

**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

Раздел	Макс. оценка	Итоговая 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	

Отчет по лабораторной работе № 1

Студент: Чапайкин Арсений Георгиевич

Группа: БИВ242

Вариант: № 171 (4, 1, 2)

Руководитель: Ерохина Е.А.

Оценка:

Дата сдачи:

Задание

1. Вычислить массив $R[1:n]$ в соответствии с формулой $R[i] = 0.3 \cos(2ax - i^2h)$.
2. Из вычисленного массива R удалить все элементы, удовлетворяющие условию $|R[i]| < 0.7$.
3. В полученном массиве $R[1:k]$, где k — число элементов, оставшихся после удаления, подсчитать среднее арифметическое элементов, расположенных после первого максимального элемента.

Требования к выполнению работы

1. Все три части оформить в одной программе.
2. Необходима проверка допустимости исходных данных.
3. В наборе тестов должны быть примеры, дающие все возможные варианты результатов.
4. Удаление элементов массива должно производиться путем сдвига, т.е. без формирования нового массива и с использованием одного цикла.
5. При отсутствии результата выполнения задания необходимо выводить на экран соответствующие сообщения.

Постановка задачи

Дано:

1. $n \in \mathbb{Z}_+$; $x, h, a \in \mathbb{R}$.
2. Нет входных данных.
3. Нет входных данных.

Результат:

1. $R[1:n]$ (массив вещественных чисел) или сообщение «Memory allocation error».
2. $R[1:k]$ (массив вещественных чисел) или сообщение «Array is empty or not initialized».
3. M (вещественное число), или сообщение «Array is empty or not initialized», или сообщение «There are no elements after first maximum».

При: $n \in \mathbb{N}$, $n \leq \text{MAXN}$.

Связь:

1. См. формулу в условии.
2. Для $i = \overline{1 \dots n} : R[i] > 0.7$ требуется найти $j = \min\{v \mid v \in [1 \dots i] \wedge (R[v] \leq 0.7 \vee v = i)\}$ и присвоить $R[j] := R[i]$.
3. Положим $ind = \min\{i \mid \forall j \in [1 \dots k] R[j] \leq R[i]\}$. Требуется найти $M = \frac{1}{k - ind - 1} \sum_{i=ind+1}^k R[i]$.

Метод решения

1.
$$\begin{cases} \text{Для } i = \overline{1 \dots n} \\ R[i] = 0.3 \cos(2ax - i^2h) \end{cases}$$
2. $pos = 0$
$$\begin{cases} \text{Для } i = \overline{1 \dots n} \\ (R[pos] = r[i], pos = pos + 1), \text{ если } |R[i]| > 0.7 \end{cases}$$
3. $fst = 0$
 $sum = 0$
$$\begin{cases} \text{Для } i = \overline{1 \dots k} \\ (fst = i, sum = 0), \text{ если } R[i] > R[fst] \\ (sum = sum + R[i]) \text{ в противном случае} \end{cases}$$
$$M = \frac{sum}{k - fst - 1}$$

Внешняя спецификация

Описание алгоритма на псевдокоде

Листинг программы

```
1 #include <stdio.h>
2 #include <math.h>
3 #include <stdlib.h>
4
5 const MAX_N = 100;
6
7 size_t n;
8 float x, h, a, *r = NULL;
9
10 int task1() {
11     printf("Input format: 'n x h a'.\n");
12     printf("Variables must be of the following types: \
13     n(positive integer), x(float), h(float), a(float).\n");
14
15     int counter = 0;
16     while (scanf("%zu%f%f%f", &n, &x, &h, &a) != 4 || n == 0 || n > MAX_N) {
17         if (++counter > 5) {
18             printf("Too many attempts.\n");
19             return 1;
20         }
21         printf("Input format is incorrect. Try again.\n");
22     }
23
24     r = (float*)calloc(n, sizeof(float));
25     if (r == NULL) {
26         printf("Memory allocation error.\n");
27         return 1;
28     }
29
30     for (int i = 1; i <= n; i++) {
31         r[i - 1] = 0.3 * cos(2 * a * x - i * i * h);
32     }
33     return 0;
34 }
35
36 int task2() {
37     if (r == NULL) {
38         printf("Array is empty or not initialized.\n");
39         return 1;
40     }
41
42     int last_free_pos = 0;
43     for (int i = 0; i < n; i++) {
44         if (abs(r[i]) > 0.7) {
45             r[last_free_pos++] = r[i];
46         }
47     }
48     n = last_free_pos;
49     r = (float*)realloc(r, n * sizeof(float));
50     return 0;
51 }
52
53 int task3() {
54     if (r == NULL) {
55         printf("Array is empty or not initialized.\n");
56         return 1;
57     }
```

```

58     if (n == 1) {
59         printf("There are no elements after first maximum.\n");
60         return 1;
61     }
62
63     int first_maximum_index = 0;
64     float sum = 0;
65
66     for (int i = 0; i < n; i++) {
67         if (r[i] > r[first_maximum_index]) {
68             first_maximum_index = i;
69             sum = 0;
70         } else {
71             sum += r[i];
72         }
73     }
74     printf("%f", sum / (n - first_maximum_index - 1));
75     return 0;
76 }
77
78 int main() {
79     if (task1()) {
80         printf("Task 1 failed.\n");
81         return 1;
82     }
83     if (task2()) {
84         printf("Task 2 failed.\n");
85         return 1;
86     }
87     if (task3()) {
88         printf("Task 3 failed.\n");
89         return 1;
90     }
91     if (r != NULL) {
92         free(r);
93     }
94     return 0;
95 }

```

Распечатка тестов к программе и результатов