Министерство образования и науки РФ

Тамбовский государственный технический университет

Кафедра

Отчет по лабораторной работе №1

по дисциплине «Программирование»

Выполнил: студент группы

.

Проверил:.

Тамбов 20

1. Постановка задания:

Сортировать матрицу размера [n,m] по убыванию методом обмена. Расставить элементы в указанном порядке:

1. Краткое описание алгоритма:
2. Создание двумерного динамического массива a размера [n,m] (n и m задаются с клавиатуры)
3. Заполнение массива случайными числами от 1 до 100
4. Преобразование двумерного массива a[n,m] в одномерный массив b[n\*m] и вывод его на экран
5. Сортировка массива b по убыванию методом обмена (сравнивание соседних элементов массива, при выполнении условия b[i]<b[i-1] менять элементы местами с помощью пустой переменной zam), вывод отсортированного одномерного массива b
6. Преобразование одномерного массива b в двумерный массив a, расставляя элементы в указанном порядке
7. Вывод двумерного массива a размера [n,m] на экран
8. Блок схема алгоритма:

l=m\*n;

Ввод n,m

Ввод an×m

1 i=1, k=1

2 j=1

2 j=2

1 i=1

1 i++

пока i<=n

2 j++, k++

пока j<=m

b[k]=a[i][j]

нет

b[j]<b[j-1]

2 j++

пока j<=l-i

zam=b[j],

b[j]=b[j-1],

b[j-1]=zam

да

i+=1, j-=1

нет

да

m>=x

x++

i=n или j=1

1 пока

i<>n и j<>m

a[i][j]=b[k]

i=1, j=1,

k=l, x=1,

z=2

1 i++

пока i<=l

нет

1

k--,

a[i][j]=b[k]

j=m, i=z,

z++

да

j=x, i=1

Вывод an×m

2 j=1

1 i=1

1. Исходный текст программы:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()

{

srand((unsigned)time(NULL));

int n,m,i,j,zam,k,l,z,x;

unsigned int \*\*a,\*b;

printf("input size n = ");

scanf("%d",&n);

printf("input size m = ");

scanf("%d",&m);

l=m\*n;

// ввод массива а

a=(unsigned int\*\*)malloc(n\*sizeof(int\*));

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

a[i]=(unsigned int\*)malloc(m\*sizeof(int));

b=(unsigned int\*)malloc(l\*sizeof(int));

// заполнение случайными числами от 1 до 100

for(j=0;j<m;j++)

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

a[i][j] = rand()%100 + 1;

// заполнение и вывод начального вектора

printf("\n");

printf("first vector:");

printf("\n");

printf("\n");

for(i=0,k=0;i<n;i++)

for(j=0;j<m;j++,k++)

{

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

printf("%d ",b[k]);

}

printf("\n");

printf("\n");

// вывод начальной матрицы

/\*for(i=0;i<n;i++)

{

for(j=0;j<m;j++)

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

printf("\n");

}

printf("\n");

\*/

// сортировка методом пузырька вектора b;

for(i=0;i<l;i++)

for(j=1;j<l-i+1;j++)

{

if (b[j]<b[j-1])

{

zam=b[j];

b[j]=b[j-1];

b[j-1]=zam;

}

}

// вывод отсортированного вектора b

printf("sorted vector:");

printf("\n");

printf("\n");

for(i=0;i<l;i++)

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

printf("\n");

// заполнение массива а

i=0;j=0;k=(l-1);x=0;z=1;

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

while(!((i==n-1)&&(j==m-1)))

{

if ((i==n-1)||(j==0))

{

x++;

if(m>x)

{ j=x; i=0; }

else

{ j=m-1; i=z;z++;}

}

else

{ i+=1; j-=1; }

k--;

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

}

// вывод конечной матрицы

printf("\n");

printf("sorted array:");

printf("\n");

printf("\n");

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

{

for(j=0;j<m;j++)

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

printf("\n");

}

return 0;

}