**­­Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет «Высшая школа экономики»**

Московский институт электроники и математики им. А.Н.Тихонова

НИУ ВШЭ

Департамент компьютерной инженерии (или департамент электронной инженерии)

**Курс: Алгоритмизация и программирование**

**ОТЧЁТ**

**по лабораторной работе №3**

|  |  |  |
| --- | --- | --- |
| **Раздел** | **Маx**  **оценка** | **Итог.**  **оценка** |
| Постановка | 0,5 |  |
| Метод | 1 |  |
| Спецификация | 0,5 |  |
| Алгоритм | 1,5 |  |
| Работа программы | 1 |  |
| Листинг | 0,5 |  |
| Тесты | 1 |  |
| Вопросы | 2 |  |
| Доп. задание | 2 |  |

**Студент: Федоров Матвей Евгеньевич**

**Группа: БИВ243**

**Вариант: №208(9, 2, 6)**

**Руководитель: Литвиненко Алексей Михайлович**

**Оценка:**

**Дата сдачи:**

**МОСКВА 2024**

**Содержание**

[Содержание 2](#_Toc179205071)

[Задание 3](#_Toc179205072)

[Постановка задачи 4](#_Toc179205073)

[Метод решения задачи 5](#_Toc179205074)

[Внешняя спецификация 6](#_Toc179205075)

[Описание алгоритма на псевдокоде 6](#_Toc179205076)

[Листинг программы 12](#_Toc179205077)

[Распечатка тестов к программе и результатов 16](#_Toc179205078)

# **Задание**

1. Оформить в виде функции общую чать в предложенных формулах:
2. Вычислить значение по формуле:



1. Дан целочисленный массив A[1:n]. Написать программу, включающую две функции с параметрами. В первой функции необходимо подсчитать количество повторений каждого элемента массива A. Вторая функция решает следующую задачу:

Сформировать новый массив, содержащий неповторяющиеся нечетные элементы массива A.

**Постановка задачи**

Дано:

1. a, b, c, d, e, f – вещ.
2. n, m – вещ., a[1:n][1:m] – вещ.
3. n – нат., A[1:n] – цел.

Результат:

1. x, y – вещ.
2. O – вещ.
3. B[1:k] или «В массиве нет нечетных чисел, либо все нечетные числа повторяются.»

При:

1. a, b, c, d, e, f ∈
2. n, m  , n, m ≤ lmax
3. n  , n ≤ lmax, a[i] ,

Связь:

1. См. формулу в условии
2. См. формулу в условии
3. i = j = : ,

a[i] = a[j]: !t = : B[t] = A[i]

# **Метод решения задачи**

# **Внешняя спецификация**

Задание №1

Введите значние переменной a:

Введите значние переменной b:

Введите значние переменной c:

Введите значние переменной d:

Введите значние переменной e:

Введите значние переменной f:

Результат работы программы: x = <<x>>, y = <<y>>

Задание №2

Введите значения для матрицы:

<mtrx[1][1]>, <mtrx[1][2]>, … , <mtrx[mtrx\_rws][mtrx\_cols]>

Введенная матрица:

<<mtrx[1][1]>>, <<mtrx[1][2]>>, … , <<mtrx[mtrx\_rws][mtrx\_cols]>>

Результат работы программы: <<O>>

Задание №3

Введите значения для массива:

<arr[1]>, <arr[2]>, … , <arr[arr\_len]>

Введенный массив:

<<arr[1]>>, <<arr[2]>>, … , <<arr[arr\_len]>>

При new\_arr\_len = 0

Иначе

**Описание алгоритма на псевдокоде**

flag := 0

lmax := 200

Алг «op\_f»

Вход: a, b, c, d, e, f

Выход res

нач

flag := 0

numerator := b \* (f - d\*c/a)

denominator := a \* (e - d\*b/a)

если denominator != 0

res = numerator / denominator

иначе

flag := 1

res := 0

все

кон

Алг «input\_n»

Вход name

Выход n

нач

вывод("Введите значние переменной <name>")

ввод(n)

кон

Алг «Задание 1»

нач

цикл пока flag = 1 или a = 0

a := вызов(input\_n("a"))

b := вызов(input\_n ("b"))

c := вызов(input\_n ("c"))

d := вызов(input\_n ("d"))

e := вызов(input\_n ("e"))

f := вызов(input\_n ("f"))

op := вызов(op\_f(a, b, c, d, e, f))

если flag = 1

вывод("Введены некорректные значения переменных.")

все

кц

x := c/a - (b/a) \* op;

y := op;

вывод("Результат работы программы: x = <y>, y = <y>")

кон

Алг «print\_mtrx»

Вход: mtrx[lmax][lmax], mtrx\_l, mtrx\_h

нач

вывод("Введенная матрица:")

цикл от i = 0 до mtrx\_h; i [шаг 1]

цикл от j = 0 до mtrx\_l [шаг 1]

вывод(mtrx[i][j])

кц

кц

кон

Алг «input\_mtrx»

Вход: mtrx[][lmax], mtrx\_l, mtrx\_h

нач

цикл пока mtrx\_l < 0 или mtrx\_h < 0 или mtrx\_l > lmax или mtrx\_h > lmax

вывод("Введите длинну и высоту матрицы:");

вывод(«Вы все равно это не прочитаете бебебе»)

ввод(mtrx\_l)

ввод(mtrx\_h)

кц

вывод("Введите значения для матрицы:\n");

цикл от i = 0 до mtrx\_h [шаг 1]

цикл от j = 0 до mtrx\_l [шаг 1]

ввод(mtrx[i][j])

кц

кц

вызов(print\_mtrx(mtrx, \*mtrx\_l, \*mtrx\_h))

кон

Алг «find\_min\_max»

Вход: mtrx[][lmax], mtrx\_l, mtrx\_h

Выход: min

нач

min := 0, max := 0, flag := 0

цикл от i := 0 до mtrx\_h [шаг 1]

max := 0

цикл от j := 0 до mtrx\_l [шаг 1]

если |mtrx[i][j]| > |max|

max := mtrx[i][j]

все

кц

если flag = 0

min := max

flag := 1

иначе

если |max| < |min|

min := max;

все

все

кц

кон

Алг «Задание 2»

нач

вызов(input\_mtrx(mtrx, mtrx\_l, mtrx\_h))

O = вызов(find\_min\_max(mtrx, mtrx\_l, mtrx\_h))

вывод("Результат работы программы: <O>")

кон

Алг «print\_arr»

Вход: arr[lmax], arr\_len

нач

цикл от i := 0 до arr\_len [шаг 1]

вывод(arr[i])

кц

кон

Алг input\_arr

Ввод: arr[lmax], arr\_len

нач

вывод("Введите элементы массива:")

цикл от i := 0 до arr\_len [шаг 1]

ввод(arr[i])

кц

вывод("Введенный массив:")

вызов(print\_arr(arr, arr\_len))

кон

Алг «count»

Вход: a[lmax], el, len

Выход: res

res := 0

цикл от i = 0 до len [шаг 1]

если a[i] = el

res := res + 1

все

кц

кон

Алг «Задание 3»

нач

new\_arr\_len := 0, cnt := 0

цикл пока arr\_len < 0 или arr\_len > lmax

вывод("Введите длинну массива:")

ввод(arr\_len)

кц

вызов(input\_arr(arr, arr\_len))

цикл от i := 0 до arr\_len [шаг 1]

cnt := вызов(count(arr, arr[i], arr\_len)

вывод("Элемент <arr[i]> встречается в массиве <cnt> раз.")

если cnt = 1 и arr[i] % 2 != 0

new\_arr[new\_arr\_len] := arr[i]

new\_arr\_len := new\_arr\_len + 1

все

кц

если new\_arr\_len = 0

вывод ("В массиве нет нечетных чисел, либо все нечетные числа повторяются.")

иначе

вывод("Полученный массив:")

вызов(print\_arr(new\_arr, new\_arr\_len))

все

кон

**Листинг программы**

Задание №1

#include "stdio.h"

**int** flag = 0;

**float** op\_f (**float** a, **float** b, **float** c, **float** d, **float** e, **float** f) {

flag = 0;

**float** numerator, denominator;

numerator = b \* (f - d\*c/a);

denominator = a \* (e - d\*b/a);

**if** (denominator != 0) {

**return** numerator / denominator;

} **else** {

flag = 1;

**return** 0;

}

}

**float** input (**char** \*name) {

**float** n;

printf("Введите значние переменной %s:\n", name);

scanf("%f", &n);

**return** n;

}

**int** main (**void**) {

**float** a, b, c, d, e, f, x, y, op;

**do** {

a = input("a");

b = input("b");

c = input("c");

d = input("d");

e = input("e");

f = input("f");

op = op\_f(a, b, c, d, e, f);

**if** (flag == 1) {

printf("Введены некорректные значения переменных.\n");

}

} **while** (flag == 1 || a == 0);

x = c/a - (b/a) \* op;

y = op;

printf("Результат работы программы: x = %f, y = %f\n", x, y);

}

Задание 2

#include "stdio.h"

#include "stdlib.h"

#define lmax 200

**int** input (**void**) {

**int** n;

scanf("%d", &n);

**return** n;

}

**void** print\_mtrx(**int** mtrx[lmax][lmax], **int** mtrx\_l, **int** mtrx\_h) {

printf("Введенная матрица:\n");

**for** (**int** i = 0; i < mtrx\_h; i++) {

**for** (**int** j = 0; j < mtrx\_l; j++) {

printf("%d ",mtrx[i][j]);

}

printf("\n");

}

}

**void** input\_mtrx(**int** mtrx[][lmax], **int** \*mtrx\_l, **int** \*mtrx\_h) {

**do** {

printf("Введите длинну и высоту матрицы:\n");

\*mtrx\_l = input();

\*mtrx\_h = input();

} **while** (\*mtrx\_l < 0 || \*mtrx\_h < 0 || \*mtrx\_l > lmax || \*mtrx\_h > lmax);

printf("Введите значения для матрицы:\n");

**for** (**int** i = 0; i < \*mtrx\_h; i++) {

**for** (**int** j = 0; j < \*mtrx\_l; j++) {

mtrx[i][j] = input();

}

}

print\_mtrx(mtrx, \*mtrx\_l, \*mtrx\_h);

}

**int** find\_min\_max (**int** mtrx[][lmax], **int** mtrx\_l, **int** mtrx\_h) {

**int** min = 0, max = 0, flag = 0;

**for** (**int** i = 0; i < mtrx\_h; i++) {

max = 0;

**for** (**int** j = 0; j < mtrx\_l; j++) {

**if** (abs(mtrx[i][j]) > abs(max)) {

max = mtrx[i][j];

}

}

**if** (flag == 0) {

min = max;

flag = 1;

} **else** {

**if** (abs(max) < abs(min)) {

min = max;

}

}

}

**return** min;

}

**int** main (**void**) {

**int** mtrx\_l, mtrx\_h, mtrx[lmax][lmax], O;

input\_mtrx(mtrx, &mtrx\_l, &mtrx\_h);

O = find\_min\_max(mtrx, mtrx\_l, mtrx\_h);

printf("Результат работы программы: %d\n", O);

}

Задание 3  
#include "stdio.h"

#define lmax 200

**int** input (**void**) {

**int** n;

scanf("%d", &n);

**return** n;

}

**void** print\_arr (**int** arr[lmax], **int** \*arr\_len) {

**for** (**int** i = 0; i < \*arr\_len; i++) {

printf("%d ", arr[i]);

}

printf("\n");

}

**void** input\_arr (**int** arr[lmax], **int** \*arr\_len) {

printf("Введите элементы массива:\n");

**for** (**int** i = 0; i < \*arr\_len; i++) {

arr[i] = input();

}

printf("Введенный массив:\n");

print\_arr(arr, &\*arr\_len);

}

**int** count(**int** a[lmax], **int** el, **int** len){

**int** res = 0;

**for** (**int** i = 0; i < len; i++){

**if** (a[i] == el)

res++;

}

**return** res;

}

**int** main (**void**) {

**int** arr[lmax], arr\_len, new\_arr[lmax], new\_arr\_len = 0, cnt = 0;

**do** {

printf("Введите длинну массива:\n");

arr\_len = input();

} **while** (arr\_len < 0 || arr\_len > lmax);

input\_arr(arr, &arr\_len);

**for** (**int** i = 0; i < arr\_len; i++) {

cnt = count(arr, arr[i], arr\_len);

printf("Элемент %d встречается в массиве %d раз;\n", arr[i], cnt);

**if** (cnt == 1 && arr[i] % 2 != 0) {

new\_arr[new\_arr\_len] = arr[i];

new\_arr\_len++;

}

}

**if** (new\_arr\_len == 0) {

printf("В массиве нет нечетных чисел, либо все нечетные числа повторяются.\n");

} **else** {

printf("Полученный массив:\n");

print\_arr(new\_arr, &new\_arr\_len);

}

}

# **Распечатка тестов к программе и результатов**

Задание №1

|  |  |  |
| --- | --- | --- |
| № | Исходные данные | Результаты |
| 1 | 4  5  6  7  8  9 | Результат работы программы: x = -1.625000, y = 2.500000 |
| 2 | 1  2  3  4  5  6 | Результат работы программы: x = -5.000000, y = 4.000000 |

Задание №2

|  |  |  |
| --- | --- | --- |
| № | Исходные данные | Результаты |
| 1 | 3  2  1  2  3  4  5  6 | Результат работы программы: 3 |
| 2 | 2  3  9  8  7  -1  -2  -3 | Результат работы программы: -3 |

Задание №3

|  |  |  |
| --- | --- | --- |
| № | Исходные данные | Результаты |
| 1 | 5  1  2  3  4  5 | Элемент 1 встречается в массиве 1 раз;  Элемент 2 встречается в массиве 1 раз;  Элемент 3 встречается в массиве 1 раз;  Элемент 4 встречается в массиве 1 раз;  Элемент 5 встречается в массиве 1 раз;  Полученный массив:  1 3 5 |
| 2 | 5  1  1  2  4  2 | Элемент 1 встречается в массиве 2 раз;  Элемент 1 встречается в массиве 2 раз;  Элемент 2 встречается в массиве 2 раз;  Элемент 4 встречается в массиве 1 раз;  Элемент 2 встречается в массиве 2 раз;  В массиве нет нечетных чисел, либо все нечетные числа повторяются. |