МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №10

з дисципліни

«Алгоритмізація та програмування І»

Виконав:

студент групи КН-108

Зінько Павло

Викладач:

Гасько Р.Т.

Львів – 2018р

Варіант №13

Тема: "Динамічні масиви"

Мета: Організація динамічних масивів.

Постановка завдання

Написати програму, у якій створюються динамічні масиви й виконати їхню обробку у відповідності до свого варіанту

Виконати завдання:

13

Сформувати двовимірний масив. Знищити з нього всі стовпці, у яких зустрічається задане число.

Код:

```
#include <stdio.h>
#include <malloc.h>
#include <time.h>
#include <stdlib.h>
#include <stdbool.h>

//мітка при якій коли функція бачить її буде переривати цикл
#define POINT 999

int count;

//функція передання вказівника
```

```
int** initialize(int size1,int size2)
{
   // вказівник на вказівник на рядок елементів
        int **arr;

        // Виділення пам'яті для вказівника на рядки
        arr = malloc(size2 * sizeof(int*));

        // цикл по рядкам
        for(int i = 0; i < size2;i++)
        {
           // Виділення пам'яті для зберегання рядків
                arr[i] = malloc(size1*sizeof(int));
        }
        return arr;
}

// функція заповнення нашого масиву
void create(int** arr,int size1, int size2)
{
   srand(time(NULL));
        for(int i = 0; i < size1;i++)
        {
                for(int j = 0;j < size2;j++)
                {
                        arr[j][i] = rand()%20;
```

```
                }
        }
}

// Вивід згенерованого масива
void print(int** arr,int size1,int size2)
{
        for(int i = 0;i < size1;i++)
        {
                for(int j = 0;j< size2;j++)
                {
                                printf("%d\t",arr[j][i]);
                }
                printf("\n");
        }
        printf("\n");
}

//  фукнція зміни та перенесення нового стовпчика
void point(int** arr,int size1,int size2)
{
   //елемент який потрібно видалити
        int Element;
        printf("Give me the element for deliting:\n");
        scanf("%d",&Element);
        printf("\n");
```

```
	for(int i = 0; i < size1;i++)
	{
		for(int j = 0; j < size2; j++)
		{
			if(arr[j][i] == Element)
			{
				for(int k = 0; k < size2;k++)
				{
				   // якщо найшло цю мітку записує кількість стовпців які потрібно замінити
					arr[j][k] = POINT;
					if(k == size2-1)
						count++;
				}
			}
		}
	}
}

// фукнція видалення помічених функцією "point" стовпців
int** kill(int** arr,int** arr2,int size1, int size2)
{
	int tmpi = 0;
	for(int i = 0; i < size2;i++)
	{
		for(int j = 0; j < size1;j++)
```

```
            {
               // якщо найшло цю мітку,перериває цикл
                    if(arr[i][j] == POINT)
                    {
                            break;
                    }

                    arr2[tmpi][j] = arr[i][j];

                    //записує кількість стовпців які потрібно видалити
                    if(j==size1-1)
                            tmpi++;
            }
        }

        // вертає новий стовпчик
        return arr2;
}

// Звільняє пам'ять та видаляє масив
void freeArr(int** arr,int size1)
{
        for(int i = 0; i < size1;i++)
        {
                free(arr[i]);
        }
```

```
}

int main()
{
	int size1;
	int size2;
	printf("Print size of array\n");
	scanf("%d",&size2);
	scanf("%d",&size1);
	printf("\n");

	// створює наш масив
	int** arr = initialize(size1,size2);

	// ініціалізує наш масив
	create(arr,size1,size2);

	//виводить згенерирований масив
	print(arr,size1,size2);
	point(arr,size1,size2);


	// ініціалізує наш масив але без данних стовпців
	int** arr1 = initialize(size1,size2-count);
	arr1 = kill(arr,arr1,size1,size2);
```

```
        // виводить наш масив але без данних стовпців
        print(arr1,size1,size2-count);

        // звільняє память
        freeArr(arr,size1);
        freeArr(arr1, size1-1);

}
```

```c
#include <stdio.h>
#include <malloc.h>
#include <time.h>
#include <stdlib.h>
#include <stdbool.h>

//мітка при якій коли функція бачить її буде переривати цикл
#define POINT 999

int count;

//функція передання вказівника
int** initialize(int size1,int size2)
{
    // вказівник на вказівник на рядок елементів
    int **arr;

    // Виділення пам'яті для вказівника на рядки
    arr = malloc(size2 * sizeof(int*));

    // цикл по рядкам
    for(int i = 0; i < size2;i++)
    {
        // Виділення пам'яті для зберегання рядків
        arr[i] = malloc(size1*sizeof(int));
    }
    return arr;
}

// функція заповнення нашого масиву
void create(int** arr,int size1, int size2)
{
    srand(time(NULL));
    for(int i = 0; i < size1;i++)
    {
        for(int j = 0;j < size2;j++)
        {
            arr[j][i] = rand()%20;
        }
    }
}

// Вивід згенерованого масива
void print(int** arr,int size1,int size2)
{
    for(int i = 0;i < size1;i++)
    {
        for(int j = 0;j< size2;j++)
        {
                printf("%d\t",arr[j][i]);
        }
        printf("\n");
    }
    printf("\n");
}

//  фукнція зміни та перенесення нового стовпчика
void point(int** arr,int size1,int size2)
{
    //елемент який потрібно видалити
    int Element;
    printf("Give me the element for deliting:\n");
    scanf("%d",&Element);
    printf("\n");

    for(int i = 0; i < size1;i++)
    {
        for(int j = 0; j < size2; j++)
        {
            if(arr[j][i] == Element)
            {
                for(int k = 0; k < size2;k++)
                {
                    // якщо найшло цю мітку записує кількість стовпців які потрібно замінити
```

```
                {
                    // якщо найшло цю мітку записує кількість стовпців які потрібно замінити
                    arr[j][k] = POINT;
                    if(k == size2-1)
                        count++;
                }
            }
        }
    }
}

// фукнція видалення помічених функцією "point" стовпців
int** kill(int** arr,int** arr2,int size1, int size2)
{
    int tmpi = 0;
    for(int i = 0; i < size2;i++)
    {
        for(int j = 0; j < size1;j++)
        {
            // якщо найшло цю мітку,перериває цикл
            if(arr[i][j] == POINT)
            {
                break;
            }

            arr2[tmpi][j] = arr[i][j];

            //записує кількість стовпців які потрібно видалити
            if(j==size1-1)
                tmpi++;
        }
    }

    // вертає новий стовпчик
    return arr2;
}

// Звільняє пам'ять та видаляє масив
void freeArr(int** arr,int size1)
{
    for(int i = 0; i < size1;i++)
    {
        free(arr[i]);
    }
}

int main()
{
    int size1;
    int size2;
    printf("Print size of array\n");
    scanf("%d",&size2);
    scanf("%d",&size1);
    printf("\n");
```

```

    // створює наш масив
    int** arr = initialize(size1,size2);

    // ініціалізує наш масив
    create(arr,size1,size2);

    //виводить згенерований масив
    print(arr,size1,size2);
    point(arr,size1,size2);


    // ініціалізує наш масив але без данних стовпців
    int** arr1 = initialize(size1,size2-count);
    arr1 = kill(arr,arr1,size1,size2);

    // виводить наш масив але без данних стовпців
    print(arr1,size1,size2-count);

    // звільняє память
    freeArr(arr,size1);
    freeArr(arr1, size1-1);

}
```

Результат:

```
~/workspace/ $ make laba10
clang -fsanitize=signed-integer-overf
~/workspace/ $ ./laba10
Print size of array
4
4

14      19      8       6
3       10      5       18
13      5       4       18
14      1       14      6

Give me the element for deliting:
5

14      6
3       18
13      18
14      6
```