

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота №5
з дисципліни
«Алгоритмізація та програмування»

Виконав:
студент групи К
Н-111
Петров Дмитро
Викладач:
Гасько Р.Т.

Львів – 2018 р.

Зміст звіту

1. Постановка завдання.
2. Варіант завдання
3. Текст програми.
4. Результат розв'язку конкретного варіанту.

2. Постановка завдання

Використовуючи функції, розв'язати зазначене у варіанті завдання.

Масив повинен передаватися у функцію як параметр.

3. Варіанти

18. Задано двовимірний масив $N \times N$. Послідовно розглядаються квадратні підмасиви, правий верхній елемент яких лежить на бічній діагоналі. У кожному такому підмасиві перебуває максимальний елемент. Шляхом перестановок рядків і стовпців (повністю) елемент треба перемістити в правий верхній кут підмасиву. Перевірити чи вийшла на бічній діагоналі спадаюча послідовність елементів.

```

#include <stdio.h>

void print(int n , int a[n][n]){

    int i, y;

    for (i= 0; i < n ; i++)
    {
        for (y = 0; y < n ; y++)
        {
            printf("%d ", a[i][y]);

        }

        printf("\n");
    }
}

void rep(int n, int a[n][n], int max_i, int max_y, int c){

//replacing

int i, y;
int b[n][n];

//replacing a string
    for(y = 0 ; y < n ; y++)
    {
        b[c][y] = a[max_i][y];
        a[max_i][y] = a[c][y];
        a[c][y] = b[c][y];
    }
//END

// replacing a row
    for (i= 0 ; i < n ; i++)
    {
        b[i][n - c - 1] = a[i][max_y];
        a[i][max_y] = a[i][n - c - 1];
        a[i][n - c - 1] = b[i][n - c - 1];
    }

    print(n, a);
    printf("\n");
//END

}

//MAIN

```

```

int main (){

int i, y, n;

scanf("%d", &n);

int a[n][n];

for (i= 0; i < n ; i++)
{
    for (y = 0; y < n ; y++)
    {
        a[i][y]= i*10 + y;
/*      printf("a[%d][%d]= ", i, y);
        scanf("%d", a[i][y]);
*/
        //printf("%d ", a[i][y]);
    }
//printf("\n");
}

//Checking small arrays for max
int max_i, max_y, c;

for (c = 0; c < n - 1 ; c++)
{

    max_i= -1;
    max_y= -1;

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

        for (y = 0 ; y < n - c ; y++)
        {

            //printf("\n a[%d][%d]= %d \n diagonal[%d][%d]=%d \n", i, y, a[i][y], 0+c ,
n-c-1, a[0+c][n-c -1]);
            if (a[i][y] > a[0+c][n-c -1])
            {
                //printf("Yes");
                max_i = i;
                max_y = y;
            }else if (max_i == -1 && max_y == -1)
            {
                //printf("No");
                max_i = 0 + c;
                max_y = n - c - 1;
            }
            //printf("%d ", a[i][y]);

```

```

    }
    //printf("\n");
    }
    printf("\n [%d][%d] \n", max_i, max_y);
    rep(n, a, max_i, max_y, c);

}
//END

```

```

print(n , a);

```

```

//check regression

```

```

for(i = 0, c = 0 ; i < n - 1 ; i++)
{
    if (a[i][n - i - 1] > a[i + 1][n - i - 2])
    {
        c++;
    }
}
if (c == n - 1)
{
    printf("\n The matrix has regression on the side diagonal \n");
}
//END

```

```

}

```

5

[4][4]

```
40 41 42 43 44
10 11 12 13 14
20 21 22 23 24
30 31 32 33 34
0 1 2 3 4
```

[3][3]

```
40 41 42 43 44
30 31 32 33 34
20 21 22 23 24
10 11 12 13 14
0 1 2 3 4
```

[2][2]

```
40 41 42 43 44
30 31 32 33 34
20 21 22 23 24
10 11 12 13 14
0 1 2 3 4
```

[3][1]

```
40 41 42 43 44
30 31 32 33 34
20 21 22 23 24
10 11 12 13 14
0 1 2 3 4
```

```
40 41 42 43 44
30 31 32 33 34
20 21 22 23 24
10 11 12 13 14
0 1 2 3 4
```

The matrix has regression on the side diagonal

Досягнення студента 'dmytroptv' (dmytroptv@gmail.com) на курсі

