|  |  |  |
| --- | --- | --- |
| **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 |  |

**Студент: Андреев Глеб Андреевич**

**Группа: \_\_\_\_\_\_\_\_\_БИВ254\_\_\_\_\_\_\_\_**

**Вариант: \_\_\_\_\_№252 (15, 3)\_\_\_\_**

**Руководитель: Альбатша Ахмад Мухаммад Хусайн**

**Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дата сдачи: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Оглавление**

[**Задание** 3](#_Toc210755941)

[**Метод решения задачи** 5](#_Toc210755942)

[**Внешняя спецификация** 6](#_Toc210755943)

[**Задача 1** 6](#_Toc210755944)

[**Задача 2** 7](#_Toc210755945)

[**Описание алгоритма на псевдокоде** 8](#_Toc210755946)

[**Задача 1** 8](#_Toc210755947)

[**Задача 2** 10](#_Toc210755948)

[**Листинг программы** 11](#_Toc210755949)

[**Задача 1** 11](#_Toc210755950)

[**Задача 2** 14](#_Toc210755951)

[**Распечатка тестов к программе и результатов** 16](#_Toc210755952)

# **Задание**

**ЦЕЛЬ РАБОТЫ:** Алгоритмы с досрочным выходом из цикла. Алгоритмы обработки целых чисел.

*1.* (15) Даны целочисленная матрица Q[1:n, 1:m] и целочисленный массив Z[1:k]. Упорядочить по убыванию те строки матрицы Q, сумма элементов которых совпадает с одним из элементов массива Z.

**ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ РАБОТЫ.**

1. Необходима проверка допустимости исходных данных.

2. Необходимо использование алгоритмов с досрочным выходом из цикла. При этом используются либо цикл с предусловием, либо цикл с постусловием.

*2.* (3) Дан массив целых положительных чисел. Сформировать новый массив, содержащий произведения цифр каждого элемента исходного массива.

**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.

2. При вычислении результата необходимо использовать целый тип. Использование строк при решении данной задачи недопустимо.**Постановка задачи**

Дано:

1. Q[1:n, 1:m] – целочисленная матрица, Z[1:n] – целочисленный массив
2. A[1:n] – цел.

Результат:

1. Q[1:n][1:m] – целочисленная матрица
2. B[1:n] – целочисленный массив

При:

Связь:

# **Метод решения задачи**

# **Внешняя спецификация**

### **Задача 1**

{ }\* до и

{ }\* до и

|  |
| --- |
| Введите элементы матрицы |

< Q[1][1] > < Q[1][2] > … < Q[n][m] >

|  |
| --- |
| Исходная матрица Q:  << Q[1][1] >> << Q[1][2] >> … << Q[n][m] >>  Введите длину массива Z |

{ }\* до и

|  |
| --- |
| Введите элементы массива Z |

< Z[1] > < Z [2] > … < Z[k] >

|  |
| --- |
| Преобразованная матрица Q:  << Q[1][1] >> << Q[1][2] >> … << Q[n][m] >> |

# **Задача 2**

{ }\* до и

< A[1] > < A [2] > … < A[n] >

|  |
| --- |
| Исходная массив A:  << A[1] >> << A[2] >> … << A[n] >>  Преобразованный массив B (произведение цифр элементов массива A)  << B[1] >> << B[2] >> … << B[n] >> |

# **Описание алгоритма на псевдокоде**

# **Задача 1**

алг «Лабораторная работа №2, задача №1»

нач

lmax := 100

вывод("Лабораторная работа №2, задача №1")

цикл пока n < 0 или n > lmax

вывод("Введите количество строк матрицы Q (от 1 до ", lmax, "): ")

ввод(n)

кц

цикл пока m < 0 или m >= lmax

вывод("Введите количество столбцов матрицы Q (от 1 до ", lmax, "): ")

ввод(m)

кц

вывод("Введите элементы матрицы:")

ввод(Q[1:n][1:m])

вывод("Исходная матрица Q:")

вывод(Q[1:n][1:m])

цикл пока k < 0 или k >= lmax

вывод("Введите длину массива Z (от 1 до ", lmax, "): ")

ввод(k)

кц

вывод("Введите элементы массива Z:")

ввод(Z[1:k])

цикл от i:=1 до n

lineSum := 0

цикл от j:=1 до m

lineSum := lineSum + Q[i][j]

кц

g := 1

цикл пока g <= k и lineSum ≠ Z[g]

g := g + 1

кц

если g <= k то

вывод("Сумма элементов ", i, "-й строки равна ", lineSum,

", совпадает с Z[", g, "] = ", Z[g])

цикл от j:=1 до m - 1

цикл от p:=j + 1 до m

если Q[i][j] < Q[i][p] то

swap := Q[i][j]

Q[i][j] := Q[i][p]

Q[i][p] := swap

кц

кц

кц

иначе

вывод("Сумма элементов ", i, "-й строки равна ", lineSum,", не совпадает ни с одним элементом массива Z")

все

кц

вывод("Преобразованная матрица Q:")

вывод(Q[1:n][1:m])

кон

# **Задача 2**

алг «Лабораторная работа №2, задача №2»

нач

lmax := 100

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

цикл пока n < 0 или n > lmax

вывод("Введите количество элементов массива A (от 1 до ", lmax, "): ")

ввод(n)

кц

вывод("Введите элементы массива A:")

ввод(A[1:n])

цикл от i:=1 до n

prod := 1

x := A[i]

цикл пока x > 0

prod := prod \* (x % 10)

x := x / 10

кц

B[i] := prod

кц

вывод("Исходный массив A:")

вывод(A[1:n])

вывод("Преобразованный массив B (произведения цифр элементов массива A):")

вывод(B[1:n])

кон

# **Листинг программы**

# **Задача 1**

#include <stdio.h>

#include <limits.h>

#define lmax 100

**int** **main**()

{

**int** m, n, Q[lmax][lmax];

**char** check;

printf("Лабораторная работа №2, задача №1**\n**");

**do** {

printf("Введите количество строк матрицы Q (от 1 до %d): ", lmax);

scanf("%d", &n);

check = getchar();

**if** (check != '\n' || n <= **0**) {

printf("Некорректнрый ввод (m должно быть натуральным).**\n**");

n = **0**;

**while** (check != '\n' && check != EOF) {

check = getchar();

}

}

} **while** (n <= **0** || n > lmax);

**do** {

printf("Введите количество столбцов матрицы Q (от 1 до %d): ", lmax);

scanf("%d", &m);

check = getchar();

**if** (check != '\n' || m <= **0**) {

printf("Некорректнрый ввод (n должно быть натуральным).**\n**");

m = **0**;

**while** (check != '\n' && check != EOF) {

check = getchar();

}

}

} **while** (m <= **0** || m > lmax);

**int** temp;

printf("Введите элементы матрицы:**\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("Некорректный ввод.**\n**");

temp = INT\_MAX;

**while** (check != '\n' && check != EOF) {

check = getchar();

}

}

} **while** (check != '\n' || temp == INT\_MAX);

Q[i][j] = temp;

}

}

printf("Исходная матрица Q:**\n**");

**for** (**int** i = **1**; i <= n; i++) {

**for** (**int** j = **1**; j <= m; j++) {

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

}

printf("**\n**");

}

**int** Z[lmax], k;

**do** {

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

scanf("%d", &k);

check = getchar();

**if** (check != '\n' || k <= **0**) {

printf("Некорректнрый ввод (k должно быть натуральным).**\n**");

k = **0**;

**while** (check != '\n' && check != EOF) {

check = getchar();

}

}

} **while** (k <= **0** || k > lmax);

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

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

**do** {

scanf("%d", &temp);

check = getchar();

**if** (check != '\n') {

printf("Некорректный ввод.**\n**");

temp = INT\_MAX;

**while** (check != '\n' && check != EOF) {

check = getchar();

}

}

} **while** (check != '\n' || temp == INT\_MAX);

Z[i] = temp;

}

**int** lineSum = **0**, g = **1**;

**for** (**int** i = **1**; i <= n; i++) {

lineSum = **0**;

**for** (**int** j = **1**; j <= m; j++) {

lineSum += Q[i][j];

}

g = **1**;

**while** (g <= k && lineSum != Z[g]) {

g++;

}

**if** (g <= k) {

printf("Сумма элементов %d-й строки матрицы Q равна %d, что совпадает с элементом Z[%d] = %d**\n**", i, lineSum, g, Z[g]);

**for** (**int** j = **1**; j < m; j++) {

**for** (**int** p = j + **1**; p <= m; p++) {

**if** (Q[i][j] < Q[i][p]) {

**int** swap = Q[i][j];

Q[i][j] = Q[i][p];

Q[i][p] = swap;

}

}

}

}

**else** {

printf("Сумма элементов %d-й строки матрицы Q равна %d, что не совпадает ни с одним элементом массива Z**\n**", i, lineSum);

}

}

printf("Преобразованная матрица Q:**\n**");

**for** (**int** i = **1**; i <= n; i++) {

**for** (**int** j = **1**; j <= m; j++) {

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

}

printf("**\n**");

}

**return** **0**;

}

# **Задача 2**

#include <stdio.h>

#include <limits.h>

#define lmax 100

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

**int** n, A[lmax];

**char** check;

printf("Лабораторная работа №2, задача №2**\n**");

**do** {

printf("Введите количество элементов массива A (от 1 до %d): ", lmax);

scanf("%d", &n);

check = getchar();

**if** (check != '\n' || n <= **0**) {

printf("Некорректнрый ввод (n должно быть натуральным).**\n**");

n = **0**;

**while** (check != '\n' && check != EOF) {

check = getchar();

}

}

} **while** (n <= **0** || n > lmax);

**int** temp;

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

**for** (**int** i = **1**; i <= n; i++) {

**do** {

scanf("%d", &temp);

check = getchar();

**if** (check != '\n' || temp <= **0**) {

printf("Некорректный ввод (элемент должен быть натуральным).**\n**");

temp = INT\_MAX;

**while** (check != '\n' && check != EOF) {

check = getchar();

}

}

} **while** (check != '\n' || temp == INT\_MAX);

A[i] = temp;

}

**int** B[lmax];

**for** (**int** i = **1**; i <= n; i++) {

**int** prod = **1**, x = A[i];

**while** (x > **0**) {

prod \*= (x % **10**);

x /= **10**;

}

B[i] = prod;

}

printf("Исходный массив A:**\n**");

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

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

printf("**\n**Преобразованный массив B (произведения цифр элементов массива A):**\n**");

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

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

}

# **Распечатка тестов к программе и результатов**

# **Задача 1**

|  |  |  |
| --- | --- | --- |
| **№** | **Исходные данные** | **Результаты** |
| 1 | n=2; m=2  Q =  k=2  Z = | Исходная матрица Q:  Сумма элементов 1-й строки матрицы Q равна 3, что совпадает с элементом Z[1] = 3  Сумма элементов 2-й строки матрицы Q равна 7, что совпадает с элементом Z[2] = 7  Преобразованная матрица Q: |
| 2 | n=3; m=4  Q =  k=1  Z = | Исходная матрица Q:  Сумма элементов 1-й строки матрицы Q равна 10, что не совпадает ни с одним элементом массива Z  Сумма элементов 2-й строки матрицы Q равна 26, что не совпадает ни с одним элементом массива Z  Сумма элементов 3-й строки матрицы Q равна 42, что не совпадает ни с одним элементом массива Z  Преобразованная матрица Q: |
| 3 | n=3; m=3  Q =  k=3  Z = | Исходная матрица Q:  Сумма элементов 1-й строки матрицы Q равна 6, что не совпадает ни с одним элементом массива Z  Сумма элементов 2-й строки матрицы Q равна 15, что совпадает с элементом Z[3] = 15  Сумма элементов 3-й строки матрицы Q равна 24, что не совпадает ни с одним элементом массива Z  Преобразованная матрица Q: |

# **Задача 2**

|  |  |  |
| --- | --- | --- |
| **№** | **Исходные данные** | **Результаты** |
| 1 | n=3  A=[123, 45, 6] | Исходный массив A: [123 45 6]  Преобразованный массив B (произведения цифр элементов массива A): [6, 30, 6] |
| 2 | n=1  A=[100] | Исходный массив A: [100]  Преобразованный массив B (произведения цифр элементов массива A): [0] |
| 3 | n=4  A=[7, 23, 101, 456] | Исходный массив A: [7, 23, 101, 456]  Преобразованный массив B (произведения цифр элементов массива A): [7, 6, 0, 120] |