**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**

**Отчет**

**по лабораторной работе №1**

Студент:Иванилов Леонид Михайлович

Группа: БИВ231

Вариант: 143(4,9,4)

|  |  |  |
| --- | --- | --- |
| Раздел | Max 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 |  |

Руководитель:

Оценка

Дата сдачи

**Содержание**

**Задание………………………………………………………………………...3**

**Постановка задачи…………………………………………………………..4**

**Метод решения……………………………………………………………….5**

**Внешняя спецификация…………………………………………………….6**

**Описание алгоритма на псевдокоде………………………………………7**

**Листинг программы……………………………………………………….10**

**Тестовые данные…………………………………………………………...14**

**Москва 2023**

**Задание**

Лабораторный практикум

Лабораторная работа 1

Цель работы: Алгоритмы циклической структуры. Алгоритм «удаление путем сдвига». Алгоритмы поиска суммы, количества, среднего арифметического элементов массива.

1. Вычислить массив R[1:n] в соответствии с формулой:

4) r [ i ]= 0,3 cos ( 2 ax−i^2h )

**II**. Из вычисленного массива R удалить:

9) все положительные элементы, расположенные после первого минимального элемента.

**III.** В полученном массиве R[1:k], где k – число элементов, оставшихся после удаления, подсчитать среднее арифметическое расположенных:

4) между первым максимальным и минимальным по модулю элементами.

**Постановка задачи**

**Дано**:

1. n-цел., x, h, a-вещ.
2. Нет входных данных
3. Нет входных данных

**Результат**:

1. R[1:n]-вещ.
2. R[1:k]-вещ. или сообщение «Минимальный элемент расположен в конце массива.» или сообщение «После минимального числа нет положительных элементов»
3. sr-вещ. или сообщение «Нет среднего значения» или сообщение «максимальные и минимальные по модулю элементы находятся рядом или совпадают».

**При**: *n* ∈ℕ*, n ≤ lmax.*

**Связь:**

1. См. формулу в условии
2. ∃n1: ∀i = 1, n

*R*[*n1*] *R*[*i*]

*min = R*[*n*1]

∃t:t = 1, *n*1 - 1;*R*[*t*] = *min*

∃*i: i* = n1, n

R[*i*] ≥ 0

3. ∃q: ∀f = 1,k

| R[q] | | R[f] |

min\_mod = R[q]

∃t1: t = 1, q - 1; R[t] = min\_mod

∃q1: ∀f1 = 1,k

| R[q1] | | R[f1] |

max\_mod = R[q1]

∃t2: t2 = 1, q1 - 1; R[t2] = max\_mod

c = max(q,q1); b = min(q,q1)

**Метод решения**

2. n1 = R[1]

Изображение k = n

*для i = 2,k*

*n1* = i, если n1 < R[i]

Изображение

для i = n1 + 1, n

i = n1+1; k = k + 1, если r[i] = r[k] и r[i] ≤ 0

3. mp = 1Изображение

*для i = 2,k*

*mp = i, если |r*[i]*| < |r*[mp]|

Изображениеmaxp = 1

*для* *i = 2,k-1*

*max = I, если |r*[*i*]| > |r[max]|

Изображение

*first = mp, если mp < maxp, иначе first = maxp*

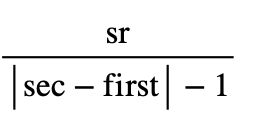
*sec = mp, если mp > maxp, иначе sec = maxp*

Изображение

*для i = first + 1, sec - 1*

*sr = sr + r*[i]

sr =



**Внешняя спецификация**

Задание 1

Введите длину массива R от 1 до <<lmax>>:

до и n ≤ max

Введите x, h, a:

< x >< h >< a >

Массив r из << n >> элементов:

<<*r* [ 1 ]>><<*r* [ 2 ]>> … <<*r* [ n ]>>

Задание 2

При k = n

Минимальный элемент расположен в конце массива.

Массив r из << k >> элементов:

<<*r* [ 1 ]>><<*r* [ 2 ]>> … <<*r* [ k ]>>

Иначе

При n1 = n

После минимального числа нет положительных элементов.

Массив r из << k >> элементов:

<<*r* [ 1 ]>><<*r* [ 2 ]>> … <<*r* [ k ]>>

Иначе

Массив r из << k >> элементов:

<<*r* [ 1 ]>><<*r* [ 2 ]>> … <<*r* [ k ]>>

Задание 3

При |mp – maxp| = 1

Минимальные и максимальные по модулю числа расположены рядом.

Не удалось посчитать среднее арифметическое.

Иначе

При minp = Правда

все числа одинаковы по модулю

Не удалось посчитать среднее арифметическое.

Иначе

sr = << sr >>

**Описание алгоритма на псевдокоде**

алг <<Лабораторная работа №1>>

нач

вывод (<<Лабораторная работа №1. Задание 1>>)

вывод (<<Введите длину массива R 1 до>>, lmax )

цикл

ввод ()

до n>0 и n<lmax // ввод до тех пор пока не подойдет условию

кц

вывод (<<Введите a, h, x: >>)

ввод(a, h, x)

цикл от i=0 до n // выполнение действия для каждого элемента массива

r [ i ]= 0,3 cos ( 2 ax−i^2h ) // формула из задания

кц

вывод(<<Массив r из >>, n, << элементов: >>)

вывод(r [ 1;n ])

вывод( << Задание 2 >> )

n1 = 1// первый минимальный элемент

цикл i:= 2 до n //поиск начиная со второго

если r[i] < r[n1] то

n1:=i

все

кц

если n1 = n то // если минимальный элемент имеет индекс последнего элемента

вывод (<<Минимальный элемент расположен в конце массива.>>)

иначе

k:=n1+1

цикл от i:=n1+1 до n - 1

если r[i] < 0 то

r[k]:=r[i]

k:=k+1

все

кц

если n = k-1 то

вывод(<<После минимального числа нет положительных элементов.>>)

иначе

вывод(<<Массив r из >>, k, << элементов: >>)

вывод(r[0;k])

все

все

вывод (<<Задание 3>>)

maxp := 1

Цикл от i:=2 до k

если | r[i] | > | r[maxp] | то

maxp = i

все

кц

mp := 1

minp := Правда

Цикл от i:=2 до k

если | r[i] | < | r[mp] | то

mp = I

minp = Ложь

все

кц

если minp = Правда то

вывод (<<Все числа одинаковы по модулю>>)

вывод (<< Не удалось посчитать среднее арифметическое. >>)

иначе

если | mp - maxp | = 1 то

вывод (<<Минимальные и максимальные по модулю числа расположены рядом.>>)

вывод (<< Не удалось посчитать среднее арифметическое. >>)

иначе

если mp > maxp то

maxi := mp

mini := maxp

иначе

maxi := maxp

mini := mp

все

цикл от i:= mini + 1 до maxi – 1

sr:= sr + r[i]

sr:= sr/(maxi -mini -1)

кц

вывод (<<Среднее арифметическое =>>,sr)

все

все

кон

**Листинг программы**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define lmax 200

#include <locale.h>

#include <stdbool.h>

int main()

{

bool minp = true;

int n, k, n1, mp, maxp, maxi, mini;

float x, h, a, r[lmax], sr;

printf("Лабораторная работа №1\n");

printf("Задание 1\n");

while (n < 1 || n > lmax)// ввод до того момента пока не выполнятся условия

{

printf("Введите длину массива R от 1 до %d: ", lmax);

scanf("%d", &n);

}

printf("Введите a,h,x:\n");

scanf("%f", &a);

scanf("%f", &h);

scanf("%f", &x);

for (int i = 0; i < n; i++)

r[i] = 0.3 \* cos(2 \* a \* x - pow(i,2) \* h);

printf("Массив R из %d элементов:\n", n);

for (int i = 0; i < n; i++)

printf("%8.3f", r[i]);

printf("\n");

printf("Задание 2\n");

n1 = 0;

for (int i = 1; i < n; i++)// поиск

{

if (r[i] < r[n1])

n1 = i;

}

if (n1 == n)

printf("Минимальный элемент расположен в конце массива.\n");

else

{

k = n1 + 1;

for (int i = (n1 + 1);i < n; i++)

{

if (r[i] < 0)

{

r[k] = r[i];

k = k + 1;

}

}

if (n == k - 1)

printf("После минимального числа нет положительных элементов.\n");

else

{

printf("Массив из %d элементов: \n", k-1);

}

for (int i = 0; i < k; i++)

{

printf("%8.3f", r[i]);

}

}

printf("\n");

printf("Задание 3\n");

maxp = 0;

mp = 0;

minp = -1;

for (int i = 1; i < k; i++)

{

if (fabs(r[i]) > fabs(r[maxp]))

{

maxp = i;

}

}

for (int i = 1; i < k; i++)

{

if (fabs(r[i]) < fabs(r[mp]))

{

mp = i;

minp = false;

}

}

if (minp == true)

{

printf("Все числа одинаковы по модулю\n");

printf("Не удалось посчитать среднее арифметическое\n");

}

else if (fabs(mp - maxp) == 1)

{

printf("Первые минимальные и максимальные по модулю числа расположены рядом.\n");

}

else if (minp == false)

{

if (mp > maxp)

{

maxi = mp;

mini = maxp;

}

else

{

maxi = maxp;

mini = mp;

}

for (int i = mini + 1; i < maxi; i++)

sr = sr + r[i];

sr = sr / (maxi - mini - 1);

printf("Среднее арифметическое:\n");

printf("%8.3f", sr);

}

}

**ТЕСТОВЫЕ ДАННЫЕ**

|  |  |  |
| --- | --- | --- |
| № | Исходные данные | Результаты |
| 1 | n = 20  a = 2  h = 5  x = 23 | R = { -0.188 0.171 -0.290 -0.298 0.253 -0.004 0.300}  R = {-0.188 0.171 -0.290 -0.298 -0.004}  Первые минимальные и максимальные по модулю числа расположены рядом. |
| 2 | n = 1  a = 2  h = 3  x = 4 | R = {-0.287}  После минимального числа нет положительных элементов.  R = { -0.287}  Все числа одинаковы по модулю одинаковы по модулю  Не удалось посчитать среднее арифметическое |
| 3 | n = 5  a = 98  h = 23  x = 45 | R = {-0.006 0.257 0.238 0.095 0.132}  R = {-0.006}  Все числа одинаковы по модулю одинаковы по модулю  Не удалось посчитать среднее арифметическое |
| 4 | n=8  x=2;  h=4;  a=8 | R = {0.250 -0.289 -0.287 -0.196 0.250 0.132 0.137 0.241}  R = {0.250 -0.289 -0.287 -0.196}  Среднее арифметическое:  -0.287 |

**Листинг программы**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define lmax 4500

#include <locale.h>

#include <stdbool.h>

**int** main()

{

setlocale(LC\_ALL, "Rus");

**bool** minp = **true**;

**int** n, k, mp, maxp, maxi, mini;

**float** x, h, a, r[lmax], sr, e, n1;

printf("Лабораторная работа №1\n");

printf("Задание 1\n");

**while** (n < 1 || n > lmax)

{

printf("Введите длину массива R от 1 до %d: ", lmax);

scanf("%d", &n);

}

printf("Введите a,h,x:\n");

scanf("%f", &a);

scanf("%f", &h);

scanf("%f", &x);

**for** (**int** i = 0; i < n; i++)

r[i] = 0.3 \* cos(2 \* a \* x - pow(i,2) \* h);

printf("Массив R из %d элементов:\n", n);

**for** (**int** i = 0; i < n; i++)

printf("%8.3f", r[i]);

printf("\n");

printf("Задание 2\n");

printf("Введите Е: ");

scanf("%f", &e);

n1 = 0;

**for** (**int** i = 0; i<n; i++)

{

**if** (r[i] < 0)

n1 = r[i];

}

k = 0;

**if** (n1 == 0)

printf("Нет отрицательных элементов");

**else**

{

**for** (**int** i = 0; i < n; i++)

{

**if** (fabs(r[i] - n1) > e)

{

r[k] = r[i];

k = k + 1;

}

}

**if** (k == 0)

printf("Полное удаление");

**else**

printf("Массив из %d элементов: \n", k);

**for** (**int** i = 0; i < k; i++)

{

printf("%8.4f", r[i]);

}

}

}

**Тестовые данны**

**Полное удаление:**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Нет удаления:**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Удаление соответственно заданной Е:**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**