# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ТАРАСА ШЕВЧЕНКА



## ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №12**

# з курсу

**«Алгоритми і структури даних»**

*Студента 1 курсу*

*групи ПП-11 спеціальності 122 «Комп'ютерні науки» ОП «Прикладне програмування»*

%username%

*Викладач:*

д.е.н., к.т.н., проф. Плескач В.Л.

к.ф.-м.н., доц. Шолохов О.В.

## Київ – 202

**1.Назва роботи**

Незв’язані динамічні структури даних

1. **Тема роботи**

Незв’язані динамічні структури даних

1. **Мета роботи**

Засвоєння теоретичного матеріалу та набуття практичного досвіду

використання незв’язаних динамічних структур даних при складанні різних алгоритмів.

1. **Умова завдання**

(4 варіант)

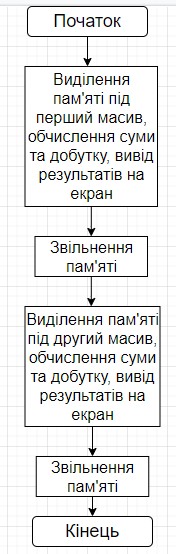
Дано дві матриці A[6,6], B[8,8]довгого цілого типу. В кожній з матриць

знайти суму елементів над головною діагоналлю і добуток елементів під нею.

1. **Рішення**

Виділяємо пам’ять під перший масив, обчислюємо задані параметри, виводимо їх на екран та одразу ж звільняємо пам’ять. Виділяємо пам’ять під другий масив, знов обчислюємо ті ж самі параметри, виводимо на екран та звільняемо виділену пам’ять.

Блок-схема:



Код:

using System;

namespace Лаба12

{

class Program

{

static void Main(string[] args) {

int[,] firstMatrix =

{

{1, 2, 4, 2, 7, 8 },

{8, 2, 4, 1, 7, 5 },

{1, 2, 4, 4, 1, 3 },

{2, 2, 2, 1, 1, 8 },

{4, 2, 4, 1, 3, 6 },

{6, 2, 4, 1, 7, 5 }

};

int sum = 0;

int prod = 1;

for (int q = 0; q < 6; q++) {

for (int w = 0; w < 6; w++) {

if (q > w) {

sum += firstMatrix[q, w];

}

}

}

for (int q = 0; q < 6; q++) {

for (int w = 0; w < 6; w++) {

if (q > w) {

prod \*= firstMatrix[q, w];

}

}

}

Console.WriteLine(sum);

Console.WriteLine(prod);

GC.Collect();

GC.WaitForPendingFinalizers();

int[,] secondMatrix =

{

{1, 5, 7, 8, 4, 7, 3, 9 },

{7, 6, 7, 6, 4, 3, 3, 5 },

{1, 9, 7, 2, 4, 5, 1, 1 },

{4, 3, 7, 8, 4, 3, 3, 9 },

{1, 1, 9, 4, 4, 2, 7, 1 },

{4, 3, 1, 8, 1, 4, 1, 9 },

{1, 2, 7, 4, 4, 7, 8, 9 },

{1, 4, 1, 8, 1, 9, 3, 9 }

};

for (int q = 0; q < 6; q++) {

for (int w = 0; w < 6; w++) {

if (q > w) {

sum += secondMatrix[q, w];

}

}

}

for (int q = 0; q < 6; q++) {

for (int w = 0; w < 6; w++) {

if (q > w) {

prod \*= secondMatrix[q, w];

}

}

}

Console.WriteLine(sum);

Console.WriteLine(prod);

GC.Collect();

GC.WaitForPendingFinalizers();

}

}

}

Результат роботи програми:

[screenshot]

**Контрольні питання**

1. Яких типів допускаються динамічні дані?

Ті, котрим пам’ять під них відводиться під час виконання програми.

При динамічному виділенні пам'ять виділяється з кучі (heap)

2. В чому полягає відмінність оголошення звичайних змінних і незв’язаних  
динамічних змінних такого самого типу?

При оголошенні незв’язних динамічних змінних, ми оголошуємо спочатку вказівник, виділяємо пам’ять, а потім присвоюємо йому адресу необхідного об’єкту

3. Що виконує операція розйменування динамічної змінної?

Ця операція дозволяє отримати значення об’єкту (змінної) за адресою, який зберігається в покажчику.

4. За допомогою яких процедур чи функцій виконується виділення пам’яті для динамічних даних?

За допомогою оператора new

1. В яких станах може знаходитись динамічна змінна?

Визначеному, невизначеному (вказує на “сміття”), null, висячому (вказує на звільнену пам’ять)

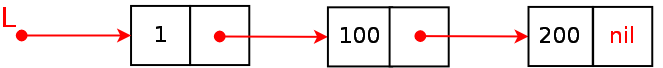
1. В яких випадках динамічна змінна може знаходитись у невизначеному стані?

Коли покажчик не ініціалізувано – тоді там може зберігатися довільне «сміття»;

коли пам’ять, на яку вказував покажчик, було звільнено (такі покажчики називають висячими); за умови розіменування нульового покажчику

7. В чому полягає відмінність стану динамічної змінної nil та її невизначеного стану?

Як аналог нуля для ссилочних змінних прийнято спеціальне значення Nil: якщо змінна має значення Nil, то це означає, що вона не вказує ні на яку змінну. Значення Nil в поле покажчика має завжди перша компонента ланцюжка динамічних змінних.



Значення Nil можна заслати оператором присвоювання: L: = nil; якщо L = nil, то ланцюжки порожня.

Щоб визначити, що даний елемент є першим в ланцюжку змінних, досить перевірити на Nil значення поля покажчика цієї змінної.

8. Які значення будуть порівнюватися, якщо виконується операція порівняння над ідентифікаторами динамічних змінних?

Числове значення адреси комірок пам’яті на які вони вказують

9. В чому полягають динамічні властивості незв’язаних динамічних даних?

Одна і та ж пам'ять може бути використана для зберігання різної інформації в процесі виконання програми. Оскільки пам'ять виділяється для певної мети і звільняється, коли її використання завершилося, то можна використовувати ту ж саму пам'ять в інший момент часу для інших цілей в іншій частині програми. Іншою перевагою динамічного виділення пам'яті є можливість створення з її допомогою пов'язаних списків, довічних дерев і інших динамічних структур даних.

Ядром динамічного виділення пам'яті мови С є функції malloc () і free (), що є частинами стандартної бібліотеки. Всякий раз, коли функцією malloc () здійснюється запит на виділення пам'яті, виділяється порція наявної вільної пам'яті. Всякий раз, коли ця пам'ять звільняється за допомогою функції free (), ця пам'ять повертається назад системі. За допомогою покажчиків можна зручно організовувати, наприклад, пов'язані списки, динамічне виділення пам'яті і т.д.

1. В яких випадках доцільно використовувати незв’язані динамічні дані?

Реалізація масивів за допомогою покажчиків (покажчики можуть використовуватися для ітерації масивом); динамічне виділення пам'яті в C ++, оскільки у цій мові програмування покажчики - єдиний спосіб динамічного виділення даних; передача великої кількості даних в функцію без копіювання цих даних; передача однієї функції в якості параметра іншої функції; поліморфізм при роботі з успадкуванням; представлення однієї структури / класу в іншій структурі / класі, формуючи, таким чином, цілі ланцюжки.

1. **Висновки**

В результаті виконання даної лабораторної роботи я оволодів практичними навичками використання незв’язаних динамічних структур даних, навчився складати програми для виконання операцій з використанням даних структур. Вважаю дану лабораторну роботу виконаною в повному обсязі.