

Міністерство освіти і науки України
Черкаський державний технологічний університет
Кафедра програмного забезпечення автоматизованих систем

Звіт

Про виконання лабораторної роботи №1
з дисципліни “Проектний практикум”

Перевішив:

Асистент Кафедри ПЗАС

Півень О. Б.

Виконав:

Студент 3-го курсу

Групи ПЗС-1644

Близнюк А. О.

Черкаси 2017

Лабораторна робота № 1

Тема роботи: Динамічні масиви.

Мета роботи: Набути навиків роботи з одно- та багатовимірними динамічними масивами.

Завдання(Варіант 2):

1. Створити масив випадкових цілих чисел A , довжина якого вводиться із клавіатури. Створити масив B , в який скопіювати послідовність елементів з A до першого входження числа 0. Знайти суму негативних елементів масиву B .
2. Створити масив випадкових цілих чисел розмірності $m \times n$ (розмірність ввести з клавіатури). Скопіювати у новий масив максимальні елементи кожного стовпчика початкового масиву.

Теоретичні відомості

У комп'ютерній науці, динамічне виділення пам'яті (також відоме як виділення пам'яті на основі купи) є розподілом пам'яті для її використання у комп'ютерній програмі під час виконання цієї програми.

Динамічно виділена пам'ять існує, поки її явно не звільнив програміст або збиральник сміття. Це відрізняється від статичного розподілу пам'яті, яка має фіксовану тривалість. Кажуть, що об'єкт, щоб виділяються має динамічний час життя.

Як правило, пам'ять виділяється з великого резерву невикористаних областей пам'яті, званої купою. Оскільки точне місце розташування виділених областей не відомо заздалегідь, то доступ до пам'яті є непрямим, як правило, через вказівники.

Змінна — це абстрактна назва комірки чи декількох комірок пам'яті. Кожна змінна має шість атрибутів: ім'я, адресу, значення, тип, область видимості і час життя. Адреса змінної — це адреса комірки пам'яті, зв'язаної з даною змінною.

Динамічні змінні не мають імен. Вони розташовуються в динамічній пам'яті, чи купі, і видаляються з неї під час виконання програми. Для звертання до цих змінних потрібні особливі засоби: вказівники і посилання.

Динамічні змінні створюються або оператором `new`.

Знищення динамічних змінних здійснюється оператором `delete`.

Лістинг програми

Завдання 1

```
#include <iostream.h>
#include <conio.h>
#include <time.h>

int main() {
    srand(time(NULL));
    int n,n1=0;
    int countMin = 0;
    cout << "Array size = \n";
    cin >> n;
    n1 = n;
    int *arrayA = new int(n);
    for(int i = 0; i < n; i++) {
        arrayA[i] = rand()%(1000)/100.-5;
    }

    int *arrayB = new int(n1);
    cout << "Array A:\n\n";
    for(int i = 0; i < n; i++) {
        cout << i << " " << arrayA[i] << endl;
    }
    for(int i = 0; i < n; i++) {
        if(arrayA[i] != 0) {
            arrayB[i] = arrayA[i];
        }
        else {
            n1 = i;
            break;
        }
    }
    cout << "Array B:\n\n";
    for(int i = 0; i < n1; i++) {
        cout << i << " " << arrayB[i] << "\n";
        if(arrayB[i] < 0) {
            countMin+=arrayB[i];
        }
    }
    cout << "Count of sum in array B value less than 0 = " << countMin << endl;
    getch();
    delete[] arrayA;
    delete[] arrayB;
    return 0;
}
```

Завдання 2

```
#include <iostream.h>
#include <conio.h>
#include <time.h>
#include <windows.h>
#include <iomanip.h>

int main() {
    srand(time(0));
    cout << "Enter the number of rows: ";
    int rows_number = 0;
```

```
cin >> rows_number;
cout << "Enter the number of columns: ";
int columns_number = 0;
cin >> columns_number;
int n1=0;
int max = 0;

int **ptrarray_a = new int* [rows_number];
int *arrayB = new int(n1);

for (int count = 0; count < rows_number; count++)
    ptrarray_a[count] = new int [columns_number];

for (int count_row = 0; count_row < rows_number; count_row++)
    for (int count_column = 0; count_column < columns_number;
count_column++)
        ptrarray_a[count_row][count_column] = rand() % 10 - 2;

for (int count_row = 0; count_row < rows_number; count_row++)
{
    for (int count_column = 0; count_column < columns_number;
count_column++)
        cout << setw(2) << ptrarray_a[count_row][count_column] << " ";
    cout << endl;
}

int *ptrarray = new int [columns_number];
max = ptrarray_a[0][0];
for (int count = 0; count < columns_number; count++)
    ptrarray[count] = 0;

for (int count = 0; count < columns_number; count++)
    if (ptrarray[count] != 0)
        cout << setw(2) << ptrarray[count] << " ";
    else
        cout << setw(3) << " ";
cout << endl;

for (int count_column = 0; count_column < columns_number; count_column++)
{
    max = 0;
    for (int count_row = 0; count_row < rows_number; count_row++) {
        if (ptrarray_a[count_row][count_column] > max) {
            max = ptrarray_a[count_row][count_column];
        }
    }
    n1++;
    arrayB[count_column] = max;
}

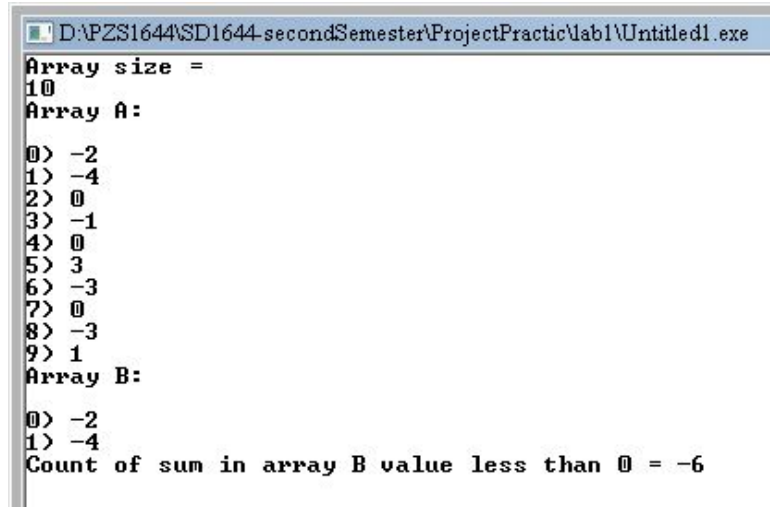
cout << endl;
cout << "max columns value: \n";
for (int i = 0; i < n1; i++) {
    cout << i << " col. = " << arrayB[i] << "\n";
}
cout << endl;
for (int count = 0; count < 2; count++)
    delete []ptrarray_a[count];

delete [] ptrarray;
```

```
    getch();  
    return 0;  
}
```

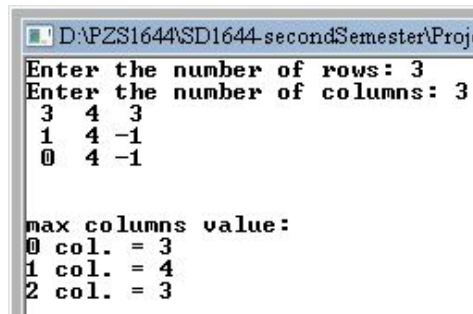
Результат роботи

Завдання 1



```
D:\PZS1644\SD1644-secondSemester\ProjectPractic\lab1\Untitled1.exe  
Array size =  
10  
Array A:  
0> -2  
1> -4  
2> 0  
3> -1  
4> 0  
5> 3  
6> -3  
7> 0  
8> -3  
9> 1  
Array B:  
0> -2  
1> -4  
Count of sum in array B value less than 0 = -6
```

Завдання 2



```
D:\PZS1644\SD1644-secondSemester\ProjectPractic\lab1\Untitled1.exe  
Enter the number of rows: 3  
Enter the number of columns: 3  
3 4 3  
1 4 -1  
0 4 -1  
  
max columns value:  
0 col. = 3  
1 col. = 4  
2 col. = 3
```

Висновок

На даній лабораторній роботі оволодів навичками створення і використання динамічних змінних. Набув навиків з одновимірними та багатовимірними динамічними масивами.