3. Описание интерфейса пользователя программы

После запуска программы на экран отображается следующая информация:

Сгенерированные значения для A и B.
Сгенерированный массив.
Массив после выполнения задачи А.
Массив после выполнения задачи Б.
Массив после выполнения задачи В.

4. Код программы

```
#include <stdio.h>
#include <vector>
#include <stdlib.h>
#include <math.h>
#include <iostream>

using namespace std;
```

```
//Прототип функции генерации массива
void CreateArray(vector<int> &vec);

//Прототип функции уменьшающей вдвое все элементы, оканчивающиеся цифрой 4
void TaskA(vector<int> &vec);

//Прототип функции заменяющий нечётные элементы на их квадраты
void TaskB(vector<int> &vec);

//Прототип функции которая чётные элементы увеличить на A
void TaskC(vector<int> &vec, int &A, int &B);

int main() {
    srand((unsigned int) time(NULL));
    int A, B;
    vector<int> vec(rand() % 10);
    A = rand() % 10;
    B = rand() % 10 + 4;
    printf("Сгенерированны следущие значения\tA: %d\tB: %d\n", A, B);
    printf("Сгенерирован следующий массив:\t\t");
    CreateArray(vec); //Заполнение вектора
    printf("\n");
    TaskA(vec); //Задача А
    printf("\n");
    TaskB(vec); //Задача Б
    printf("\n");
    TaskC(vec, A, B); //Задача В
    printf("\n");
}

void CreateArray(vector<int> &vec) {
    for (int i = 0; i < vec.size(); i++) {
        vec.at(i) = rand() % 10 + 11;
        printf("%d\t", vec[i]);
    }
}

void TaskA(vector<int> &vec) {
    printf("Массив после выполнения задачи А:\t");
    for (int i = 0; i < vec.size(); i++) {
        if (vec.at(i) % 10 == 4) {
            printf("%d\t", vec[i] / 2);
        } else {
            printf("%d\t", vec[i]);
        }
    }
}

void TaskB(vector<int> &vec) {
    printf("Массив после выполнения задачи Б:\t");
    for (int i = 0; i < vec.size(); i++) {
        if (vec.at(i) % 2 == 0) {
            printf("%.0f\t", pow(vec[i], 2));
        } else {
            printf("%d\t", vec[i]*2);
        }
    }
}

void TaskC(vector<int> &vec, int &A, int &B) {
    printf("Массив после выполнения задачи В:\t");
    for (int i = 0; i < vec.size(); i++) {
        if (vec.at(i) % 2 == 0) {
```

```
            printf("%d\t", vec[i] + A);
        }
        else if (i % 2 == 0) {
            printf("%d\t", vec[i] - B);
        } else {
            printf("%d\t", vec[i]);
        }
    }
}
```

5. Результат работы программы

```
Сгенерированны следущие значения        A: 1    B: 11
Сгенерирован следующий массив:          14      12      19      17      13      19
Массив после выполнения задачи А:       7       12      19      17      13      19
Массив после выполнения задачи Б:       196     144     38      34      26      38
Массив после выполнения задачи В:       15      13      8       17      2       19
```

Задача №34

1. Условие задачи

Дан массив вещественных чисел. Выяснить:
а) верно ли, что количество положительных элементов не превышает 5;
б) верно ли, что количество элементов, которые не больше 50,55, кратно четырем

2. Словесное описание алгоритма

Массив моделируется контейнером вектор.

Данные для массива генерируются датчиком случайных чисел в функции CreateArray(), после чего массив выводится на экран.

В функции TaskA() выполняются действия над исходным массивом установленные условием задачи а) – Проверка что количество положительных элементов не превышает 5. После чего результат выводится на экран.

В функции TaskB() выполняются действия над исходным массивом установленные условием задачи б) – Проверка что количество элементов, которые не больше 50,55, кратно четырем. После чего результат выводится на экран.

3. Описание интерфейса пользователя программы

После запуска программы на экран отображается следующая информация:

Сгенерированный массив и результаты проверок условий задачи.

4. Код программы

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

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

using namespace std;

//Прототип функции генерации массива
void CreateArray(vector<float> &vec);

//Прототип функции проверки количества положительных элементов
void TaskA(vector<float> &vec);

//Прототип функции проверки количества элементов кратных 4
void TaskB(vector<float> &vec);

int main() {
    srand((unsigned int) time(NULL));
    vector<float> vec(rand() % 30);
    printf("Сгенерирован следующий массив:\n");
    CreateArray(vec); //Заполнение вектора
    printf("\n");
    TaskA(vec); //Задача А
    printf("\n");
    TaskB(vec); //Задача Б
}

void CreateArray(vector<float> &vec) {
    for (int i = 0; i < vec.size(); i++) {
        vec.at(i) = (float) (rand() % (10000 * vec.size())) / 1000 - 30;
        printf("%.2f\t", vec[i]);
    }
}
void TaskA(vector<float> &vec) {
    printf("Количество положительных элементов менее 5?\t");
    int times = 0;
    for (int i = 0; i < vec.size(); i++) {
        if (vec.at(i) > 0)
            times++;
    }
    if (times > 5)
        printf("Нет!");
    else
        printf("Да!");
}
void TaskB(vector<float> &vec) {
    int times = 0, times2 = 0;
    for (int i = 0; i < vec.size(); i++) {
        if (vec.at(i) < 50)
            times++;
        if (vec.at(i) < 55)
            times2++;
    }
    if (times % 4 == 0 && times != 0)
        printf("Количество элементов больших чем 50, кратно 4!\n");
    else
        printf("Количество элементов меньших чем 50, не кратно 4!\n");
    if (times2 % 4 == 0 && times != 0)
        printf("Количество элементов больших чем 55, кратно 4!");
    else
        printf("Количество элементов меньших чем 55, не кратно 4!");
}
```

5. Результат работы программы

```
Сгенерирован следующий массив:
```

```
24.21   5.25    28.32   11.96   -10.43  -27.41
Количество положительных элементов менее 5?     Да!
Количество элементов меньших чем 50, не кратно 4!
Количество элементов меньших чем 55, не кратно 4!
```

Тема 3. Двумерные массивы

Задача №15

1. Условие задачи

Дан двумерный массив. Определить:
а) сумму квадратов элементов второй строки массива;
б) сумму квадратов элементов с-го столбца массива.

2. Словесное описание алгоритма

Генерируются значения переменных количества столбцов и количества строк будущего массива. Также генерируется значение переменной для задачи Б – номер столбца.

Массив моделируется контейнером вектор.

Данные для массива генерируются датчиком случайных чисел в функции CreateArray(), после чего массив выводится на экран.

В функции TaskA() вычисляется и выводится сумма квадратов элементов второй строки массива.

В функции TaskB() вычисляется и выводится Сумма квадратов элементов с-го столбца массива.

3. Описание интерфейса пользователя программы

После запуска программы на экран отображается следующая информация:

Сгенерированный массив.
Сумма квадратов элементов второй строки.
Сумма квадратов элементов с-го столбца.

4. Код программы

```
#include <stdio.h>
#include <vector>
#include <stdlib.h>
#include <ctime>
#include <math.h>

using namespace std;

//Прототип функции генерации массива
void CreateArray(vector<vector<int> >&vec);

//Прототип функции поиска суммы квадратов элементов второй строки
void TaskA(int H, vector<vector<int> >&vec);

//Прототип функции поиска суммы квадратов элементов с-го столбца
void TaskB(int W, int C, vector<vector<int> >&vec);

int main() {
```

```
    srand((unsigned int) time(NULL));
    int W = 2 + rand() % 10, H = 2 + rand() % 10; // W - Количество строк
массива, H - количество столбцов массива
    int C = 1 + rand() % H; // Число для вставки
    vector< vector<int>> vec(W, vector<int>(H));
    printf("Сгенерирован следующий массив:\n");
    CreateArray(vec); //Заполнение вектора
    TaskA(H, vec); //Задача А
    TaskB(W, C, vec); //Задача Б
}

void CreateArray(vector<vector<int> >&vec) {

    for (int i = 0; i < vec.size(); i++) {
        for (int j = 0; j < vec[i].size(); j++) {
            vec[i][j] = rand() % 10;
            printf("%d\t", vec[i][j]);
        }
        printf("\n");
    }

}

void TaskA(int H, vector<vector<int> >&vec) {
    int times = 0;
    for (int i = 0; i < H; i++) {
        times += pow(vec[1][i], 2);
    }
    printf("Сумма квадратов элементов второй строки = %d\n", times);
}

void TaskB(int W, int C, vector<vector<int> >&vec) {
    int times = 0;
    for (int i = 0; i < W; i++) {
        times += pow(vec[i][C - 1], 2);
    }
    printf("Сумма квадратов элементов %d-го столбца = %d", C, times);
}
```

5. Результат работы программы

```
Сгенерирован следующий массив:

9       6       4       5

0       2       9       3

2       3       4       2

Сумма квадратов элементов второй строки = 94

Сумма квадратов элементов 1-го столбца = 85
```

Задача №32

1. Условие задачи

Дан двумерный массив. Определить:
а) сумму элементов второго столбца массива, больших 10;
б) сумму элементов третьей строки массива, не превышающих 25;
в) количество ненулевых элементов первой строки массива;
г) количество элементов второго столбца массива, больших 15.

2. Словесное описание алгоритма

Генерируются значения переменных количества столбцов и количества строк будущего массива.

Массив моделируется контейнером вектор.

Данные для массива генерируются датчиком случайных чисел в функции CreateArray(), после чего массив выводится на экран.

В функции TaskA() вычисляется и выводится сумма элементов второго столбца массива, больших 10.

В функции TaskB() вычисляется и выводится сумма элементов третьей строки массива, не превышающих 25.

В функции TaskC() вычисляется и выводится количество ненулевых элементов первой строки массива.

В функции TaskD() вычисляется и выводится количество элементов второго столбца массива, больших 15.

3. Описание интерфейса пользователя программы

После запуска программы на экран отображается следующая информация:

Сгенерированный массив и результаты решения условий задачи.

4. Код программы

```
#include <stdio.h>
#include <vector>
#include <stdlib.h>
#include <ctime>
#include <math.h>

using namespace std;

//Прототип функции генерации массива
void CreateArray(vector<vector<int> >&vec);

//Прототип функции поиска суммы элементов второго столбца массива, больших 10
void TaskA(int W, vector<vector<int> >&vec);

//Прототип функции поиска суммы элементов третьей строки массива, не превышающих
25
void TaskB(int H, vector<vector<int> >&vec);

//Прототип функции поиска количества ненулевых элементов первой строки массива
void TaskC(int H, vector<vector<int> >&vec);

//Прототип функции поиска количества элементов второго столбца массива > 15
void TaskD(int W, vector<vector<int> >&vec);
int main() {
    srand((unsigned int) time(NULL));
    int W = rand() % 10 + 2, H = rand() % 10 + 2; // W - Количество строк
массива, H - количество столбцов массива
```

```
    vector< vector<int>> vec(W, vector<int>(H));
    printf("Сгенерирован следующий массив:\n");
    CreateArray(vec); //Заполнение вектора
    TaskA(W, vec); //Задача А
    TaskB(H, vec); //Задача Б
    TaskC(H, vec); //Задача В
    TaskD(W, vec); //Задача Г
}

void CreateArray(vector<vector<int> >&vec) {

    for (int i = 0; i < vec.size(); i++) {
        for (int j = 0; j < vec[i].size(); j++) {
            vec[i][j] = rand() % 30;
            printf("%d\t", vec[i][j]);
        }
        printf("\n");
    }

}

void TaskA(int W, vector<vector<int> >&vec) {
    int times = 0;
    for (int i = 0; i < W; i++) {
        if (vec[i][1] > 10)
            times += vec[i][1];
    }
    printf("Сумма элементов второго столбца массива, больших 10 = %d\n", times);
}

void TaskB(int H, vector<vector<int> >&vec) {
    int times = 0;
    for (int i = 0; i < H; i++) {
        if (vec[2][i] < 25)
            times += vec[2][i];
    }
    printf("Сумма элементов третьей строки массива, не превышающих 25 = %d\n",
times);
}

void TaskC(int H, vector<vector<int> >&vec) {
    int times = 0;
    for (int i = 0; i < H; i++) {
        if (vec[0][i] != 0)
            times++;
    }
    printf("Количество ненулевых элементов первой строки массива = %d\n", times);
}

void TaskD(int W, vector<vector<int> >&vec) {
    int times = 0;
    for (int i = 0; i < W; i++) {
        if (vec[i][1] > 15)
            times++;
    }
    printf("Количество элементов второго столбца массива, больших 15 = %d",
times);
}
```

5. Результат работы программы

```
25      16      6       27

26      8       27      14
```

```
9       6       21      13
20      11      27      2
20      1       13      6
1       10      9       6
22      23      21      15
29      22      29      16
0       5       14      27
13      11      11      15
Сумма элементов второго столбца массива, больших 10 = 83
Сумма элементов третьей строки массива, не превышающих 25 = 49
Количество ненулевых элементов первой строки массива = 4
Количество элементов второго столбца массива, больших 15 = 3
```

Использованные источники (список литературы и веб-ресурсов)

1. Язык программирования С. Керниган Брайан У., Ритчи Деннис М.