

**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

Курс: Алгоритмизация и программирование

Отчёт

по лабораторной работе №3

Раздел	Мак 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: Андреев Глеб Андреевич

Группа: БИВ254

Вариант: №252 (3, 1, 1)

**Руководитель: Альбатша Ахмад
Мухаммад Хусайн**

Оценка: _____

Дата сдачи: _____

Оглавление

<i>Задание</i>	<i>3</i>
<i>Метод решения задачи</i>	<i>5</i>
<i>Внешняя спецификация.....</i>	<i>7</i>
<i>Задача 1.....</i>	<i>7</i>
<u> Задача 2.....</u>	<i>8</i>
<u> Задача 3.....</u>	<i>9</i>
<i>Описание алгоритма на псевдокоде</i>	<i>10</i>
<u> Задача 1.....</u>	<i>10</i>
<u> Задача 2.....</u>	<i>12</i>
<u> Задача 3.....</u>	<i>14</i>
<i>Листинг программы.....</i>	<i>17</i>
<u> Задача 1.....</u>	<i>17</i>
<u> Задача 2.....</u>	<i>19</i>
<u> Задача 3.....</u>	<i>21</i>
<i>Распечатка тестов к программе и результатов</i>	<i>23</i>
<u> Задача 1.....</u>	<i>23</i>
<u> Задача 2.....</u>	<i>24</i>
<u> Задача 3.....</u>	<i>25</i>

Задание

ЦЕЛЬ РАБОТЫ: Использование функций.

1. (3) Оформить в виде функции общую часть в предложенных формулах:

Функция $y = f(x)$ задана таблицей:

X_i	X_1	X_2	X_3
Y_i	Y_1	Y_2	Y_3

Вычислить приближенное значение $y = f(x)$ в точке $x_1 \leq x \leq x_3$ по формуле:

$$f(x) = \begin{cases} y_1 + \frac{x-x_1}{x_2-x_1}(y_2-y_1), & \text{если } x_1 \leq x < x_2, \\ y_2 + \frac{x-x_2}{x_3-x_2}(y_3-y_2), & \text{если } x_2 \leq x < x_3, \\ y_3, & \text{если } x = x_3. \end{cases}$$

ЗАМЕЧАНИЕ. При решении задачи должна получиться одна функция, вызываемая несколько раз.

2. (1) Дана целочисленная матрица. Вычислить значение по формуле:

$$1) V = \min_{i=1,n} \max_{j=1,m-1} |c_{i,j} - c_{i,j+1}|, \text{ где } c_{i,j} - \text{элементы матрицы } C[1:n, 1:m];$$

3. (1) Дан целочисленный массив $A[0:n-1]$. К элементам массива обращаться при помощи указателя. Написать программу, включающую две функции с параметрами. В первой функции необходимо подсчитать количество повторений каждого элемента массива A . Вторая функция решает следующую задачу: удалить из массива A все повторяющиеся элементы путем сдвига (дополнительный массив не использовать).

ЗАМЕЧАНИЕ. Каждую часть оформить как отдельную программу.

ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ РАБОТЫ.

1. Необходима проверка допустимости исходных данных, в том числе недопустим ввод строки вместо числа и вещественного числа вместо целого.
2. При решении задачи необходимо оформить как отдельные функции ввод данных, вывод результатов, обработку массива.

Постановка задачи

Дано:

1. $x, x_1, x_2, x_3, y_1, y_2, y_3$ – вещественные числа
2. $A[1:n][1:m]$ – целочисленная матрица
3. $A[0:n-1]$ – целочисленный массив

Результат:

1. $result$ – вещественное число
2. V – целое число или сообщение “Некорректная матрица.”
3. Количество повторений элементов и (A – целочисленный массив или сообщение “Новый массив пуст.”)

При: $m, n \in N, m, n \leq lmax$

Связь:

1. См. формулу в условии
2. См. формулу в условии
3. $\forall i = \overline{0, n-1} \exists count = \{0, \text{если } \nexists j = \overline{i, n-1} : A[i] = A[j]; \text{ иначе } count + 1\}$

$$\forall i = \overline{0, n-1} : \exists j = \overline{i, j} : A[i] = A[j], \exists g = \overline{i, n-1} : R[i] = R[g]$$

Метод решения задачи

$$1. \quad \left\{ \begin{array}{l} \text{если } x_1 \leq x < x_2 \\ \text{то } x = y_1 + \frac{x-x_1}{x_2-x_1} (y_2 - y_1) \\ \text{иначе если } x_2 \leq x < x_3 \\ \text{то } x = y_2 + \frac{x-x_2}{x_3-x_2} (y_3 - y_2) \\ \text{иначе} \\ x = y_3 \end{array} \right.$$

$$2. \quad \left\{ \begin{array}{l} \min = INT_MAX \\ \text{для } i = \overline{1, n} \\ \max = A[i][1] - A[i][2] \\ \left\{ \begin{array}{l} \text{для } j = \overline{1, m-1} \\ \left\{ \begin{array}{l} \text{если } |A[i][j] - A[i][j-1]| > |\max| \\ \text{то } \max = A[i][j] - A[i][j+1] \end{array} \right. \\ \text{если } \max < \min \\ \min = \max \end{array} \right. \end{array} \right.$$

3.

$$a. \left\{ \begin{array}{l} \text{для } i = \overline{0, n-1} \\ \quad count = 0 \\ \quad processed = 0 \\ \quad \left\{ \begin{array}{l} \text{для } j = \overline{0, i-1} \\ \left\{ \begin{array}{l} \text{если } A[i] = A[j] \\ \text{то } processed = 1 \end{array} \right. \\ \left\{ \begin{array}{l} \text{если } processed \\ \quad continue \end{array} \right. \end{array} \right. \\ \quad \left\{ \begin{array}{l} \text{для } j = \overline{i+1, n-1} \\ \left\{ \begin{array}{l} \text{если } A[i] = A[j] \\ count = count + 1 \end{array} \right. \end{array} \right. \end{array} \right.$$

b. $i = 0$

$$\left\{ \begin{array}{l} \text{пока } i < n \\ isFound = 0 \\ \left\{ \begin{array}{l} \text{для } j = \overline{i+1, n-1} \\ \left\{ \begin{array}{l} \text{если } A[i] = A[j] \\ \text{то} \\ \left\{ \begin{array}{l} \text{если } j \neq n-1 \\ \text{то} \\ \left\{ \begin{array}{l} \text{для } g = \overline{j, n-1} \\ A[g] = A[g+1] \end{array} \right. \\ n = n-1 \end{array} \right. \\ isFound = 1 \end{array} \right. \\ \left\{ \begin{array}{l} \text{если } isFound = 1 \\ \left\{ \begin{array}{l} \text{для } j = \overline{i, n-1} \\ A[j] = A[j+1] \end{array} \right. \\ n = n-1 \\ i = 0 \end{array} \right. \\ \text{иначе } i = i+1 \end{array} \right. \end{array} \right.$$

Внешняя спецификация

Задача 1

Лабораторная работа №3, задача №1

Введите вещественное число x_1

{ }*

Введите вещественное число x_2

{ }*

Введите вещественное число x_3

{ }*

Введите вещественное число y_1

{ }*

Введите вещественное число y_2

{ }*

Введите вещественное число y_3

{ }*

Введите вещественное число x

{ }*

Примерное значение $f(x)$:

<< result >>

Задача 2

Лабораторная работа №3, задача №2

Введите количество строк матрицы A (от 1 до $\ll lmax \gg$)

{ $\ll n \gg$ }^{*} до $n > 0$ и $n \leq lmax$

Введите количество столбцов матрицы A (от 1 до $\ll lmax \gg$)

{ $\ll m \gg$ }^{*} до $m > 0$ и $m \leq lmax$

Введите элементы матрицы

$\langle A[1][1] \rangle \langle A[1][2] \rangle \dots \langle A[n][m] \rangle$

Исходная матрица A :

$\ll A[1][1] \gg \ll A[1][2] \gg \dots \ll A[n][m] \gg$

Запрошенное значение V :

$\ll V \gg$

Задача 3

Лабораторная работа №3, задача №3

Введите количество элементов массива A (от 1 до $\ll lmax \gg$)

{ $\ll n \gg$ } * до $n > 0$ и $n \leq lmax$

Введите элементы массива A :

$\langle A[0] \rangle \langle A[1] \rangle \dots \langle A[n - 1] \rangle$

Исходный массив A :

$\langle\langle A[0] \rangle\rangle \langle\langle A[1] \rangle\rangle \dots \langle\langle A[n - 1] \rangle\rangle$

{ для $i = \overline{0, n - 1}$
Количество повторений элемента $A[i]$:
 $\ll count_i \gg$

{ если $n = 0$
Полученный массив пуст
иначе
Новый массив A :
 $\langle\langle A[0] \rangle\rangle \langle\langle A[1] \rangle\rangle \dots \langle\langle A[n - 1] \rangle\rangle$

Описание алгоритма на псевдокоде

Задача 1

алг <<функция input_vars>>

ВХОД $x_1, x_2, x_3, y_1, y_2, y_3$ – вещ.

ВЫХОД $x_1, x_2, x_3, y_1, y_2, y_3$ – вещ.

нач

ввод(x_1)

ввод(x_2)

ввод(x_3)

ввод(y_1)

ввод(y_2)

ввод(y_3)

кон

алг <<функция calculate_function>>

ВХОД $x_1, x_2, x_3, y_1, y_2, y_3$ – вещ.

ВЫХОД –

нач

ЦИКЛ-ПОКА $x < x_1$ ИЛИ $x > x_3$

ввод(x – вещ.)

кц

если $x < x_2$

то $result := y_1 + (x - x_1) * (y_2 - y_1) / (x_2 - x_1)$

иначе если $x < x_3$

то $result := y_2 + (x - x_2) * (y_3 - y_2) / (x_3 - x_2)$

иначе

$result := y_3$

все

все

вывод(“Примерное значение $f(“x, “) = “$, result)

кон

алг «Главная Функция»

нач

вывод(“Лабораторная работа №3, задача №1”)

{ вызов функции } input_vars($x_1, x_2, x_3, y_1, y_2, y_3$)

{ вызов функции } calculate_function($x_1, x_2, x_3, y_1, y_2, y_3$)

кон

Задача 2

алг <<функция insert_matrix>>

вход A – целочисленная матрица, n, m – целые числа

выход –

нач

вывод(“Введите элементы матрицы A:”)

ввод($A[1:n][1:m]$)

кон

алг <<функция print_matrix>>

вход A – целочисленная матрица, n, m – целые числа

выход –

нач

вывод($A[1:n][1:m]$)

кон

алг <<функция calculate>>

вход A – целочисленная матрица, n, m – целые числа

выход min – целое число

нач

min := INT_MAX

цикл от $i := 1$ до n

max = $A[i][1] - A[i][2]$

цикл от $j := 1$ до $m - 1$

если $|A[i][j] - A[i][j + 1]| > \text{max}$

то max = $A[i][j] - A[i][j + 1]$

все

кц

если max < min

то min = max

все

кц

кон

алг «Главная Функция»

нач

вывод("Лабораторная работа №3, задача №2")

$l_{\max} := 100$

цикл пока $n < 0$ или $n > l_{\max}$

вывод("Введите количество строк матрицы Q (от 1 до ", l_{\max} , "): ")

ввод(n)

кц

цикл пока $m < 0$ или $m \geq l_{\max}$

вывод("Введите количество столбцов матрицы Q (от 1 до ", l_{\max} , "): ")

ввод(m)

кц

если $n = 1$ и $m = 1$

вывод("Некорректная матрица")

все

{ вызов функции } insert_matrix(A, n, m)

вывод("Исходная матрица A: ")

{ вызов функции } print_matrix (A, n, m)

$V :=$ { вызов функции } calculate (A, n, m)

Вывод("Значение $V =$ ", V)

кон

Задача 3

алг <<функция insert_array>>

вход A – целочисленная массив, n – целые числа

выход –

нач

вывод(“Введите элементы массива A:”)

ввод($A[1:n][1:m]$)

кон

алг <<функция print_array>>

вход A – целочисленная массив, n – целые числа

выход –

нач

вывод($A[1:n][1:m]$)

кон

алг <<функция calculate >>

вход A – целочисленная массив, n – целые числа

выход n – целое число

нач

$i := 0$

цикл-пока $i < n$

$is_found := 0$

цикл от $j := i + 1$ до $n - 1$

если $A[i] = A[j]$ то

если $j \neq n - 1$ то

цикл от $g := j$ до $n - 2$

$A[g] = A[g+1]$

кц

$n := n - 1$

$is_found := 1$

```

        всё

        кц
    если is_found = 0 то
        цикл от j := i до n - 2
            A[j] = A[j + 1]

        кц
        n := n - 1
        i := 0

    иначе
        i = i + 1

    всё

    кц

кон

```

алг <<функция count_duplicates>>

вход A — целочисленная массив, n — целые числа

выход —

нач

вывод(“Подсчёт элементов”)

цикл от i := 0 до n - 1

count := 0

processed := 0

цикл от j := 0 до i - 1

если A[i] = A[j] то

processed := 1

всё

кц

если processed = 1 то

продолжить

всё

цикл от j := i + 1 до n - 1

```

        если  $A[i] = A[j]$  то
            count := count + 1
        все
    кц
    вывод("Элемент", A[i], "повторяется", count, "раз")
кц
кон

алг «Главная Функция»
    нач
        вывод("Лабораторная работа №3, задача №3")
        lmax := 100
        цикл пока  $n < 0$  или  $n > lmax$ 
            вывод("Введите количество элементов массива A (от 1 до ", lmax, "): ")
            ввод(n)
        кц

        { вызов функции } insert_array(A, n,)
        вывод("Исходный массив A: ")
        { вызов функции } print_array (A, n)
        { вызов функции } count_duplicates (A, n)
        n := { вызов функции } calculate (A, n, m)
        если  $n = 0$  то
            вывод("Новый массив пуст.")
        иначе
            вывод("Новый массив:", A[0:n-1])
        все
    кон

```


Листинг программы

Задача 1

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

float calculate_function(float x1, float x2, float x3, float y1, float y2,
float y3) {
    float x;
    int ret;
    do {
        printf("Insert value of point x (It must belong to the segment [x1;
x3]): \n");
        ret = scanf("%f", &x);
        while (getchar() != '\n');
    } while (ret != 1 || x < x1 || x > x3);

    float result;
    if (x < x2) {
        result = y1 + (x - x1) * (y2 - y1) / (x2 - x1);
    } else if (x < x3) {
        result = y2 + (x - x2) * (y3 - y2) / (x3 - x2);
    } else {
        result = y3;
    }

    printf("Approximate value f(%.2f) = %.2f \n", x, result);
    return result;
}

void input_vars(float *x1, float *x2, float *x3, float *y1, float *y2, float
*y3) {
    int ret;
    do {
        printf("Enter the real variable x1: \n");
        ret = scanf("%f", x1);
        while (getchar() != '\n');
    } while (ret != 1);

    do {
        printf("Enter the real variable x2: \n");
        ret = scanf("%f", x2);
        while (getchar() != '\n');
    } while (ret != 1);

    do {
        printf("Enter the real variable x3: \n");
        ret = scanf("%f", x3);
        while (getchar() != '\n');
    } while (ret != 1);

    do {
        printf("Enter the real variable y1: \n");
        ret = scanf("%f", y1);
        while (getchar() != '\n');
    } while (ret != 1);
}
```

```

do {
    printf("Enter the real variable y2: \n");
    ret = scanf("%f", y2);
    while (getchar() != '\n');
} while (ret != 1);

do {
    printf("Enter the real variable y3: \n");
    ret = scanf("%f", y3);
    while (getchar() != '\n');
} while (ret != 1);
}

int main() {
    float x1, x2, x3, y1, y2, y3;
    printf("Лабораторная работа №3, задача №2\n");
    input_vars(&x1, &x2, &x3, &y1, &y2, &y3);
    calculate_function(x1, x2, x3, y1, y2, y3);
    return 0;
}

```

Задача 2

```
#include <stdio.h>
#include <limits.h>
#include <math.h>
#define lmax 100

void insert_matrix(int A[][lmax], int n, int m) {
    char check;
    int temp;
    printf("Enter the elements of matrix A:\n");
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            do {
                scanf("%d", &temp);
                check = getchar();
                if (check != '\n') {
                    printf("Invalid input.\n");
                    temp = INT_MAX;
                    while (check != '\n' && check != EOF) {
                        check = getchar();
                    }
                }
            } while (check != '\n' || temp == INT_MAX);
            A[i][j] = temp;
        }
    }
}

void print_matrix(int A[][lmax], int n, int m) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            printf("%d ", A[i][j]);
        }
        printf("\n");
    }
}

int calculate(int A[][lmax], int n, int m) {
    int min, max;
    min = INT_MAX;
    for (int i = 1; i <= n; i++) {
        max = A[i][1] - A[i][2];
        for (int j = 1; j < m - 1; j++)
            if (abs(A[i][j] - A[i][j + 1]) > abs(max))
                max = A[i][j] - A[i][j + 1];

        if (max < min)
            min = max;
    }
    return min;
}

int main() {
    int A[lmax][lmax], n, m, V;
    char check;
    printf("Lab work #3, task #2\n");
    do {
```

```

        printf("Enter the number of rows in matrix A (from 1 to %d): ",
lmax);
        scanf("%d", &n);
        check = getchar();
        if (check != '\n' || n <= 0) {
            printf("Invalid input (n must be a natural number).\n");
            n = 0;
            while (check != '\n' && check != EOF) {
                check = getchar();
            }
        }
    } while (n <= 0 || n > lmax);

    do {
        printf("Enter the number of columns in matrix A (from 1 to %d): ",
lmax);
        scanf("%d", &m);
        check = getchar();
        if (check != '\n' || m <= 0) {
            printf("Invalid input (m must be a natural number).\n");
            m = 0;
            while (check != '\n' && check != EOF) {
                check = getchar();
            }
        }
    } while (m <= 0 || m > lmax);
    if (n == 1 && m == 1) {
        printf("Incorrect matrix A. \n");
        return 0;
    }
    insert_matrix(A, n, m);
    printf("Original matrix A: \n");
    print_matrix(A, n, m);
    V = calculate(A, n, m);
    printf("Your desired value V = %d: \n", V);
    return 0;
}

```

Задача 3

```
#include <stdio.h>
#include <limits.h>
#include <math.h>
#define lmax 100

void insert_array(int A[], int n) {
    char check;
    int temp;
    printf("Enter the elements of array A:\n");
    for (int i = 0; i < n; i++) {
        do {
            scanf("%d", &temp);
            check = getchar();
            if (check != '\n') {
                printf("Invalid input.\n");
                temp = INT_MAX;
                while (check != '\n' && check != EOF) {
                    check = getchar();
                }
            }
        } while (check != '\n' || temp == INT_MAX);
        A[i] = temp;
    }
}

void print_array(int A[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d ", A[i]);
    printf("\n");
}

int calculate(int A[], int n) {
    int is_found = 0, i = 0;
    while (i < n) {
        is_found = 0;
        for (int j = i + 1; j < n; j++) {
            if (A[i] == A[j]) {
                if (j != n - 1)
                    for (int g = j; g < n - 1; g++)
                        A[g] = A[g + 1];
                n--;
                is_found = 1;
            }
        }
        if (is_found) {
            for (int j = i; j < n - 1; j++)
                A[j] = A[j + 1];
            n--;
            i = 0;
        }
        else
            i++;
    }
    return n;
}
```

```

void count_duplicates(int A[], int n) {
    printf("Element counts:\n");
    int processed = 0;

    for (int i = 0; i < n; i++) {
        int count = 0;
        processed = 0;
        for (int j = 0; j < i; j++) {
            if (A[i] == A[j]) {
                processed = 1;
                continue;
            }
        }
        if(processed)
            continue;
        for (int j = i + 1; j < n; j++){
            if (A[i] == A[j])
                count++;
        }
        printf("Element %d repeats %d time(s)\n", A[i], count);
    }
}

int main() {
    int A[lmax][lmax], n, m, V;
    char check;
    printf("Lab work #3, task #3\n");
    do {
        printf("Enter the number of elements in array A (from 1 to %d): ",
lmax);
        scanf("%d", &n);
        check = getchar();
        if (check != '\n' || n <= 0) {
            printf("Invalid input (n must be a natural number).\n");
            n = 0;
            while (check != '\n' && check != EOF) {
                check = getchar();
            }
        }
    } while (n <= 0 || n > lmax);

    insert_array(*A, n);
    printf("Original array A: \n");
    print_array(*A, n);
    count_duplicates(*A, n);
    n = calculate(*A, n);
    if (n == 0) {
        printf("New array is empty. \n");
        return 0;
    }
    printf("New array: \n");
    print_array(*A, n);
    return 0;
}

```

Распечатка тестов к программе и результатов

Задача 1

№	Исходные данные	Результаты
1	$x_1 = 1.123$ $x_2 = 4.1$ $x_3 = 2$ $y_1 = 3.5$ $y_2 = 12$ $y_3 = 1$ $x = 1.4$	Approximate value $f(1.40) = 4.29$
2	$x_1 = -12$ $x_2 = 0.4$ $x_3 = 3$ $y_1 = -43$ $y_2 = -3$ $y_3 = 5$ $x = 3$	Approximate value $f(3.00) = 5.00$

Задача 2

№	Исходные данные	Результаты
1	n=1 m=1	Incorrect matrix A.
2	n=2 m=3 $A = \begin{bmatrix} 1 & 4 & 5 \\ -1 & 2 & 3 \end{bmatrix}$	Original matrix A: $\begin{bmatrix} 1 & 4 & 5 \\ -1 & 2 & 3 \end{bmatrix}$ Your desired value V = -3
3	n=2 m=4 $A = \begin{bmatrix} 4 & 7 & 1 & 3 \\ 5 & -1 & 0 & 5 \end{bmatrix}$	Original matrix A: $\begin{bmatrix} 4 & 7 & 1 & 3 \\ 5 & -1 & 0 & 5 \end{bmatrix}$ Your desired value V = 6

Задача 3

№	Исходные данные	Результаты
1	n=5 A = [1, 1, 1, 1, 1]	Original array A: [1, 1, 1, 1, 1] Element counts: Element 1 repeats 4 time(s) New array is empty.
2	n=6 A = [1, 1, 2, 3, 2, 5]	Original array A: [1, 1, 2, 3, 2, 5] Element counts: Element 1 repeats 1 time(s) Element 2 repeats 1 time(s) Element 3 repeats 0 time(s) Element 5 repeats 0 time(s) New array: [3, 5]
3	n=5 A = [1, 2, 3, 4, 5]	Original array A: [1, 2, 3, 4, 5] Element counts: Element 1 repeats 0 time(s) Element 2 repeats 0 time(s) Element 3 repeats 0 time(s) Element 4 repeats 0 time(s) Element 5 repeats 0 time(s) New array: [1, 2, 3, 4, 5]