Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

**Алгоритмізація та програмування 2: [Процедурне програмування](https://classroom.google.com/u/1/c/NDQ5MDIwMzkxMzg1" \t "https://classroom.google.com/u/1/c/NDQ5MDIwMzkxMzg1/a/NDYwMzEwMzE1MTE4/_self)**

ЗВІТ

до лабораторної роботи №2

«**Динамічне виділення пам’яті для одно- та двовимірних масивів**»

*(ТЕМА)*

Варіант № 1

Дата «27» квітень 2022 Виконав: студент 1 курсу

гр. ТР-15

Руденко Владислав Ігорович

*(П.І.Б.)*

Оцінка «\_\_\_\_\_\_\_\_\_» Перевірив:

Крячок Олександр Степанович

Дата «\_\_\_» \_\_\_\_\_\_\_ 20\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(П.І.Б., підпис)*

Київ – 2022

***Завдання до роботи:***

1. Ознайомитись з особливостями роботи з динамічними одно- та двовимірними масивами.  
2. Розробити Блок-схему програмного алгоритму.  
3. Виконати індивідуальне завдання.  
4. Оформити ЗВІТ до лабораторної роботи згідно вимог та методичних рекомендацій.  
5. Вихідні дані (завдання) обрати згідно свого варіанта у Додатку B-2.  


Додаток В- 2

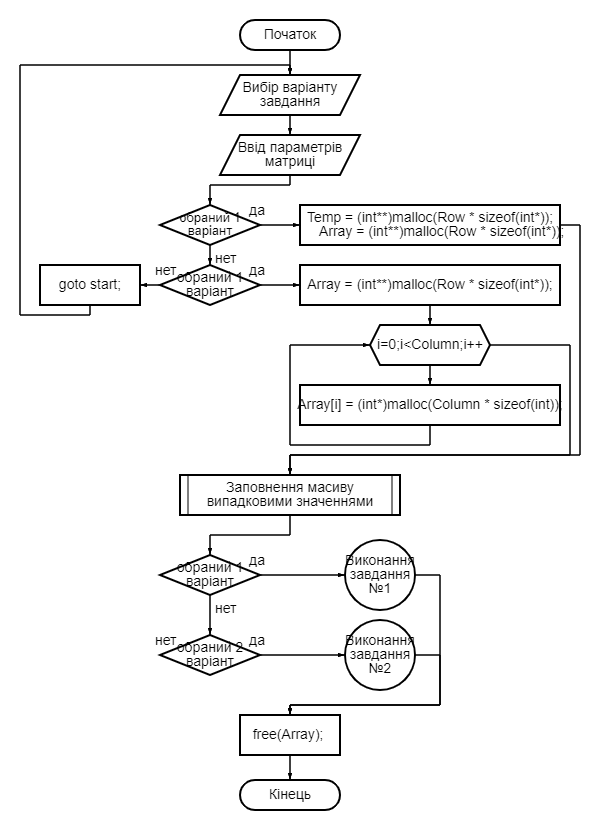
***Теоритичні відомості:***Для використання функцій динамічного виділення пам'яті необхідно описати покажчик, що є початковою адресою зберігання елементів масиву. Початкова адреса статичного масиву визначається компілятором в момент його оголошення і не може бути змінена.

Для динамічного масиву початкова адреса надається оголошеному покажчику на масив у процесі виконання програми.

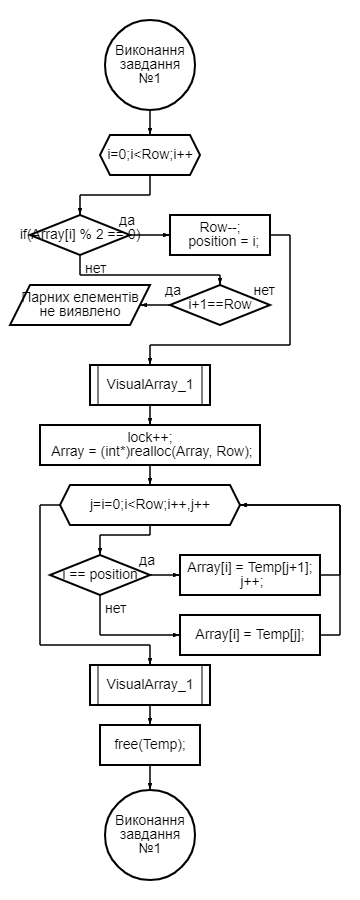
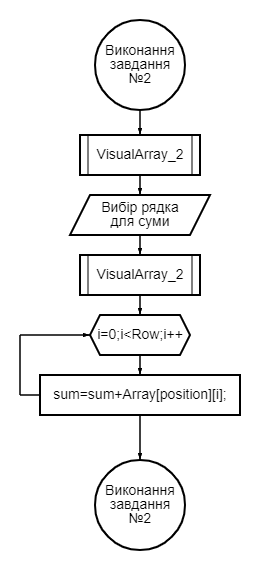
Основними функціями при динамічному виділенні пам’яті є:

***malloc()*** - функція використовується для ініціалізації вказівників необхідним обсягом пам'яті. Пам'ять виділяється з сектора оперативної пам'яті доступного для будь-яких програм, виконуваних на даній машині. Аргументом [функції malloc()](http://cppstudio.com/uk/post/856/) є кількість байт пам'яті, яку необхідно виділити, повертає функція – покажчик на виділений блок в пам'яті. Функція malloc() працює також як і будь-яка інша функція, нічого нового.

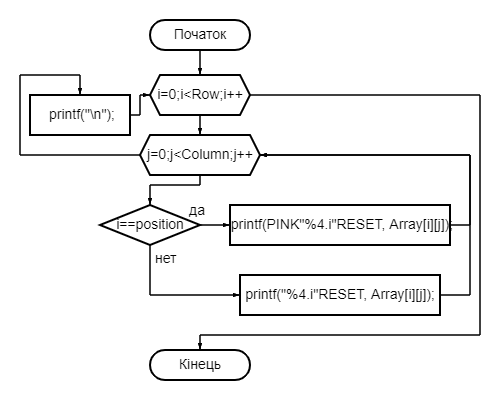
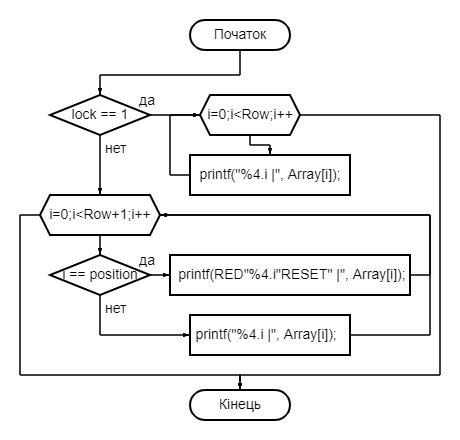
Так як різні типи даних мають різні вимоги до пам'яті, ми якось повинні навчитися отримати розмір в байтах для даних різного типу. Наприклад, нам потрібна ділянка пам'яті під масив значень типу int – це один розмір пам'яті, а якщо нам потрібно виділити пам'ять під масив того ж розміру, але вже типу char – це інший розмір. Тому потрібно якось обчислювати розмір пам'яті. Це може бути зроблено за допомогою операції sizeof(), яка приймає вираз і повертає його розмір. Наприклад, sizeof(int) поверне кількість байтів, необхідних для зберігання значення типу int.   
  
***Free(Array)*** - функція вивільнення пам’яті данного масиву.  
  
***realloc() -*** Змінює розмір раніше виділеного блоку пам'яті, початку якого вказує покажчик bl, до розміру ns байт. Якщо покажчик bl має значення NULL, тобто пам'ять не виділялася, то дія функції аналогічна дії malloc

***Блок схема алгоритму:  
***

Блок-Схема 1 (Основна частина коду)

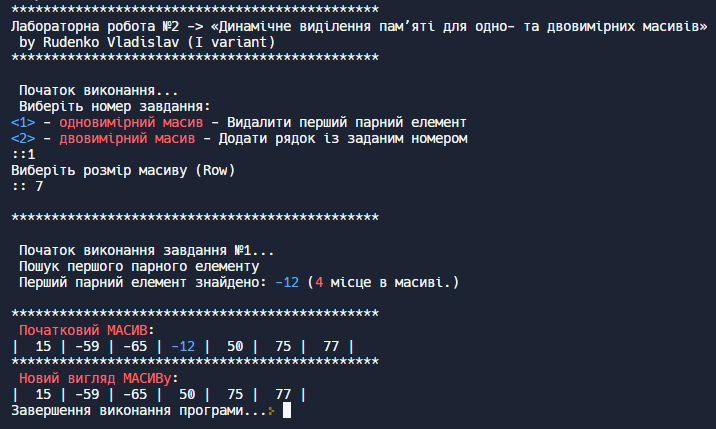
 

Блок-Схема 2 (Блок завдання 1) Блок-Схема 3 (Блок завдання 2)

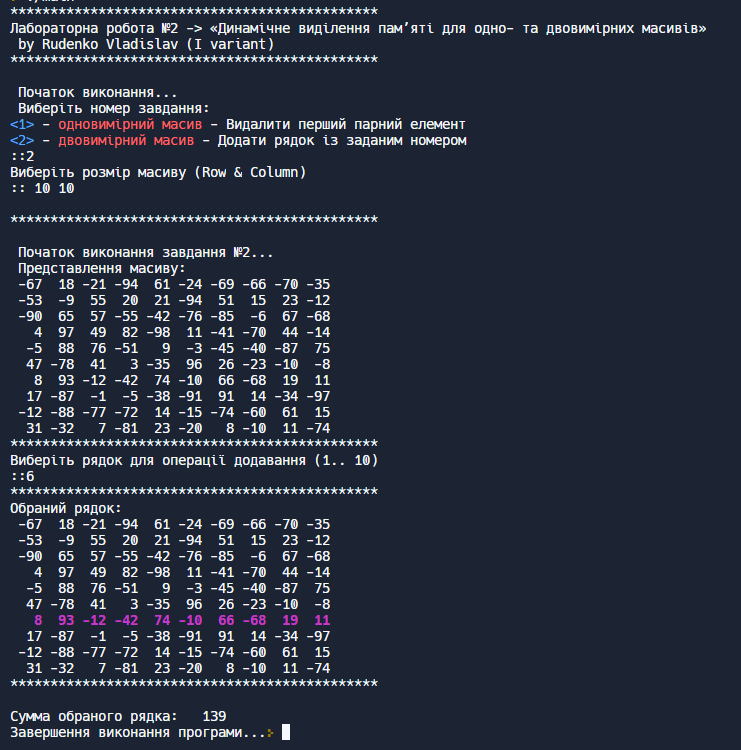
******

Блок-Схема 4 (Підпрограма VisualArray\_1 ) Блок-Схема 5 (Підпрограма VisualArray\_2)

***Результати роботи:***



Результати роботи - 1 (результати завдання 1)



Результати роботи - 2 (результати завдання 1)

***Посилання на програму в repl.it:  
<https://replit.com/join/giqxijlmzf-hetik>***

***Висновки:***

Під час виконання лабораторної роботи №2 було ознайомлено з особливостями роботи з динамічними одно- та двовимірними масивами. Виконане індивідуальне завдання в якому необхідно створити динамічний масив за допомогою нових функцій, в 1 завданні було видалено 1 парний елемент масиву, в 2 завданні було створено 2-вимірну матрицю де користувач обирає рядок, який буде додаватись. Під час виконання роботи було виявлено та виправлено декілька помилок.

***Програмний код (у текстовому варіанті):****#include <stdio.h>*

*#include <malloc.h>*

*#include <stdlib.h>*

*#define RED "\x1b[31m"*

*#define BLUE "\x1b[34m"*

*#define YELLOW "\033[1;33m"*

*#define PINK "\033[1;35m"*

*#define RESET "\x1b[0m"*

*///////////////////////////////////////////////////////////////////////*

*void VisualArray\_1(int\*\* Array,int Row,int Column, int position, int lock)*

*{*

*int i,j;*

*if(lock == 1)*

*{*

*printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n "RED"Новий вигляд МАСИВу"RESET":\n|");*

*for(i=0;i<Row;i++)*

*printf("%4.i |", Array[i]);*

*}*

*else*

*{*

*printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n "RED"Початковий МАСИВ"RESET":\n|");*

*for(i=0;i<Row+1;i++)*

*{*

*if (i == position)*

*printf(RED"%4.i"RESET" |", Array[i]);*

*else*

*printf("%4.i |", Array[i]);*

*}*

*}*

*}*

*///////////////////////////////////////////////////////////////////////*

*void VisualArray\_2(int\*\* Array,int Row,int Column, int position)*

*{*

*printf("");*

*int i,j;*

*for(i=0;i<Row;i++)*

*{*

*for(j=0;j<Column;j++)*

*{*

*if(i==position)*

*printf(PINK"%4.i"RESET, Array[i][j]);*

*else*

*printf("%4.i"RESET, Array[i][j]);*

*}*

*printf("\n");*

*}*

*printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");*

*}*

*///////////////////////////////////////////////////////////////////////*

*int main(void)*

*{*

*srand(time(NULL));*

*int \*\*Array;*

*int \*\*Temp;*

*int choose\_ex, i, j, Row, Column, Error = 0, position=-1, BYBA = 0,finish=0, sum=0, lock = 0;*

*char temp; FILE \*XIXI;*

*printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nЛабораторна робота №2 -> «Динамічне виділення пам’яті для одно- та двовимірних масивів» \n by Rudenko Vladislav (I variant)\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n Початок виконання...");*

*start:*

*printf("\n Виберіть номер завдання: \n"BLUE"<1>"RESET" - "RED"одновимірний масив"RESET" - Видалити перший парний елемент\n"BLUE"<2>"RESET" - "RED"двовимірний масив"RESET" - Додати рядок із заданим номером\n::");*

*scanf("%i", &choose\_ex);*

*switch(choose\_ex)*

*{*

*case 1:*

*printf("Виберіть розмір масиву (Row)\n:: ");*

*scanf("%i", &Row);*

*Temp = (int\*\*)malloc(Row \* sizeof(int\*));*

*Array = (int\*\*)malloc(Row \* sizeof(int\*));*

*break;*

*case 2:*

*printf("Виберіть розмір масиву (Row & Column)\n:: ");*

*scanf("%i %i", &Row, &Column);*

*Array = (int\*\*)malloc(Row \* sizeof(int\*));*

*for (i=0;i<Column;i++)*

*Array[i] = (int\*)malloc(Column \* sizeof(int));*

*break;*

*case 15:*

*//Пасхалка*

*printf(RED"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"BLUE);*

*XIXI = fopen("XIXI.txt", "r");*

*fseek(XIXI, 0,0);*

*while(feof(XIXI) == 0)*

*{*

*temp = getc(XIXI);*

*if(temp == 'S')*

*printf(YELLOW);*

*else if(feof(XIXI) == 0)*

*printf("%c",temp);*

*}*

*fclose(XIXI);*

*printf(RED"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"RESET);*

*Error = 1;*

*break;*

*default:*

*printf(RED"Error..."RESET" Ви вибрали недопустиме значення. \nВ наступний рза пощастить!\n\n");*

*Error = 1;*

*break;*

*}*

*if (Error == 1)*

*{*

*Error=0;*

*goto start;*

*}*

*///////////////////////////////////////////////////////*

*for(i=0;i<Row;i++)*

*{*

*if (choose\_ex == 2)*

*for(j=0;j<Column;j++)*

*Array[i][j] =-99 + rand()%198;*

*else*

*Temp[i] = Array[i] =-99 + rand()%198;*

*}*

*///////////////////////////////////////////////////////*

*switch(choose\_ex)*

*{*

*case 1:*

*printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n Початок виконання завдання №1...\n Пошук першого парного елементу\n");*

*for(i=0;i<Row;i++)*

*{*

*BYBA = Array[i];*

*if(BYBA % 2 == 0)*

*{*

*Row--;*

*position = i;*

*printf(" Перший парний елемент знайдено: "BLUE"%i"RESET" ("RED"%i"RESET" місце в масиві.)\n", Array[i], i+1);*

*break;*

*}*

*else if (i+1==Row){*

*printf(RED"Парних елементів не знайдено..."RESET"\n Завершення виконання...");*

*exit(0);*

*}*

*}*

*///////////////////////////////////*

*VisualArray\_1(Array,Row,Column,position,lock);*

*lock++;*

*///////////////////////////////////*

*Array = (int\*)realloc(Array, Row);*

*for(j=i=0;i<Row;i++,j++)*

*{*

*if(i == position)*

*{*

*Array[i] = Temp[j+1];*

*j++;*

*}*

*else*

*Array[i] = Temp[j];*

*}*

*///////////////////////////////////*

*VisualArray\_1(Array,Row,Column,position,lock);*

*free(Temp);*

*break;*

*case 2:*

*printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n Початок виконання завдання №2...\n Представлення масиву: \n");*

*VisualArray\_2(Array,Row,Column,position);*

*printf("Виберіть рядок для операції додавання (1..%3.i)\n::",Row);*

*scanf("%i",&position);*

*printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nОбраний рядок:\n");*

*VisualArray\_2(Array,Row,Column,position);*

*for(i=0;i<Row;i++)*

*sum=sum+Array[position][i];*

*printf("\nСумма обраного рядка: %5.i", sum);*

*break;*

*}*

*free(Array);*

*printf("\nЗавершення виконання програми...");*

*}*