**Отчет ПО ЛАБОРАТОРНЫМ РАБОТАМ**

**По информатике**

Лабораторная работа №6.

Обработка двумерных массивов

(наименование лабораторной работы в соответствии с учебным планом)

***Цель и задачи работы:***

Изучение алгоритмов формирования и обработки двумерных массивов, программирование и отладка программ формирования и обработки массивов.

***Задание №1***

1. Дана матрица размера *M×N* и целое число *K* (1 ≤ *K* ≤ *M*). Вывести элементы *K*-й строки данной матрицы.

2. Дана матрица размера *M×N* (*M* и *N*— четные числа). Поменять местами левую верхнюю и правую нижнюю четверти матрицы.

3. Даны числа n и m. После чего задаётся матрица размером n×m из 0 и 1. Будем считать, что эта матрица задаёт карту лабиринта, при этом 0 является пустым пространством, а 1 стеной. В данном лабиринте будем считать, что точкой входа является клетка с координатами (1,1), а точкой выхода (n,m). Необходимо определить, возможно ли из точки входа добраться до точки выхода, если можно двигаться только в четырёх возможных направлениях (север, юг, запад, восток) и перемещаться можно только в свободную клетку.

***Листинг программы:***

1) C++

1)

#include <iostream>

#include <vector>

using namespace std;

int main() {

cout << "Enter the number of lines: ";

int M;

cin >> M;

cout << "\nEnter the number of columns: ";

int N;

cin >> N;

vector <vector<int>> a(M, vector<int>(N));

for (int i = 0; i < M; i++) {

for (int j = 0; j < N; j++) {

a[i][j] = i \* 10 + j;

}

}

for (vector<int> item : a) {

for (int element : item) {

if (element < 10) cout << " 0" << element;

else cout << " " << element;

}

cout << endl;

}

cout << "\n\nEnter the line you would like to display (1 <= K <= " << M << " ) : ";

int K;

cin >> K;

cout << "\n\nAnswer:";

for (int j = 0; j < N; j++) {

if (a[K-1][j] < 10) cout << " 0" << a[K - 1][j];

else cout << " " << a[K - 1][j];

}

}

2)

#include <iostream>

using namespace std;

int main() {

cout << "Enter the number of lines: ";

int M;

cin >> M;

cout << "\nEnter the number of columns: ";

int N;

cin >> N;

int\*\* ary;

ary = new int\* [M];

cout << endl;

for (int i = 0; i < M; i++) {

ary[i] = new int[N];

}

for (int i = 0; i < M; i++) {

for (int j = 0; j < N; j++) {

ary[i][j] = (i) \* 10 + (1 + j);

if (i == 0) cout << " 0" << ary[i][j];

else cout << " " << ary[i][j];

}

cout << endl;

}

int z = N / 2;

int l = M / 2;

int x = z;

int y = l;

cout << endl;

int k;

for (int i = 0; i < z; i++) {

y = l;

if ((x - 3) != i) x += 1;

for (int j = 0; j < l; j++) {

k = ary[i][j];

ary[i][j] = ary[x][y];

ary[x][y] = k;

y++;

}

}

for (int i = 0; i < M; i++) {

for (int j = 0; j < N; j++) {

if (ary[i][j] < 10) cout << " 0" << ary[i][j];

else cout << " " << ary[i][j];

}

cout << endl;

}

}

3)

#include <iostream>

using namespace std;

bool shag(short y, short x, bool\*\* arr, short& m, short& n)

{

if (y == m - 1 && x == n - 1) return 1;

arr[y][x] = 1;

bool result1 = 0, result2 = 0, result3 = 0, result4 = 0;

if (y - 1 >= 0)

if (arr[y - 1][x] == 0)

result1 = shag(y - 1, x, arr, m, n);

if (y + 1 < m)

if (arr[y + 1][x] == 0)

result2 = shag(y + 1, x, arr, m, n);

if (x - 1 >= 0)

if (arr[y][x - 1] == 0)

result3 = shag(y, x - 1, arr, m, n);

if (x + 1 < n)

if (arr[y][x + 1] == 0)

result4 = shag(y, x + 1, arr, m, n);

return (result1 || result2 || result3 || result4);

}

int main()

{

short m, n;

cout << "Enter the number of lines: ";

cin >> m;

cout << "\nEnter the number of columns: ";

cin >> n;

srand(time(NULL));

bool\*\* arr = new bool\* [m];

for (short i = 0; i < m; i++)

arr[i] = new bool[n];

for (short i = 0; i < m; i++)

for (short j = 0; j < n; j++)

arr[i][j] = rand() % 2;

arr[0][0] = 0;

arr[m - 1][n - 1] = 0;

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

cout << " " << arr[i][j];

}

cout << endl;

}

cout << (shag(0, 0, arr, m, n) ? "YES" : "NO");

return 0;

}

2) Python

1) A = [[1,2,3], [4,5,6], [7,8,9]]

for i in range(len(A)):

for j in range(len(A[i])):

print(A[i][j], end=' ')

print()

print("Enter the line you would like to display: ")

k = int(input())

print("Answer: ")

for i in range(len(A[k-1])):

print(A[k-1][i], end=" ")

2) print("Enter the number of lines:")

M = int(input())

print("Enter the number of columns:")

N= int(input())

A = [0]\*M

for i in range(len(A)):

A[i] = [0]\*N

for i in range(len(A)):

for j in range(len(A[i])):

A[i][j] = i\*10 + (1 + j)

print()

for i in range(len(A)):

for j in range(len(A[i])):

print(A[i][j], end = " ")

print()

z = int(N/2)

l = int(M/2)

x = z

y = l

print()

print()

print()

for i in range(z):

y = l

if ((x-3) != i):

x +=1

for j in range(l):

k = A[i][j]

A[i][j] = A[x][y]

A[x][y]=k

y += 1

for i in range(len(A)):

for j in range(len(A[i])):

print(A[i][j], end = " ")

print()

3) import random

def shag(y,x,A, m, n):

if ((y == m-1) and (x == n-1)):

return 1

A[y][x] = 1

result1 = 0

result2 = 0

result3 = 0

result4 = 0

if(y-1 >=0):

if(A[y-1][x] == 0):

result1=shag(y-1,x,A,m,n)

if (y + 1 < m):

if (A[y + 1][x] == 0):

result2 = shag(y + 1, x, A, m, n);

if (x - 1 >= 0):

if (A[y][x - 1] == 0):

result3 = shag(y, x - 1, A, m, n);

if (x + 1 < n):

if (A[y][x + 1] == 0):

result4 = shag(y, x + 1, A, m, n);

return (result1 or result2 or result3 or result4);

print("Enter the number of lines:")

M = int(input())

print("Enter the number of columns:")

N = int(input())

A= [0] \* M

for i in range(len(A)):

A[i] = [0] \* N

for i in range(len(A)):

for j in range(len(A[i])):

A[i][j] = random.randint(0,1)

A[0][0] = 0

A[M-1][N-1] = 0

for i in range(len(A)):

for j in range(len(A[i])):

print(A[i][j], end= " ")

print()

print()

print()

z = shag(0,0,A,M,N)

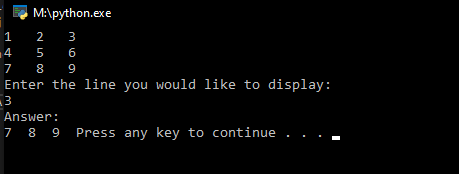
if (z == 1):

print("YES")

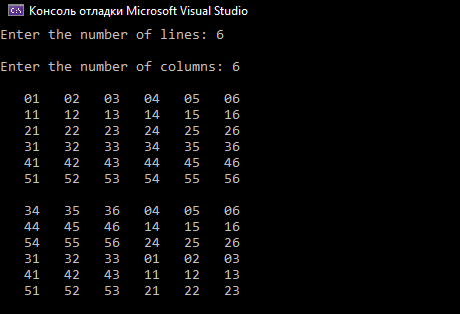
if (z == 0):

print("NO")

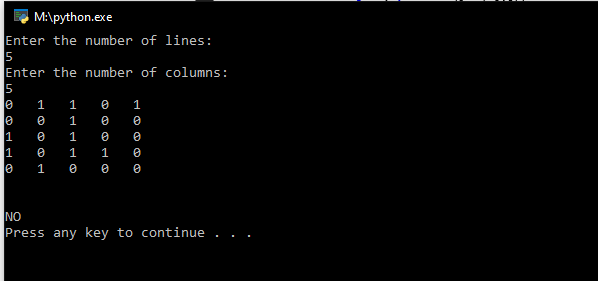
***Пример работы программы:***



*(рис. 1 – пример работы 1 программы на Python)*



*(рис. 2 – пример работы 2 программы на C++)*



*(рис. 3 – пример работы 3 программы на Python)*

**Задание №2.**

Многоалфавитное шифрование методом сложной замены (подстановки). Дешифрование текста. Написать программу решения задачи в соответствии с индивидуальным вариантом. (cистему шифрования Вижинера для преобразования исходного текста на английском языке)

***Листинг программы:***

1) C++

#include <iostream>

#include <vector>

using namespace std;

void cod(vector <int> message, vector <int> key, int code[], const int n) {

int a;

int b;

for (int i = 0; i < message.size(); i++) {

a = ((message[i] + key[i]) % n);

b = ((message[i] + key[i]) % n) % 26 + 65;

code[i] = a;

cout << char(b);

}

}

void decod(vector <int> message, int code[], char decode[], const int n, vector <int> key) {

int a;

for (int i = 0; i < message.size(); i++) {

a = ((code[i] + n - key[i]) % n);

decode[i] = a;

}

}

int main() {

const int n = 255;

char a\_input[255];

char key\_input[255];

int a\_coder[255];

char a\_decoder[255];

vector<int> message;

vector<int> key;

int64\_t i, num;

cout << "Enter the message: ";

cin >> a\_input;

for (int i = 0; a\_input[i]; i++) {

message.push\_back(static\_cast<int>(a\_input[i]));

}

cout << "\nEnter the key: ";

cin >> key\_input;

for (int i = 0; key\_input[i]; i++) {

key.push\_back(static\_cast<int>(key\_input[i]));

}

if (key.size() < message.size()) {

for (i = 0; key.size() < message.size(); i++) {

num = key[i];

key.push\_back(num);

}

}

else

if (key.size() > message.size()) {

while (key.size() > message.size()) {

key.pop\_back();

}

}

cout << "\nResulte of crypt: ";

cod(message, key, a\_coder, n);

cout << endl;

cout << endl << "\nResult of decrypt: ";

decod(message, a\_coder, a\_decoder, n, key);

for (i = 0; i < message.size(); i++) {

cout << a\_decoder[i];

}

cout << "\n\n\n";

}

2) Python

N = 255

array\_input = []

array\_key = []

print("Enter message: ",end='')

string = input()

for i in range (0,len(string)):

a = ord(string[i])

array\_input.append(a)

print("Enter key: ",end='')

input\_key = input()

for i in range (0,len(input\_key)):

a = ord(input\_key[i])

array\_key.append(a)

if (len(array\_key)<len(array\_input)):

i = 0

while (len(array\_key)<len(array\_input)):

array\_key.append(array\_key[i])

i+=1

if (len(array\_key)>len(array\_input)):

j = 0

while (len(array\_key)>len(array\_input)):

array\_key.pop()

code=[]

for i in range(0,len(array\_input)):

code.append((array\_input[i]+array\_key[i])%N)

print("Crypted: ",end='')

for i in range (0,len(code)):

print(code[i],end='')

print()

decode=[]

for i in range(0,len(array\_input)):

decode.append(chr((code[i]+ N -array\_key[i])%N))

print("Decrypted: ",end='')

for i in range (0,len(decode)):

print(decode[i],end='')

print()

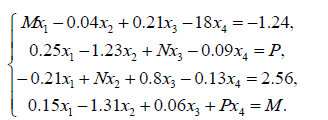
***Пример работы программы:***



*(рис. 4 – пример работы программы на C++)*

**Задание №3:**

Из табл.6.3.1 выбрать данные для системы линейных уравнений. Найти решение этой системы прямым и приближенным методами с точностью до ε=10-3 . Варианты нечетные решить систему уравнений методом Гаусса с выбором главного элемента или методом простых итераций, четные варианты – методом Холесского или методом простых итераций.



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **M** | **N** | **P** | **№** | **M** | **N** | **P** |
| **1** | -0.77 | 0.16 | 1.12 | **9** | -1.13 | 0.14 | 0.87 |
| **2** | 0.93 | 0.07 | -0.84 | **10** | 0.91 | -0.23 | -1.04 |
| **3** | -1.14 | -0.17 | 0.95 | **11** | -0.88 | 0.1 | 0.91 |
| **4** | 1.08 | 0.22 | -1.16 | **12** | 1.25 | -1.14 | -1.09 |
| **5** | 0.87 | -0.19 | 1.08 | **13** | 0.79 | 0.18 | -0.86 |
| **6** | -1.21 | 0.2 | 0.88 | **14** | -1.19 | -0.21 | 1.21 |
| **7** | 1.09 | -0.16 | 0.84 | **15** | 0.89 | 0.12 | -1.15 |
| **8** | 0.89 | 0.08 | -1.21 | **16** | 1.08 | 0.22 | -1.16 |

***Листинг программы:***

1)C++

#include <iostream>

#include <iomanip>

using namespace std;

int main() {

cout << "The program considers the solution of a system of linear equations: \n" << endl;

cout << " Mx(1) - 0,04x(2) + 0,21x(3) - 18x(4) = -1,24 \n 0,25x(1) - 1,23x(2) + Nx(3) - 0,09x(4) = P \n -0,21x(1) + Nx(2) + 0,8x(3) - 0,13x(4) = 2,56 \n 0,15x(1) -1,31x(2) + 0,06x(3) + Px(4) = M \n";

cout << "\nPlease enter M: ";

double M;

cin >> M;

cout << "\nPlease enter N: ";

double N;

cin >> N;

cout << "\nPlease enter P: ";

double P;

cin >> P;

double\*\* a;

a = new double\* [4];

cout << endl;

for (int i = 0; i < 4; i++) {

a[i] = new double[5];

}

a[0][0] = M;

a[0][1] = -0.04;

a[0][2] = 0.21;

a[0][3] = -18;

a[0][4] = -1.24;

a[1][0] = 0.25;

a[1][1] = -1.23;

a[1][2] = N;

a[1][3] = 0.09;

a[1][4] = P;

a[2][0] = -0.21;

a[2][1] = N;

a[2][2] = 0.8;

a[2][3] = -0.13;

a[2][4] = 2.56;

a[3][0] = 0.15;

a[3][1] = -1.31;

a[3][2] = 0.06;

a[3][3] = P;

a[3][4] = M;

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 5; j++) {

cout << left << setw(10) << a[i][j];

}

cout << endl;

}

double first\_element;

for (int i = 0; i < 4; i++) {

first\_element = a[i][0];

for (int j = 0; j < 5; j++) {

a[i][j] = round(a[i][j] / first\_element \* 1000) / 1000 ;

if (i > 0) a[i][j] = a[i][j] - a[0][j];

}

}

cout << "\n\nEach line was divided into 1 of their element and subtract 1 line from each line (starting from 2): \n" << endl ;

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 5; j++) {

cout << left << setw(10) << a[i][j];

}

cout << endl;

}

for (int i = 1; i < 4; i++) {

first\_element = a[i][1];

for (int j = 1; j < 5; j++) {

a[i][j] = round(a[i][j] / first\_element \* 1000) / 1000;

if (i > 1) a[i][j] = a[i][j] - a[1][j];

}

}

cout << "\n\nEach line was divided into 2 of their element and subtract 2 line from each line (starting from 3): \n" << endl ;

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 5; j++) {

cout << left << setw(10) << a[i][j];

}

cout << endl;

}

for (int i = 2; i < 4; i++) {

first\_element = a[i][2];

for (int j = 2; j < 5; j++) {

a[i][j] = round(a[i][j] / first\_element \* 1000) / 1000;

if (i > 2) a[i][j] = a[i][j] - a[2][j];

}

}

cout << "\n\nEach line was divided into 3 of their element and subtract 3 line from each line (starting from 4): \n" << endl ;

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 5; j++) {

cout << left << setw(10) << a[i][j];

}

cout << endl;

}

double x[4] = { 0,0,0,round((a[3][4] / a[3][3]) \* 1000) / 1000 };

int l = 3;

for (int i = 2; i >= 0; i--) {

first\_element = a[i][4];

for (int j = 3; j > 0; j--) {

if ((j == 0) && (a[i][j] == 1)) break;

if ((a[i][j] == 1) && (a[i][j - 1] == 0)) break;

first\_element -= a[i][j] \* x[j];

}

x[l-1] = round(first\_element \* 1000) / 1000;

l--;

}

cout << "\n\nAnswer: ";

for (int i = 0; i <= 3; i++) {

cout << "\nx[" << i + 1 << "] = " << x[i];

}

}

2) Python

N = 4

exp = 3

a = [[0.87, -0.04, 0.21, -18, -1.24],

[ 0.25, -1.23, -0.19, -0.09, 1.08],

[-0.21, -0.19, 0.8, -0.13, 2.56],

[ 0.15, -1.31, 0.06, 1.08, 0.87]]

x = [0,0,0,0,0]

for k in range (0,N):

aa = abs(a[k][k])

i = k

for m in range (k+1,N):

if (abs(a[m][k]) > aa):

i = m

aa = abs(a[m][k])

if (aa==0):

break

if (i!=k):

for j in range (k,N+1):

bb = a[k][j]

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

a[i][j] = bb

aa = a[k][k]

a[k][k]=1

for j in range (k+1,N+1):

a[k][j] = a[k][j] / aa;

for i in range (k+1,N):

bb = a[i][k]

a[i][k] = 0

if (bb!=0):

for j in range (k+1,N+1):

a[i][j] = a[i][j] - bb\*a[k][j]

i = N-1

j = N

while (i >= 0 ):

x[i] = 0

aa = a[i][N]

j=N

while (j > i):

aa = aa - a[i][j]\*x[j]

j-=1

x[i] = aa

i-=1

print( "Result:" )

for i in range (0,N):

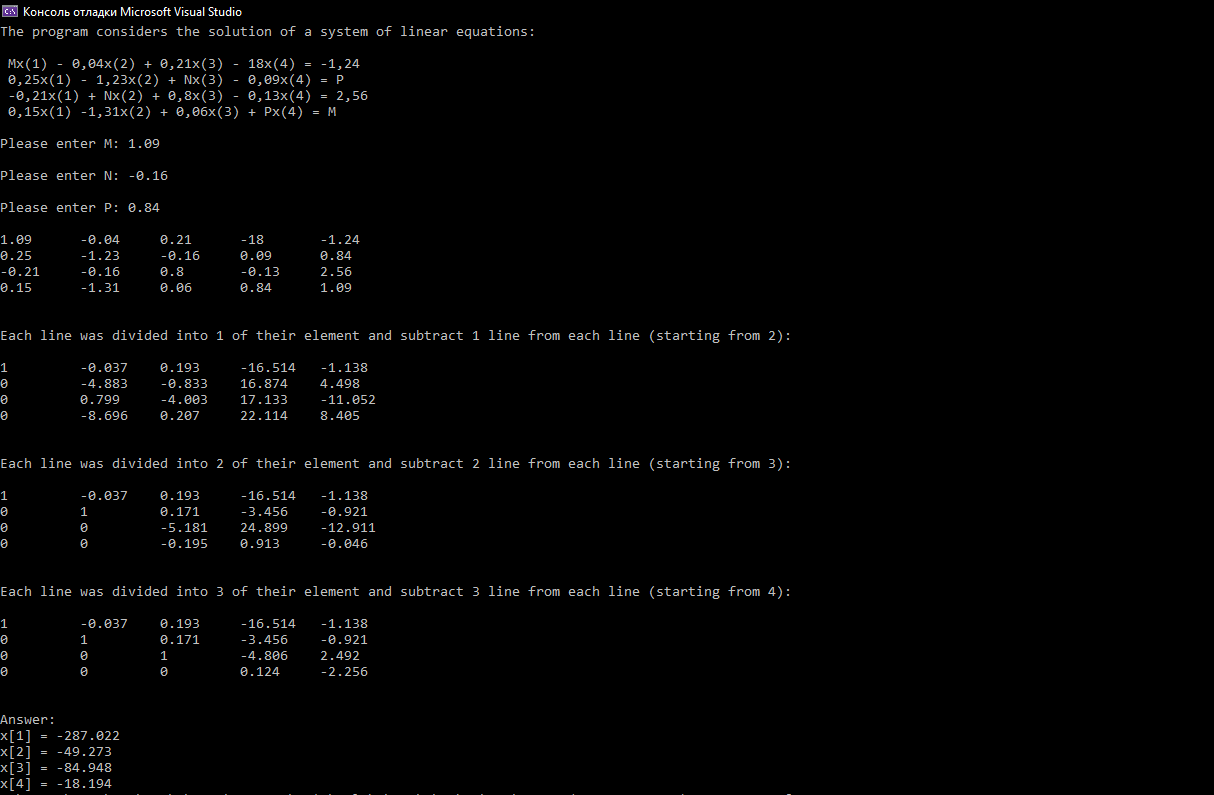
print("x(",end='')

print(i+1,end='')

print(") = ",end='')

print( round(x[i],3))

***Пример работы программы:***



(*рис. 5 – пример работы программы на C++*)