

Лабораторна робота №5

Тема: «Функції і масиви»

Виконав:

**Дерев'янний Андрій,
КН-108(11 Варіант)**

1. Постановка завдання:

11. Написати функцію, яка перевіряє чи по зростанню або спаданню впорядкований зазначений рядок двовимірної масиви. Впорядкувати по зростанню всі рядки двовимірної масиви, які не впорядковані по спаданню.

2. Програма Розв'язку:

```
#include <stdio.h>
```

```
#include <cs50.h>
```

```
int check(int arr1[3][3], int k);
```

```
int main()
```

```
{
```

```
    int arr[3][3]=
```

```
    {
```

```
        {2, 5, 9},
```

```
        {7, 4, 3},
```

```
        {1, 8, 6}
```

```
    };
```

```
    int n, l, j, o, t;
```

```
    printf("Введіть Номер Рядка: "); //запит рядка для перевірки
```

```
    scanf("%d", &n);
```

```
    switch(check(arr, n-1)) //виведення типу перевіреного рядка
```

```
    {
```

```
        case 1: printf("Елементи Рядка Зростають\n");
```

```
            break;
```

```
        case 2: printf("Елементи Рядка Спадають\n");
```

```
            break;
```

```
default: printf("Елементи Розташовані Хаотично\n");  
}
```

```
for(l=0; l<3; l++) //умова для сортування масиву  
    switch(check(arr, l)) //виведення відсортованого масиву  
    {  
        case 1:  
            for(j=0; j<3; j++)  
            {  
                printf("| %d ", arr[l][j]);  
            }  
  
            printf("\n");  
            break;  
  
        case 2:  
            for(j=0; j<3; j++)  
            {  
                printf("| %d ", arr[l][j]);  
            }  
  
            printf("\n");  
            break;  
  
        default:  
            o = 0;  
            while ( o < 3-1 )  
            {  
                if( arr[l][o+1] >= arr[l][o] )  
                    o++;  
            }  
        }  
    }
```

```

        else
        {
            t = arr[l][o];
            arr[l][o] = arr[l][o+1];
            arr[l][o+1] = t;
            o = 0;
        }

    }

    for(j=0; j<3; j++)
    {
        printf("| %d ", arr[l][j]);
    }
    printf("\n");
}
return 0;
}

```

int check(int arr1[3][3], int k) //функція перевіряє послідовність елементів вказаного рядка

```

{
    int h=0, i;

    for(i=0; i<2; i++)
        if(arr1[k][i]<arr1[k][i+1])
            h=1;
    else
    {
        h=2;
    }
}

```

```
        break;
    }
    if(h==1)
        return h;
    else
        for (i=0; i<2; i++)
            if(arr1[k][i]>arr1[k][i+1])
                h=2;
            else
            {
                h=3;
                break;
            }
    return h;
}
```

3. Результати Програми:

```
~/workspace/Alg-Programming/Lab5/ (master) $ ./2DArr
Введіть Номер Рядка: 1
Елементи Рядка Зростають
| 2 | 5 | 9 |
| 7 | 4 | 3 |
| 1 | 6 | 8 |
~/workspace/Alg-Programming/Lab5/ (master) $ ./2DArr
Введіть Номер Рядка: 2
Елементи Рядка Спадають
| 2 | 5 | 9 |
| 7 | 4 | 3 |
| 1 | 6 | 8 |
~/workspace/Alg-Programming/Lab5/ (master) $ ./2DArr
Введіть Номер Рядка: 3
Елементи Розташовані Хаотично
| 2 | 5 | 9 |
| 7 | 4 | 3 |
| 1 | 6 | 8 |
```

4. Оновлена Програма Розв'язку:

```
#include <stdio.h>
```

```
#include <cs50.h>
```

```
int check(int arr1[5][5], int k);
```

```
int main()
```

```
{
```

```
int arr[5][5], m, n;
```

```
for(m=0; m<5; m++)
```

```
{
```

```
    for(n=0; n<5; n++)
```

```
    {
```

```
        printf("Введіть %d-ий Елемент %d-рядка: ", n+1, m+1);
```

```
        scanf("%d", &arr[m][n]);
```

```
    }
```

```
    printf("\n");
```

```
}
```

```
int l, j, o, t, a;
```

```
printf("Введіть Номер Рядка: "); //запит рядка для перевірки
```

```
scanf("%d", &a);
```

```
switch(check(arr, a-1)) //виведення типу перевіреного рядка
```

```
{
```

```
    case 1: printf("Елементи Рядка Зростають\n");
```

```
        break;
```

```
    case 2: printf("Елементи Рядка Спадають\n");
```

```

        break;

    default: printf("Елементи Розташовані Хаотично\n");
}

printf("\n");

printf("Відсортований масив виглядатиме так:");

for(l=0; l<5; l++) //умова для сортування масиву
    switch(check(arr, l)) //виведення відсортованого масиву
    {
        case 1:
            for(j=0; j<5; j++)
            {
                printf("| %d ", arr[l][j]);
            }

            printf("\n");
            break;

        case 2:
            for(j=0; j<5; j++)
            {
                printf("| %d ", arr[l][j]);
            }

            printf("\n");
            break;

        default:
            o = 0;
            while ( o < 5-1 )
            {

```



```

        if( arr[l][o+1] >= arr[l][o] )
            o++;
        else
        {
            t = arr[l][o];
            arr[l][o] = arr[l][o+1];
            arr[l][o+1] = t;
            o = 0;
        }

    }

    for(j=0; j<5; j++)
    {
        printf("| %d ", arr[l][j]);
    }

    printf("\n");

}

return 0;
}

int check(int arr1[5][5], int k) //функція перевіряє послідовність елементів
вказаного рядка
{
    int h=0, i;

```

```
for(i=0; i<4; i++)
    if(arr1[k][i]<arr1[k][i+1])
        h=1;
    else
        {
            h=2;
            break;
        }
if(h==1)
    return h;
else
    for (i=0; i<4; i++)
        if(arr1[k][i]>arr1[k][i+1])
            h=2;
        else
            {
                h=3;
                break;
            }
return h;
}
```