**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**

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

**ОТЧЕТ**

**по лабораторной работе №2**

**Студент: Иванилов Леонид Михайлович**

**Группа: БИВ231**

**Вариант: №143 (8,4)**

**Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Оценка: \_\_\_\_\_\_\_\_\_\_**

**Дата сдачи\_\_\_\_\_\_\_\_\_\_**

**МОСКВА 2023**

**ЗАДАНИЯ**

I. *(8)Даны целочисленная матрица A[1:n, 1:m] и целочисленный массив B[1:k]. Упорядочить по убыванию те столбцы матрицы A, все элементы которых присутствуют в массиве B.*

*II.(4) Сформировать новый массив, содержащий все элементы исходного массива, являющиеся палиндромами. Палиндромом называется число, в котором порядок следования цифр одинаковый как при чтении справа налево, так и слева направо. Например, 123321.*

Оглавление

[Задание 2](#_Toc147349208)

[Постановка задачи 3](#_Toc147349209)

[Метод решения 4](#_Toc147349210)

[Внешняя спецификация 5](#_Toc147349211)

[Описание алгоритма на псевдокоде 7](#_Toc147349212)

[Листинг программы №1 9](#_Toc147349213)

[Листинг программы №2 12](#_Toc147349214)

[Распечатка тестов к программе и результатов 14](#_Toc147349215)

**Постановка задачи № 1.**

**Дано:**

na, ma, nb , A[1:na, 1:ma], B[1:nb] – цел.

Результат:

A’[1:na, 1:ma] – цел. или <<Матрица без перестановки>>

При:

na, ma , nb ∈ , ma ≤ lmax, na ≤ lmax, nb ≤ lmax

**Связь:** :

**Метод решения № 1.**

**1.**

**Внешняя спецификация № 1.**

Задание 1.

Введите число строк и столбцов матрицы.

{<na> <ma>}\* до na>0 и ma > 0; na≤lmax и ma≤lmax; na = [na]; ma = [ma]

Введите матрицу <<na>> на <<ma>>

для i= , j= {<X[i][j]>}\* до X[i][j] ≥ 0; X[i][j] = [X[i][j]]

Введите длину массива

{<\nb>}\* до nb > 0; nb ≤ lmax;

Введите массив b из <<nb>> элементов

для i= {<B[i]>}\* до B[i] ≥ 0; nb = [nb]

**Описание алгоритма на псевдокоде задание № 1**

Алг «Лабораторная работа №2»​

нач​

вывод(<<Введите число строк и столбцов матрицы>>)

цикл

ввод(na, ma)

 до ma>0 и ma≤lmax и ma=[ma] и na>0 и na≤lmax и na=[na]

кц

вывод(<<Введите матрицу>>, na, <<на>>, ma)

ввод(A[1:na,1:ma])

вывод(<<Введите длину массива b>>)

 цикл

ввод(nb)

  до nb>0 и nb≤lmax и nb=[nb]

кц

ввод(B[1:nb])

цикл от i = 1 до ma

flag1 := 1

j := 1

цикл-пока flag1 = 1и j <= na

flag1 := 0

b1 := 1

цикл-пока flag1 = 0 и b1 <= na

если A[j][i] = B[b1] то

flag1 := 1

все

b1 := b1 + 1

кц

j := j + 1

кц

если flag1 = 1 то

цикл от j := 1 до na - 1

цикл от k := j + 1 до na

если a[j][i] < a[k][i] то

ba := a[j][i]

a[j][i] := a[k][i]

a[k][i] := ba

count := count + 1

все

кц

кц

кц

если сount 0 то

вывод(<<Матрица после перестановки>>)

цикл от i = 1 до na

цикл от j = 1 до ma

вывод(a[i][j])

кц

кц

иначе

вывод(<<Матрица без перестановки>>)

все

кон

**Листинг программы №1**

#include <stdio.h>

#define lmax 20

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

{**int** a[lmax][lmax], b[lmax], na, ma, q, nb, i, j = 0, k, flag1, mk, ba, count = 0;

mk = 0;

printf("Введите число строк и столбцов матрицы a\n");

**do**{k=scanf("%d%d",&na,&ma);

**while**(getchar()!='\n') ;

}**while**(na<=0||ma<=0||na>=lmax||ma>=lmax||k!=2);

printf("Введите матрицу %d на %d\n",na,ma);

**for**( i=1;i<=na;i++)

**for**( j=1;j<=ma;j++)

**do** {

q = scanf("%d", &a[i][j]);

**if** (q != 1) {

printf("Введите заново\n");

**while** (getchar() != '\n');

}

} **while** (q != 1);

printf("Введите длину массива b\n");

**do**{k=scanf("%d",&nb);

**while**(getchar()!='\n') ;

} **while**(nb>=lmax||nb<0||k!=1);

printf(" Введите массив b из %d элементов\n", nb);

**for**( i=1;i<=nb;i++)

**do** {

q = scanf("%d", &b[i]);

**if** (q != 1) {

printf("Введите заново\n");

**while** (getchar() != '\n');

}

} **while** (q != 1);

**for** (i = 1; i <= ma; i ++)

{

flag1 = 1;

**for** (j = 1; j <= na&&flag1==1; j ++)

{

flag1 = 0;

**for**(**int** b1 = 0; b1 <= na&&flag1==0;b1++)

{

**if** (a[j][i] == b[b1])

flag1 = 1;

}

}

**if** (flag1 == 1)

{

**for** (**int** j = 1; j < na; j++)

**for**( k=j+1;k<=na;k++)

**if** (a[j][i] < a[k][i])

{

ba = a[j][i];

a[j][i] = a[k][i];

a[k][i] = ba;

count+=1;

}

}

}

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

printf("Матрица после перестановки\n");

**for**( i=1;i<=na;i++)

{

**for**( j=1;j<=ma;j++)

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

printf("\n");

}

}

**else**

printf("Матрица без перестановки\n");

**return** 0;

}

}**Тестовые данные для задачи №1**

|  |  |  |
| --- | --- | --- |
| № | Исходные данные | Результаты |
| 1 | na = 5  ma = 1  nb = 5  A = [1]  [2]  [3]  [4]  [5]  B = {1, 2, 3, 4, 5} | Матрица после перестановки  A’= [5]  [4]  [3]  [2]  [1] |
| 2 | na = 5  ma = 2  nb = 5  A = [1] [2]  [2] [3]  [3] [4]  [4] [5]  [5] [6]  B = {1, 2, 3, 4, 5} | Матрица после перестановки  A’ = [5] [2]  [4] [3]  [3] [4]  [2] [5]  [1] [6] |

|  |  |  |
| --- | --- | --- |
| 3 | na = 5  ma = 2  nb = 5  A = [5] [2]  [4] [3]  [3] [4]  [2] [5]  [1] [6]  B = {1, 2, 3, 4, 5} | Матрица без перестановки |

**Постановка задачи № 2**

**Дано:**

B[1:nb] – цел. , nb - цел

**Результат:**

B1[1:b1] – цел. или <<Пустой массив>>

**При:**

nb ∈ , nb ≤ lmax; b1 ≤ lmax;

**Связь:** k = 1

*t =*

**Метод решения №2.**

**Внешняя спецификация № 2.**

Введите длину массива.

{<nb>}\* до nb, nb = [nb], nb > 0, nb ≤ lmax.

Введите массив b из <<nb>> элементов

для i= {<B[i]>}\* до B[i] ≥ 0; nb = [nb]

**Описания алгоритма на псевдокоде № 2**

Алг<<Новый массив>>

Нач

Вывод(<<Введите длину массива b>>)

Ввод(nb)

Вывод(<<Введите массив b из>>, nb, <<элементов>>)

Ввод(b[1:nb])

kk := 1

Цикл от i = 1 до nb

n := n1

n := b[i]

*l* := 0

цикл

*l* := *l* + 1

*n := n \ 10*

*до n 0*

*flag1 := 1*

*цикл-пока flag = 1 и l > 1*

*st := 10 ^ (l – 1)*

*если (n1 % 10) (n1 \ st) то*

*flag1 := 0*

*иначе*

*l := l - 2*

*n1 := n1 – ((n1\st) \* st)*

*n1 := n1 / 10*

*все*

*кц*

*если flag1 = 1то*

*b1[kk] := b[i]*

*kk := kk + 1*

*все*

*кц*

*если kk = 1 то*

*вывод(<<Пустой массив>>)*

*иначе*

*вывод(<<Новый массив>>)*

*вывод(b1[1:kk-1])   
все*

*кон*

**Листинг программы № 2**

#include <stdio.h>

#define lmax 20

#include <math.h>

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

{

**int** nb, b[lmax], b1[lmax], k, q = 0, i, flag1 = 1, n, n1, kk = 1;

printf("Введите длину массива b\n");

**do**{k=scanf("%d",&nb);

**while**(getchar()!='\n');

} **while**(nb>=lmax||nb<0||k!=1);

printf("Введите массив b из %d элементов\n", nb);

**for**( i=1;i<=nb;i++)

**do** {

q = scanf("%d", &b[i]);

**if** (q != 1) {

printf("Введите заново\n");

**while** (getchar() != '\n');

}

} **while** (q != 1);

**for** (i = 1; i <= nb; i++)

{

n = n1 = b[i];

**int** l = 0;

**do** {

l++;

n /= 10;

} **while**(n);

flag1 = 1;

**while** (flag1 == 1 && l > 1)

{

**int** st = (pow(10, l - 1));

**if** ((n1 % 10) != n1 / st)

flag1 = 0;

**else**{

l-=2;

n1-= (n1/st) \* st;

n1/=10;

}

}

**if** (flag1 == 1)

{

b1[kk] = b[i];

kk+=1;

}

}

**if** (kk == 1)

{

printf("Пустой массив\n");

}

**else**

{

printf("Новый массив\n");

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

printf("%d ", b1[i]);

}

**return** 0;

}

**Тестовые данные**

|  |  |  |
| --- | --- | --- |
| № | Исходные данные | Результаты |
| 1 | nb = 5  B={111, 545, 11, 8, 456} | B={111, 545, 11, 8} |
| 2 | nb = 4  B={123, 12332, 12, 459, 343} | B={343} |
| 3 | nb = 1  B={12345} | Пустой массив |

**Задача № 3.**

**Листинг программы**

#include <stdio.h>

#define lmax 20

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

{**int** a[lmax][lmax], b[lmax], max, na, ma, q, nb, i, j = 0, k, flag1, mk, ba,actual\_row = 0, count = 0;

**float** r, r1;

mk = 0;

**do**{

printf("Введите число строк и столбцов матрицы a\n");

k=scanf("%f%f",&r,&r1);

**if**(k == 2)

{

na = (**int**)r;

ma = (**int**)r1;

}

**while**(getchar()!='\n') ;

}**while**(na<=0||ma<=0||na>=lmax||ma>=lmax||k!=2||r1 != ma||r!=na);

printf("Введите матрицу %d на %d\n",na,ma);

**for**( i=1;i<=na;i++)

**for**( j=1;j<=ma;j++)

**do** {

q = scanf("%d", &a[i][j]);

**if** (q != 1) {

printf("Введите заново\n");

**while** (getchar() != '\n');

}

} **while** (q != 1);

actual\_row = 0;

max = a[1][1];

**for** (i = 1; i <= na; i++)

**for** (j = 2; j <= ma;j++)

**if** (a[i][j] > max)

max = a[i][j];

printf("%d\n", max);

actual\_row = 0;

**for** (j = 1; j <= ma; j++) {

**int** f = 0;

i = 1;

**while** (i <= na && !f) {

**if** (a[i][j] == max) {

f = 1;

}

i++;

}

**if** (!f) {

actual\_row++;

**for** (i = 1; i <= na; i++) {

a[i][actual\_row] = a[i][j];

}

}

}

**if** (actual\_row == na) {

printf("Нет изменений\n");

} **else** **if** (actual\_row == 0) {

printf("Полное удаление");

} **else** {

printf("Новая матрица\n");

**for** (i = 1; i <= na; i++) {

**for** (j = 1; j <= actual\_row; j++) {

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

}

printf("\n");

}

}

**return** 0;

}

Тестовые данные

№1

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

}

**№2**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**№3**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**