Динамічні структури даних - це будь-яка структура даних, обсяг пам'яті якої не є фіксованим. Приклади динамічних структур даних: однозв'язний і двозв'язний списки, дерево, купа, черга з пріоритетом, динамічний масив.

Статичні структури даних - це будь-яка структура даних, обсяг пам'яті якої є фіксованим, тобто не може змінюватись при роботі програми. Приклади статичних структур даних: стек, ДЕК, черга, масив.

**Розглянемо динамічний масив.**

Задається динамічний масив за допомогою команди new int, приклад:

int \* arr = new int [5];

Але при цьому треба не забувати звільняти пам'ять командою delete []:

delete [] arr;

На відміно від статичного масиву, для задання динамічного не обов’язково використовувати константні значення, користувач може ввести зміну під час роботи програми і цю змінну можна використати для задання розміру масиву, приклад:

int n;

cin >> n;

int \* arr = new int [n];

Якщо не звільняти пам'ять, то система не зможе її використовувати, через що, коли пам'ять закінчиться, програма аварійно завершить роботу, це явище отримало спеціальну назву *витік пам'яті*.

Так само ми можемо ініціалізувати значення динамічного масиву при оголошенні

int \* arr = new int [5] {9, 7, 5, 3, 1};

При цьому якщо значень у фігурних дужках більше ніж розмір масиву то надасть помилку, якщо менше, то не ініціалізовані елементи дорівнюватимуть нулю.

Також як і зі статичними масивами, можна створювати багатовимірні динамічні масиви, приклад оголошення двовимірного масиву:

int \*\* array = new int \* [5];

for (int i = 0; i < 5; i++)

array [i] = new int [5];

Коли у динамічному масиві нема вільних елементів, то нові вільні елементи треба додати за допомогою окремої функції:

Int main()

{

int n = 10;

int \*arr = new int[n];

…

arr = addsize(arr, n) // виклик функції для розширення динамічного масиву

…

}

int \*addsize(int \*arr, int n)

{

n++;

int \*arr2 = new int[n];

for (int i = 0; i < n; i++)

arr2[i] = arr[i];

delete[] arr;

return arr2;

}