Структури даних. Масиви

Масив – це великий простір чогось однорідного за типом.

З Оксфордського словника англійської мови

Масиви

Масив - структура даних (статична), що складається з фіксованої кількості елементів, одного типу.

Змінна масив:

- складається з елементів (компонент);
- всі елементи одного типу;
- кількість елементів фіксується в означенні;
- кожен елемент ідентифікується номером (індексом);
- займає неперервну область пам`яті при розміщенні;
- доступність елемента (час) не залежить від номеру.

Математика:

- вектори, матриці;
- функція A: <індекси> → <елементи>

Масиви

Визначення:

<тип> <ім`я>[<кількість елементів>] [<ініціалізатор>];

Пам`ять:

<об`єм пам`яті> = sizeof(<тип>) * <кількість елементів>

Звернення:

<ім`я> [<номер елемента>]

Зауваження:

- нумерація елементів (індекси) починаються з 0;
- для елементів глобальних масивів автоматично - ініціалізація 0;
- відсутній контроль на вихід індексу за межі.

Приклади

```
long vect[10];
vect[0] = 3;   vect[1] = vect[0]  * 2; vect[i+k] = vect[1];

long arr[3] = {10, 20, 30};

long arr[3] = {10, 20};     // arr[0]=10, arr[1]=20, arr[2]=0

long arr[] = {10, 20, 30};

float cost[30], nm[7];

char dig[10];

kilk_el = sizeof arr / sizeof (long);
```

Приклад

```
const short arr_size = 20;
int arr[arr_size];
for (int i=0; i<arr_size; ++i) {
    arr[i] = 2*i+2; }
//виведення в прямому та оберненому порядках
for (int i=0; i<arr_size; ++i) {
  cout << i << "   " << arr[i] << endl;}
cout << "-----------------------------" << endl;
for (int i=arr_size-1; i>=0; --i) {
  cout << i << "   " << arr[i] << endl;}
```

Приклад

```
const short arr_size = 20;
int arr[arr_size], el=2;
for (int i=0; i<arr_size; ++i) {
  arr[i] = el; el += 2; }
//виведення в прямому та оберненому порядках
for (int i=0; i<arr_size; ++i) {
  cout << i << "   " << arr[i] << endl;}
cout << "-----------------------------" << endl;
for (int i=arr_size-1; i>=0; --i) {
  cout << i << "   " << arr[i] << endl;}
```

Приклад

```
int main() {
 const int n = 20;
 int i, sum;
 int marks[n] = {3, 4, 5, 4, 4, 4};

for (i=0, sum=0; i<n; i++) sum += marks[i];

 cout << "  Sum = " << sum << endl;

 return 0;
}
```

Приклад сортування вибором

Метод :

для кожного i від 0 до n-1
  знайти a[k] - найменший серед a[i], …, a[n-1]
  поміняти місцями a[i] та a[k]

Використовує $O(n^2)$ операцій (порівнянь).

Багатовимірні масиви

Визначення:

<тип> <ім`я>[<кількість1>] …[<кількістьN>] [<ініціалізатор>];

Звернення:

<ім`я> [<номер1>] …[< номерN>]

Зауваження:

- при розташуванні швидше змінюється останній індекс (“рядками”);
- для ініціалізації значення вказуються згідно з порядком розташування;
- при зверненні кожний індекс у власних дужках.

Приклади

```
int matr[2][4];
 matr[1][i+j] = 5;

int matr[2][4] = {1, 2, 3, 4,
                  5, 6, 7, 8};

int matr[2][4] = {{1, 2, 3, 4},
                  {5, 6, 7, 8}};

int matr[][4] = {{1, 2, 3, 4},
                 {5, 6, 7, 8}};
```

Приклад

В матриці з цілих чисел визначити номер рядка з максимальною кількістю нулевих елементів.

Приклад

Злиття двох впорядкованих одновимірних масивів.

Масиви – параметри функцій

- Визначення масиву – відповідна змінна зберігає адресу першого елементу. Доступ до елементів можливий як за індексом, так й шляхом адресної арифметики.
- Формальні параметри описуються традиційним чином (вказуючи тип елементів та їх кількість).
- Фактичний параметр – ім`я масиву.
- Виклик функції передає фактично адресу першого елемента масиву.
- Передача таких параметрів “***по посиланню***”.

Зауваження

- При визначенні вказується кількість елементів. Індексація починається з 0.
- Потрібно не допускати виходу значення індексу за межі визначеного діапазону.
- Для багатовимірних масивів [i][j] не можна замінити [i,j].

Підсумки

- Розглянули лише найпростіші можливості що до створення та використання масивів.
- Але навіть розглянуті можливості дозволяють суттєво розширити клас задач.

Задачі

- Оптимальний розрахунок здачі:
  - необмежені ресурси;
  - обмежені ресурси.
- Для матриці розміром n*m визначити кількість “vip” елементів:
  - а) більше суми всіх інших елементів свого стовпчика;
  - б) у рядку ліворуч всі елементи менші, а праворуч більші.
- Визначити чи є квадратна матриця симетричною.
- Здійснити транспонування квадратної матриці.

Задачі

- Для дійсної матриці розміром n*m впорядкувати її рядки за не спаданням:
  - а) їх перших елементів;
  - б) суми їх елементів;
  - в) їх найбільших елементів.
- Для заданої дійсної матриці знайти індекси всіх її “сідлових точок” (елементи, що є одночасно найменшими у рядку й найбільшими у стовпчику, або навпаки.)
- Визначити чи є ціла квадратна матриця “магічним квадратом” (суми елементів у всіх рядках та стовпчиках однакові).

Задачі

- На вході послідовність рядкових букв латинського алфавіту, що закінчується “.” . Підрахувати кількість різних пар букв.
- В місті M діє p-ічна система числення, а номери тролейбусних квитків містять 2k розрядів. Квиток вважається щасливим, якщо сума перших k розрядів дорівнює сумі останніх k розрядів.

  Вхід: Значення p та k.

  Вихід: Кількість щасливих квитків.