



Phần trình bày của:

**ĐẬU HẢI PHONG**

*Giảng viên*

*Đại Nam, ngày 19 tháng 01 năm 2023*

# LƯU Ý

**KHÔNG NÓI  
CHUYỆN RIÊNG**



**KHÔNG SỬ DỤNG  
ĐIỆN THOẠI**



**KHÔNG NGỦ GẬT**



**GHI CHÉP ĐẦY ĐỦ**



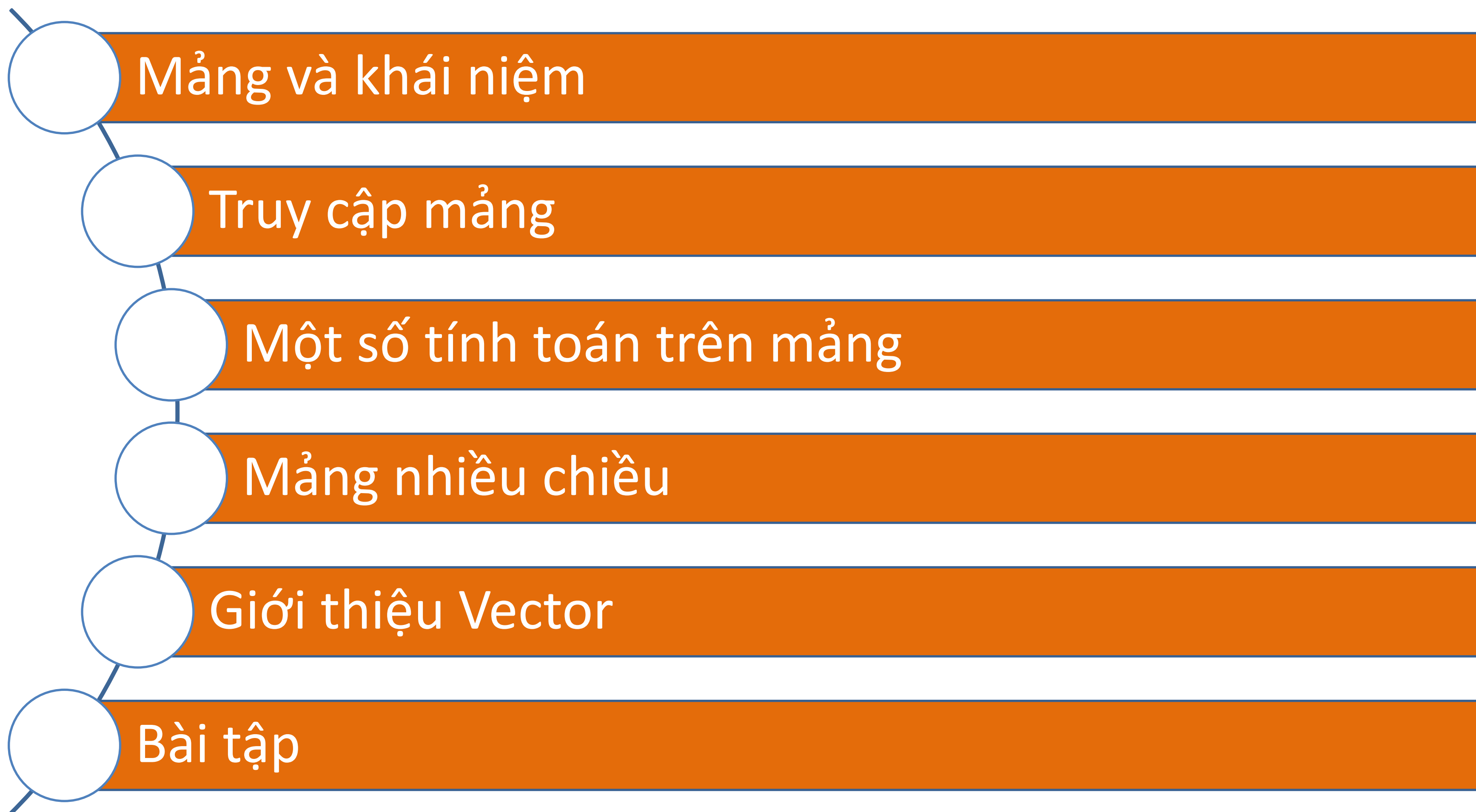


# LẬP TRÌNH CƠ BẢN



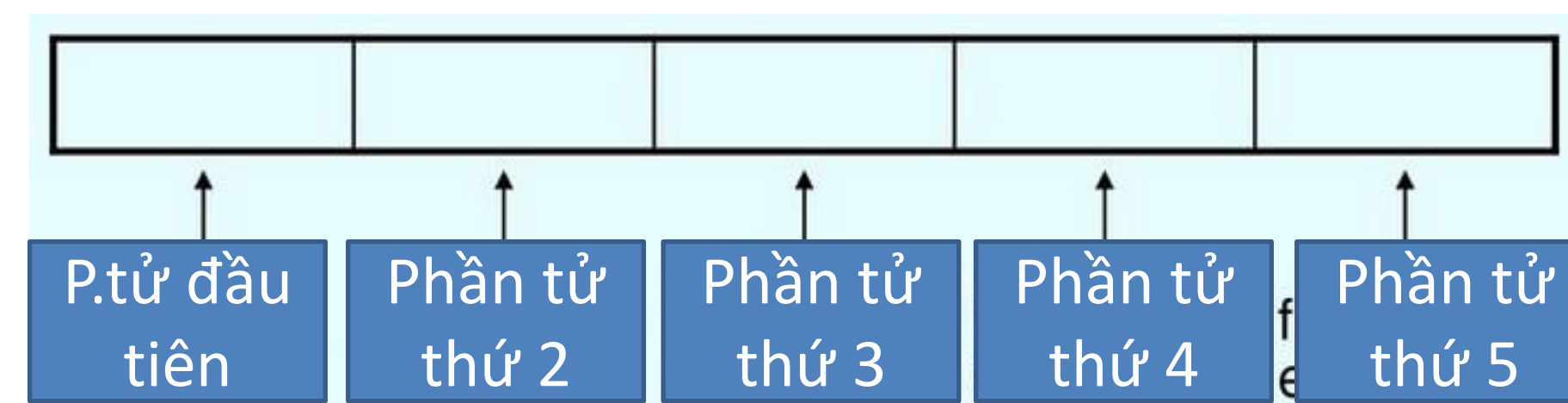
# CHƯƠNG 6

## MẢNG VÀ VECTOR



# Mảng & khái niệm liên quan

- Mảng: biến có thể lưu trữ **nhiều giá trị có cùng kiểu**
  - Giá trị lưu trữ trong **vị trí ô nhớ liên tiếp nhau**
  - Khai báo dùng toán tử []
  - Ví dụ: khai báo mảng
    - `int tests[5];`
    - Trong đó:
      - `int` là kiểu dữ liệu của các phần tử trong mảng;
      - `Tests` là tên mảng;
      - `5` là số lượng phần tử trong mảng (kích thước khai báo)
    - Bộ nhớ được cấp phát như sau:



# Mảng & khái niệm liên quan

- Kích thước của mảng = số lượng phần tử \* kích thước mỗi phần tử
- Ví dụ:
  - `int tests[5]` là 1 mảng 20 byte, giả sử `int` là 4 byte
  - `long double measures[10]` là 80 byte, giả sử `long double` là 8 byte
- Khai báo kích thước mảng thường là hằng số  
`const int SIZE = 5;`  
`int tests[SIZE];`

# Truy cập mảng

- Mỗi phần tử trong mảng được gán một phần tử duy nhất
- Chỉ số bắt đầu từ 0
- Chỉ số cuối cùng là  $n-1$  nếu mảng có  $n$  phần tử.



- Các phần tử của mảng có thể được **sử dụng như 1 biến**
  - `tests[0] = 79;`
  - `cout<<tests[0];`
  - `cin>>tests[0];`
  - `tests[4] = tests[0] + tests[1];`
- Mảng phải được truy cập qua từng phần tử
  - `cout<<tests; //`**Sai**

# Truy cập mảng

## Program 7-1

```
1 // This program asks for the number of hours worked
2 // by six employees. It stores the values in an array.
3 #include <iostream>
4 using namespace std;
5
6 int main()
7 {
8     const int NUM_EMPLOYEES = 6;
9     int hours[NUM_EMPLOYEES];
10
11     // Get the hours worked by each employee.
12     cout << "Enter the hours worked by "
13         << NUM_EMPLOYEES << " employees: ";
14     cin >> hours[0];
15     cin >> hours[1];
16     cin >> hours[2];
17     cin >> hours[3];
18     cin >> hours[4];
19     cin >> hours[5];
20
```

```
21 // Display the values in the array.
22 cout << "The hours you entered are:";
23 cout << " " << hours[0];
24 cout << " " << hours[1];
25 cout << " " << hours[2];
26 cout << " " << hours[3];
27 cout << " " << hours[4];
28 cout << " " << hours[5] << endl;
29 return 0;
30 }
```

### Program Output with Example Input Shown in Bold

Enter the hours worked by 6 employees: **20 12 40 30 30 15** [Enter]  
The hours you entered are: 20 12 40 30 30 15

của mảng hours, với các  
giá trị được gán ở dòng

hours[0]	hours[1]	hours[2]	hours[3]	hours[4]	hours[5]
20	12	40	30	30	15

# Truy cập mảng

- Có thể truy cập phần tử của mảng với 1 hằng số hoặc chỉ số.
  - `cout<<tests[3]<<endl;`
- Có thể sử dụng biểu thức nguyên làm chỉ số:
  - `int i = 3;`
  - `cout<<tests[i]<<endl;`
- Sử dụng vòng lặp để duyệt các phần tử mảng
  - VD: Định nghĩa mảng numbers và gán 99 cho các phần tử của mảng  
`const int ARRAY_SIZE = 5;`  
`int numbers[ARRAY_SIZE];`  
`for (int count = 0; count < ARRAY_SIZE; count++)`  
`numbers[count] = 99;`

# Khởi tạo giá trị cho mảng

- VD1: Khởi tạo số ngày cho các tháng

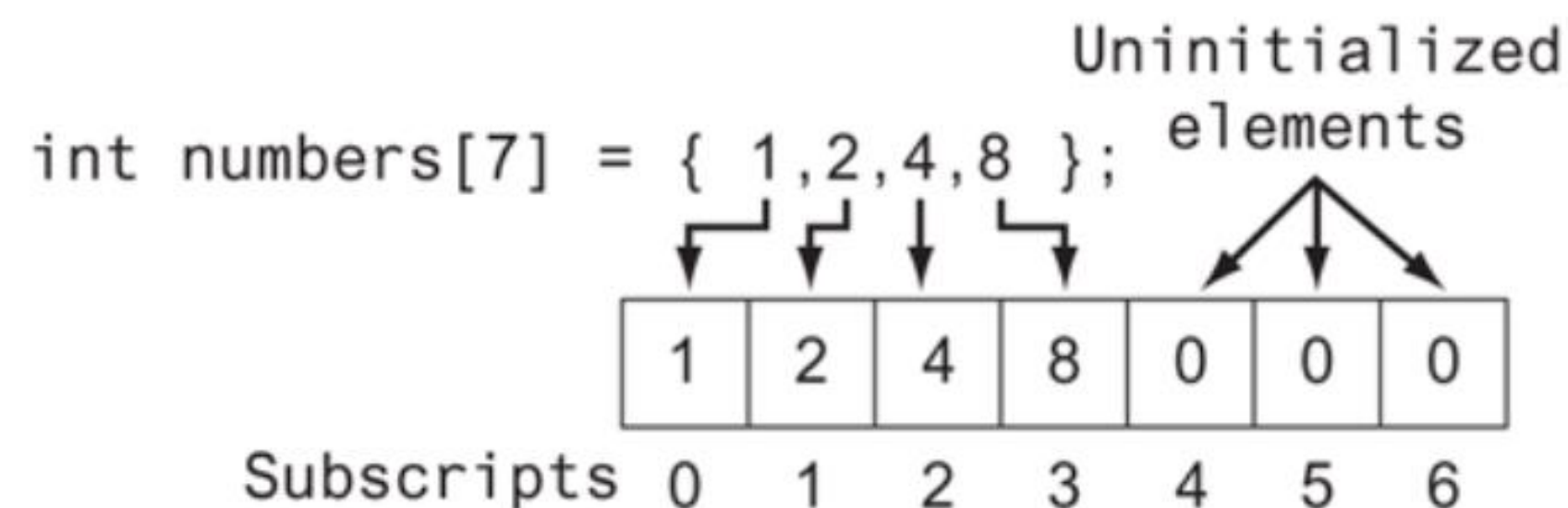
```
const int MONTHS = 12;
```

```
int days[MONTHS] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
```

0	1	2	3	4	5	6	7	8	9	10	11
31	28	31	30	31	30	31	31	30	31	30	31

- VD2: Khởi tạo 1 phần của mảng

```
int numbers[7] = {1, 2, 4, 8};
```



- Không khởi tạo giá trị ngắt quãng cho mảng

```
int numbers[6] = {2, 4, , 8, , 12}; // SAI
```

- Kích thước ngầm định

```
double ratings[] = {1.0, 1.5, 2.0, 2.5, 3.0}; //Gồm 5 phần tử
```

# Đọc/Ghi từ tệp vào mảng

```
1 // This program reads data from a file into an array.
2 #include <iostream>
3 #include <fstream>
4 using namespace std;
5 int main()
6 {
7     const int ARRAY_SIZE = 10; // Array size
8     int numbers[ARRAY_SIZE]; // Array with 10 elements
9     int count = 0; // Loop counter variable
10    ifstream inputFile; // Input file stream object
11
12    // Open the file.
13    inputFile.open("TenNumbers.txt");
14
15    // Read the numbers from the file into the array.
16    while (count < ARRAY_SIZE && inputFile >> numbers[count])
17        count++;
18
19    // Close the file.
20    inputFile.close();
21
22    // Display the numbers read:
23    cout << "The numbers are: ";
24    for (count = 0; count < ARRAY_SIZE; count++)
25        cout << numbers[count] << " ";
26    cout << endl;
27    return 0;
28 }
```

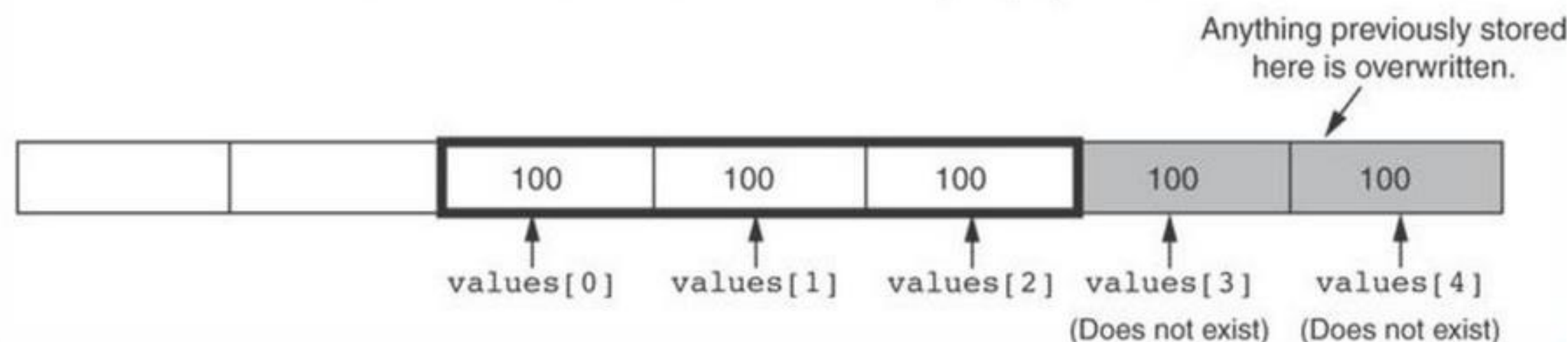
```
1 // This program writes the contents of an array to a file.
2 #include <iostream>
3 #include <fstream>
4 using namespace std;
5
6 int main()
7 {
8     const int ARRAY_SIZE = 10; // Array size
9     int numbers[ARRAY_SIZE]; // Array with 10 elements
10    int count; // Loop counter variable
11    ofstream outputFile; // Output file stream object
12
13    // Store values in the array.
14    for (count = 0; count < ARRAY_SIZE; count++)
15        numbers[count] = count;
16
17    // Open a file for output.
18    outputFile.open("SavedNumbers.txt");
19
20    // Write the array contents to the file.
21    for (count = 0; count < ARRAY_SIZE; count++)
22        outputFile << numbers[count] << endl;
23
24    // Close the file.
25    outputFile.close();
26    // That's it!
27    cout << "The numbers were saved to the file.\n ";
28    return 0;
29 }
```

# Không kiểm tra giới hạn

- C++ không kiểm tra khi vượt quá giới hạn phần tử mảng
- VD: định nghĩa mảng 3 phần tử, nhưng lưu trữ 5 phần tử.

```
9      const int SIZE = 3;  // Constant for the array size
10     int values[SIZE];    // An array of 3 integers
11     int count;           // Loop counter variable
12
13     // Attempt to store five numbers in the three-element array.
14     cout << "I will store 5 numbers in a 3 element array!\n";
15     for (count = 0; count < 5; count++)
16         values[count] = 100;
```

- Kết quả:



- Đoạn code sau bị lỗi:

```
const int SIZE = 100;
int numbers[SIZE];
for (int count = 1; count <= SIZE; count++)
    numbers[count] = 0;
```

# Không kiểm tra giới hạn

- C++ không kiểm tra khi vượt quá giới hạn phần tử

```
4  ?  
4  #include <iostream>  
5  using namespace std;  
6  
7  int main()  
8  {  
9      const int SIZE = 3; // Constant for the array size  
10     int values[SIZE]; // An array of 3 integers  
11     int count; // Loop counter variable  
12  
13     // Attempt to store five numbers in the 3-element array.  
14     cout << "I will store 5 numbers in a 3-element array!\n";  
15     for (count = 0; count < 5; count++)  
16     |     values[count] = 100;  
17  
18     // If the program is still running, display the numbers.  
19     cout << "If you see this message, it means the program\n";  
20     cout << "has not crashed! Here are the numbers:\n";  
21     for (count = 0; count < 5; count++)  
22     |     cout << values[count] << endl;  
23     return 0;  
24 }
```

```
I will store 5 numbers in a 3-element array!  
If you see this message, it means the program  
has not crashed! Here are the numbers:  
100  
100  
100  
100  
100
```

# Kiểm tra



- 6.1. Định nghĩa các mảng sau:
  - A. *empNums*, gồm 100 phần tử kiểu nguyên
  - B. *payRates*, gồm 25 phần tử kiểu float
  - C. *miles*, gồm 14 phần tử kiểu long
  - D. *cityName*, gồm 26 phần tử kiểu string
  - E. *lightYears*, 1000 phần tử kiểu double
- 6.2. Định nghĩa mảng nào dưới đây sai?
  - A. *int readings[-1];*
  - B. *float measurements[4.5];*
  - C. *int size;*
  - D. *string names[size];*
- 6.3. Hãy chỉ ra các chỉ số hợp lệ của mảng gồm 4 phần tử kiểu double?

# Kiểm tra



- 6.4. Sự khác nhau giữa kích thước mảng và chỉ số mảng?
- 6.5. Đoạn code sau có đầu ra
- 6.6. Đoạn chương trình dưới

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int values[5], count;
6      for (count = 0; count < 5; count++)
7          values[count] = count + 1;
8      for (count = 0; count < 5; count++)
9          cout << values[count] << endl;
10     return 0;
11 }
```

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      const int NUM_FISH = 20;
6      int fish[NUM_FISH];
7      // You must finish this program. It should ask how many fish were caught by fishermen 1-20,
8      //and store this data in the array fish.
9      return 0;
10 }
```

# Kiểm tra



- 7.7. Định nghĩa các mảng sau:
  - A. Mảng *ages* gồm 10 phần tử kiểu *int* với giá trị khởi tạo là: 5, 7, 9, 14, 15, 17, 18, 19, 21 và 23.
  - B. Mảng *temps* gồm 7 phần tử kiểu *float* với giá trị khởi tạo là: 14.7, 16.3, 18.43, 21.09, 17.9, 18.76 và 26.7.
  - C. Mảng *alpha* gồm 8 phần tử kiểu *char* với giá trị khởi tạo là: 'J', 'B', 'L', 'A', '\*', '\$', 'H' và 'M'.
- 7.8. Định nghĩa mảng nào đúng/sai? Tại sao?
  - A. `int numbers[10] = {0, 0, 1, 0, 0, 1, 0, 0, 1, 1};`
  - B. `int matrix[5] = {1, 2, 3, 4, 5, 6, 7};`
  - C. `double radii[10] = {3.2, 4.7};`
  - D. `int table[7] = {2, , , 27, , 45, 39};`
  - E. `char codes[] = {'A', 'X', '1', '2', 's'};`
  - F. `int blanks[];`

# Kiểm tra



## Checkpoint

7.1 Define the following arrays:

- A) empNums, a 100-element array of ints
- B) payRates, a 25-element array of floats
- C) miles, a 14-element array of longs
- D) cityName, a 26-element array of string objects
- E) lightYears, a 1,000-element array of doubles

7.2 What's wrong with the following array definitions?

```
int readings[-1];  
float measurements[4.5];  
int size;  
string names[size];
```

7.3 What would the valid subscript values be in a 4-element array of doubles?

7.4 What is the difference between an array's size declarator and a subscript?

7.5 What is "array bounds checking"? Does C++ perform it?

7.6 What is the output of the following code?

```
int values[5], count;  
for (count = 0; count < 5; count++)  
    values[count] = count + 1;  
for (count = 0; count < 5; count++)  
    cout << values[count] << endl;
```

# Một số chú ý

- Khi dùng toán tử ++, -- không nhầm lẫn như sau:
  - `tests[i]++;` //Thêm 1 vào `tests[i]`
  - `tests[i++];` //tăng chỉ số của mảng lên 1
- Không thể copy mảng như sau:
  - `newTests = tests;` **// SAI**
- Có thể copy từng phần tử của mảng như sau:  

```
for(i=0; i<ARRAY_SIZE; i++)  
    newTests[i] = tests[i];
```
- Có thể hiển thị mảng kí tự như sau:  

```
char fName[] = "Phong";  
cout<<fName<<endl;
```
- Để hiển thị nội dung mảng có các kiểu khác như sau:  

```
for(i=0; i<ARRAY_SIZE; i++)  
    cout<<tests[i]<<endl;
```

# Một số tính toán thường gặp

- Tính trung bình:

```
5  const int NUM_SCORES = 10;
6  double scores[NUM_SCORES];
7  double total = 0; // Initialize accumulator
8  double average; // Will hold the average
9  for (int count = 0; count < NUM_SCORES; count++)
10     total += scores[count];
11  average = total / NUM_SCORES;
```

```
5  const int SIZE = 50;
6  int numbers[SIZE];
7  int count;
8  int highest;
9  highest = numbers[0];
10 for (count = 1; count < SIZE; count++)
11 {
12     if (numbers[count] > highest)
13         highest = numbers[count];
14 }
```

- Tìm số lớn nhất:

```
5  const int SIZE = 50;
6  int numbers[SIZE];
7  int count;
8  int lowest;
9  lowest = numbers[0];
10 for (count = 1; count < SIZE; count++)
11 {
12     if (numbers[count] < lowest)
13         lowest = numbers[count];
14 }
```

- Tìm số nhỏ nhất:

# Một số tính toán thường gặp

- Thêm các phần tử vào mảng:
  - Nhiều khi không biết số phần tử mà mảng sẽ nắm giữ:
    - Tạo ra 1 mảng đủ lớn để lưu
    - Sử dụng 1 biến đếm để lưu số lượng phần tử đang nắm giữ

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      const int SIZE = 100;
6      int numbers[SIZE];
7      int num, count;
8      cout << "Enter a number or -1 to quit: ";
9      cin >> num;
10     while (num != -1 && count < SIZE)
11     {
12         count++;
13         numbers[count - 1] = num;
14         cout << "Enter a number or -1 to quit: ";
15         cin >> num;
16     }
17 }
```

```
17     for (int index = 0; index < count; index++)
18     {
19         cout << numbers[index] << endl;
20     }
```

# Một số tính toán thường gặp

- So sánh mảng:

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      const int SIZE = 5;
6      int firstArray[SIZE] = { 5, 10, 15, 20, 25 };
7      int secondArray[SIZE] = { 5, 10, 15, 20, 25 };
8      bool arraysEqual = true; // Flag variable
9      int count = 0; // Loop counter variable
10     // Determine whether the elements contain the same data.
11     while (arraysEqual && count < SIZE)
12     {
13         if (firstArray[count] != secondArray[count])
14             arraysEqual = false;
15         count++;
16     }
17     if (arraysEqual)
18         cout << "The arrays are equal.\n";
19     else
20         cout << "The arrays are not equal.\n";
21 }
```

# Một số tính toán thường gặp

- Mảng song song:
  - Gồm 1 hoặc **nhiều mảng có liên quan về dữ liệu với nhau.**
  - **Một chỉ số dùng cho các mảng:** các phần tử cùng chỉ số có liên quan đến nhau.
  - Các mảng có thể khác nhau về kiểu.

```
const int SIZE = 5;    // Array size
int id[SIZE];          // student ID
double average[SIZE]; // course average
char grade[SIZE];      // course grade
...
for(int i = 0; i < SIZE; i++)
{
    cout << "Student ID: " << id[i]
          << " average: " << average[i]
          << " grade: " << grade[i]
          << endl;
}
```

# Một số tính toán thường gặp

## Program 7-12

```

1 // This program uses two parallel arrays: one for hours
2 // worked and one for pay rate.
3 #include <iostream>
4 #include <iomanip>
5 using namespace std;
6
7 int main()
8 {
9     const int NUM_EMPLOYEES = 5;    // Number of employees
10    int hours[NUM_EMPLOYEES];        // Holds hours worked
11    double payRate[NUM_EMPLOYEES];   // Holds pay rates
12
13    // Input the hours worked and the hourly pay rate.
14    cout << "Enter the hours worked by " << NUM_EMPLOYEES
15         << " employees and their\n"
16         << "hourly pay rates.\n";
17    for (int index = 0; index < NUM_EMPLOYEES; index++)
18    {
19        cout << "Hours worked by employee #" << (index+1) << ": ";
20        cin >> hours[index];
21        cout << "Hourly pay rate for employee #" << (index+1) << ": ";
22        cin >> payRate[index];
23    }
24
25    // Display each employee's gross pay.
26    cout << "Here is the gross pay for each employee:\n";
27    cout << fixed << showpoint << setprecision(2);
28    for (int index = 0; index < NUM_EMPLOYEES; index++)
29    {
30        double grossPay = hours[index] * payRate[index];
31        cout << "Employee #" << (index + 1);
32        cout << ": $" << grossPay << endl;
33    }
34    return 0;
35 }

```

## Program Output with Example Input Shown in Bold

Enter the hours worked by 5 employees and their hourly pay rates.

Hours worked by employee #1: **10** [Enter]

Hourly pay rate for employee #1: **9.75** [Enter]

Hours worked by employee #2: **15** [Enter]

Hourly pay rate for employee #2: **8.62** [Enter]

Hours worked by employee #3: **20** [Enter]

Hourly pay rate for employee #3: **10.50** [Enter]

Hours worked by employee #4: **40** [Enter]

Hourly pay rate for employee #4: **18.75** [Enter]

Hours worked by employee #5: **40** [Enter]

Hourly pay rate for employee #5: **15.65** [Enter]

Here is the gross pay for each employee:

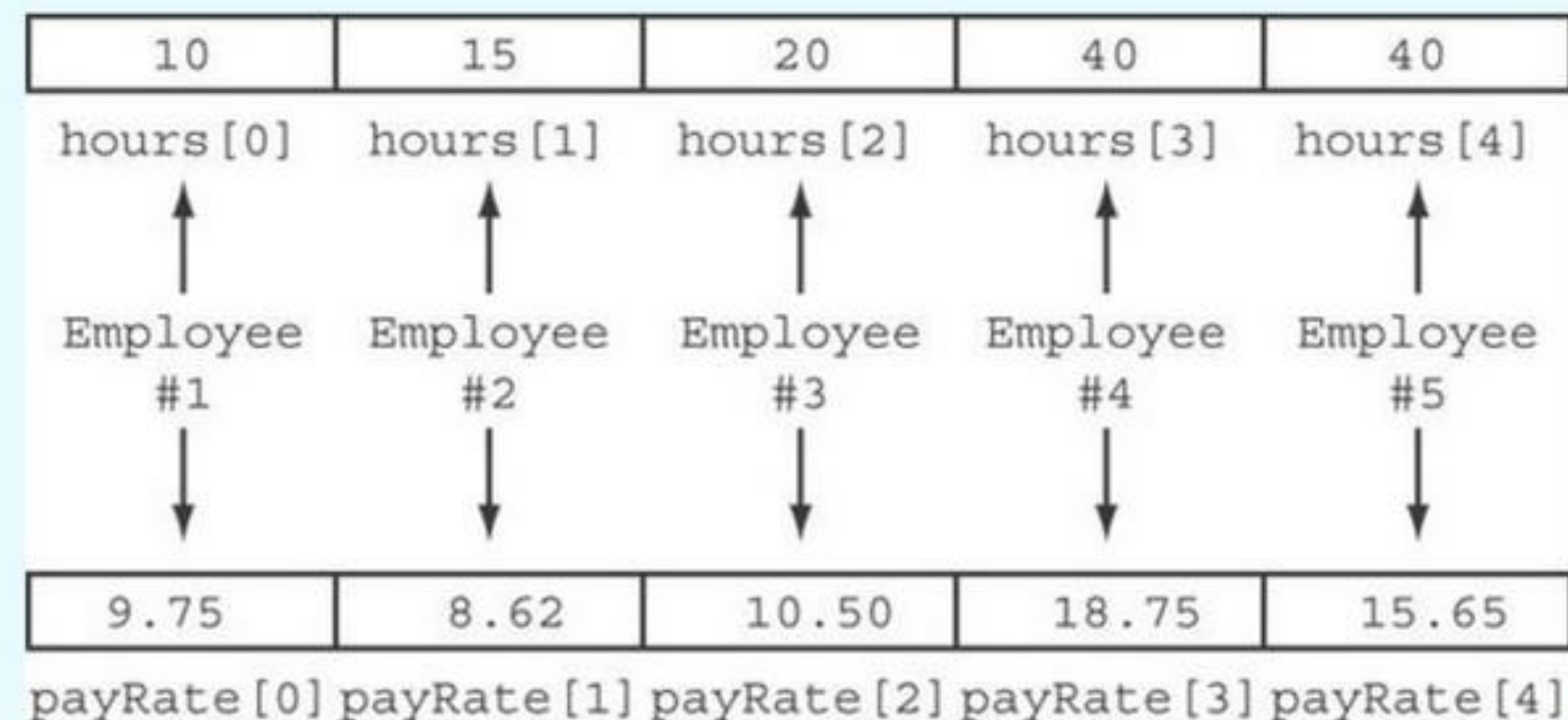
Employee #1: \$97.50

Employee #2: \$129.30

Employee #3: \$210.00

Employee #4: \$750.00

Employee #5: \$626.00



# Kiểm tra



- 6.10. Cho mảng `int values[] = {2, 6, 10, 14}`; Các câu dưới đây hiển thị gì?
  - A. `cout << values[2];`
  - B. `cout << ++values[0];`
  - C. `cout << values[1]++;`
  - D. `x = 2;`  
`cout << values[++x];`
- 6.11. Cho mảng `int nums[5] = {1, 2, 3}`; Câu lệnh sau hiển thị gì?
  - `cout << nums[3];`

# Kiểm tra



- *6.12. Các đoạn code sau có đầu ra là gì?*

What is the output of the following code? (You may need to use a calculator.)

```
double balance[5] = {100.0, 250.0, 325.0, 500.0, 1100.0};  
const double INTRATE = 0.1;
```

```
cout << fixed << showpoint << setprecision(2);  
for (int count = 0; count < 5; count++)  
    cout << (balance[count] * INTRATE) << endl;
```

What is the output of the following code? (You may need to use a calculator.)

```
const int SIZE = 5;  
int time[SIZE] = {1, 2, 3, 4, 5},  
    speed[SIZE] = {18, 4, 27, 52, 100},  
    dist[SIZE];  
  
for (int count = 0; count < SIZE; count++)  
    dist[count] = time[count] * speed[count];  
for (int count = 0; count < SIZE; count++)  
{  
    cout << time[count] << " ";  
    cout << speed[count] << " ";  
    cout << dist[count] << endl;  
}
```

# Mảng là tham số của hàm

- Để truyền 1 mảng cho hàm thì sử dụng tên mảng
  - `showScores(tests);`
- Để định nghĩa hàm có mảng là tham số, sử dụng `[]` trống
  - `void showScores(int []);`//hàm nguyên mẫu
  - `void showScores(int tests[]);`//định nghĩa hàm
- Khi truyền 1 mảng cho hàm, **thường truyền cả kích thước mảng** để biết bao nhiêu phần tử cần xử lý.
  - `showScores(tests, ARRAY_SIZE);`
- Kích thước mảng cũng phải được thể hiện trong nguyên mẫu và định nghĩa hàm.
  - `void showScores(int [], int);`//hàm nguyên mẫu
  - `void showScores(int tests[], int size);`//định nghĩa hàm

# Mảng là tham số của hàm

## Program 7-14

```
1 // This program demonstrates an array being passed to a function.
2 #include <iostream>
3 using namespace std;
4
5 void showValues(int [], int); // Function prototype
6
7 int main()
8 {
9     const int ARRAY_SIZE = 8;
10    int numbers[ARRAY_SIZE] = {5, 10, 15, 20, 25, 30, 35, 40};
11
12    showValues(numbers, ARRAY_SIZE);
13    return 0;
14 }
15
16 //*****
17 // Definition of function showValue.
18 // This function accepts an array of integers and
19 // the array's size as its arguments. The contents
20 // of the array are displayed.
21 //*****
22
23 void showValues(int nums[], int size)
24 {
25     for (int index = 0; index < size; index++)
26         cout << nums[index] << " ";
27     cout << endl;
28 }
```

### Program Output

5 10 15 20 25 30 35 40

# Sự thay đổi mảng trong hàm

- Tên mảng trong hàm giống như 1 biến **tham chiếu** – sự thay đổi mảng trong hàm sẽ liên quan đến mảng thực tế khi gọi hàm.
- Cần phải thận trọng không phải mảng bị thay đổi bởi hàm

```
1  #include <iostream>
2  using namespace std;
3  void ReferenceArray(int [],int);
4  int main()
5  {
6      const int SIZE = 3; // Constant for the array size
7      int values[SIZE]={1,2,3}; // An array of 3 integers
8      cout<<"Trước khi gọi mảng:";
9      cout<<values[0]<<" "<<values[1]<<" "<<values[2]<<endl;
10     //Gọi mảng
11     ReferenceArray(values,SIZE);
12     cout<<"Sau khi gọi mảng:";
13     cout<<values[0]<<" "<<values[1]<<" "<<values[2]<<endl;
14
15     return 0;
16 }
17 void ReferenceArray(int tests[],int size)
18 {
19     tests[0]=3;
20     tests[1]=2;
21     tests[2]=1;
22 }
```

```
Trước khi gọi mảng:1 2 3
Sau khi gọi mảng:3 2 1
```

# Mảng 2 chiều

- Có thể định nghĩa 1 mảng cho nhiều tập hợp dữ liệu
- Giống như 1 bảng trong bảng tính
- Sử dụng 2 kích thước để khai báo
  - `const int ROWS = 4, COLS = 3;`
  - `int exams[ROWS][COLS];`
- Khai báo đầu là số dòng, khai báo thứ hai là số cột

r o w s	columns		
	<code>exams[0][0]</code>	<code>exams[0][1]</code>	<code>exams[0][2]</code>
	<code>exams[1][0]</code>	<code>exams[1][1]</code>	<code>exams[1][2]</code>
	<code>exams[2][0]</code>	<code>exams[2][1]</code>	<code>exams[2][2]</code>
	<code>exams[3][0]</code>	<code>exams[3][1]</code>	<code>exams[3][2]</code>

- Sử dụng chỉ số để truy cập mảng:
  - `exams[2][2] = 86;`

# Mảng 2 chiều

## Program 7-18

```
1 // This program demonstrates a two-dimensional array.
2 #include <iostream>
3 #include <iomanip>
4 using namespace std;
5
6 int main()
7 {
8     const int NUM_DIVS = 3;           // Number of divisions
9     const int NUM_QTRS = 4;           // Number of quarters
10    double sales[NUM_DIVS][NUM_QTRS]; // Array with 3 rows and 4 columns.
11    double totalSales = 0;             // To hold the total sales.
12    int div, qtr;                      // Loop counters.
13
14    cout << "This program will calculate the total sales of\n";
15    cout << "all the company's divisions.\n";
16    cout << "Enter the following sales information:\n\n";
17
18    // Nested loops to fill the array with quarterly
19    // sales figures for each division.
20    for (div = 0; div < NUM_DIVS; div++)
21    {
22        for (qtr = 0; qtr < NUM_QTRS; qtr++)
23        {
24            cout << "Division " << (div + 1);
25            cout << ", Quarter " << (qtr + 1) << ": $";
26            cin >> sales[div][qtr];
27        }
28        cout << endl; // Print blank line.
29    }
30
31    // Nested loops used to add all the elements.
32    for (div = 0; div < NUM_DIVS; div++)
33    {
34        for (qtr = 0; qtr < NUM_QTRS; qtr++)
35            totalSales += sales[div][qtr];
36    }
37
38    cout << fixed << showpoint << setprecision(2);
39    cout << "The total sales for the company are: $";
40    cout << totalSales << endl;
41    return 0;
42 }
```

## Program Output with Example Input Shown in Bold

This program will calculate the total sales of all the company's divisions.  
Enter the following sales data:

Division 1, Quarter 1: \$**31569.45** [Enter]  
Division 1, Quarter 2: \$**29654.23** [Enter]  
Division 1, Quarter 3: \$**32982.54** [Enter]  
Division 1, Quarter 4: \$**39651.21** [Enter]

Division 2, Quarter 1: \$**56321.02** [Enter]  
Division 2, Quarter 2: \$**54128.63** [Enter]  
Division 2, Quarter 3: \$**41235.85** [Enter]  
Division 2, Quarter 4: \$**54652.33** [Enter]

Division 3, Quarter 1: \$**29654.35** [Enter]  
Division 3, Quarter 2: \$**28963.32** [Enter]  
Division 3, Quarter 3: \$**25353.55** [Enter]  
Division 3, Quarter 4: \$**32615.88** [Enter]

The total sales for the company are: \$456782.34

# Khởi tạo mảng 2 chiều

- Mảng 2 chiều được khởi tạo dòng đến dòng
  - `const int ROWS = 4, COLS = 3;`
  - `int exams[ROWS][COLS] = {{84,78}, {92,97}};`

84	78
92	97

- Có thể bỏ qua trong {}, các phần tử trong mảng không khởi tạo sẽ nhận giá trị 0 hoặc NULL.

# Sử dụng mảng 2 chiều là tham số, đối số

- Sử dụng tên mảng như đối số trong lời gọi hàm
  - `getExams(exams,2);`
- VD: Sử dụng []-rỗng và khai báo kích thước cột trong hàm nguyên mẫu, tiêu đề hàm
  - `const int COLS = 2;`
  - `void getExams(int[][COLS], int);` //Hàm nguyên mẫu
  - `void getExams(int exam[][COLS], int rows);` //Tiêu đề hàm

# Ví dụ: Hàm hiển thị mảng 2 chiều - showArray

```
15  int table1[TBL1_ROWS][COLS] = {{1, 2, 3, 4},
16                                   {5, 6, 7, 8},
17                                   {9, 10, 11, 12}};
18  int table2[TBL2_ROWS][COLS] = {{10, 20, 30, 40},
19                                   {50, 60, 70, 80},
20                                   {90, 100, 110, 120},
21                                   {130, 140, 150, 160}};
22
23  cout << "The contents of table1 are:\n";
24  showArray(table1, TBL1_ROWS);
25  cout << "The contents of table2 are:\n";
26  showArray(table2, TBL2_ROWS);
30  //*****
31  // Function Definition for showArray *
32  // The first argument is a two-dimensional int array with COLS *
33  // columns. The second argument, rows, specifies the number of *
34  // rows in the array. The function displays the array's contents. *
35  //*****
36
37  void showArray(int array[][COLS], int rows)
38  {
39      for (int x = 0; x < rows; x++)
40      {
41          for (int y = 0; y < COLS; y++)
42          {
43              cout << setw(4) << array[x][y] << " ";
44          }
45          cout << endl;
46      }
47  }
```

# Một số ví dụ trong mảng 2 chiều

## Tổng tất cả các phần tử trong mảng 2 chiều

```
const int NUM_ROWS = 5; // Number of rows
const int NUM_COLS = 5; // Number of columns
int total = 0;           // Accumulator
int numbers[NUM_ROWS][NUM_COLS] =
    {{2, 7, 9, 6, 4},
     {6, 1, 8, 9, 4},
     {4, 3, 7, 2, 9},
     {9, 9, 0, 3, 1},
     {6, 2, 7, 4, 1}};
```

```
// Sum the array elements.
for (int row = 0; row < NUM_ROWS; row++)
{
    for (int col = 0; col < NUM_COLS; col++)
        total += numbers[row][col];
}

// Display the sum.
cout << "The total is " << total << endl;
```

# Một số ví dụ trong mảng 2 chiều

Tổng tất cả các phần tử theo dòng và hiển thị trung bình theo dòng

```
const int NUM_STUDENTS = 3;
const int NUM_SCORES = 5;
double total;    // Accumulator
double average; // To hold average scores
double scores[NUM_STUDENTS][NUM_SCORES] =
    {{88, 97, 79, 86, 94},
     {86, 91, 78, 79, 84},
     {82, 73, 77, 82, 89}};
```

```
// Get each student's average score.
for (int row = 0; row < NUM_STUDENTS; row++)
{
    // Set the accumulator.
    total = 0;
    // Sum a row.
    for (int col = 0; col < NUM_SCORES; col++)
        total += scores[row][col];
    // Get the average
    average = total / NUM_SCORES;
    // Display the average.
    cout << "Score average for student "
          << (row + 1) << " is " << average << endl;
}
```

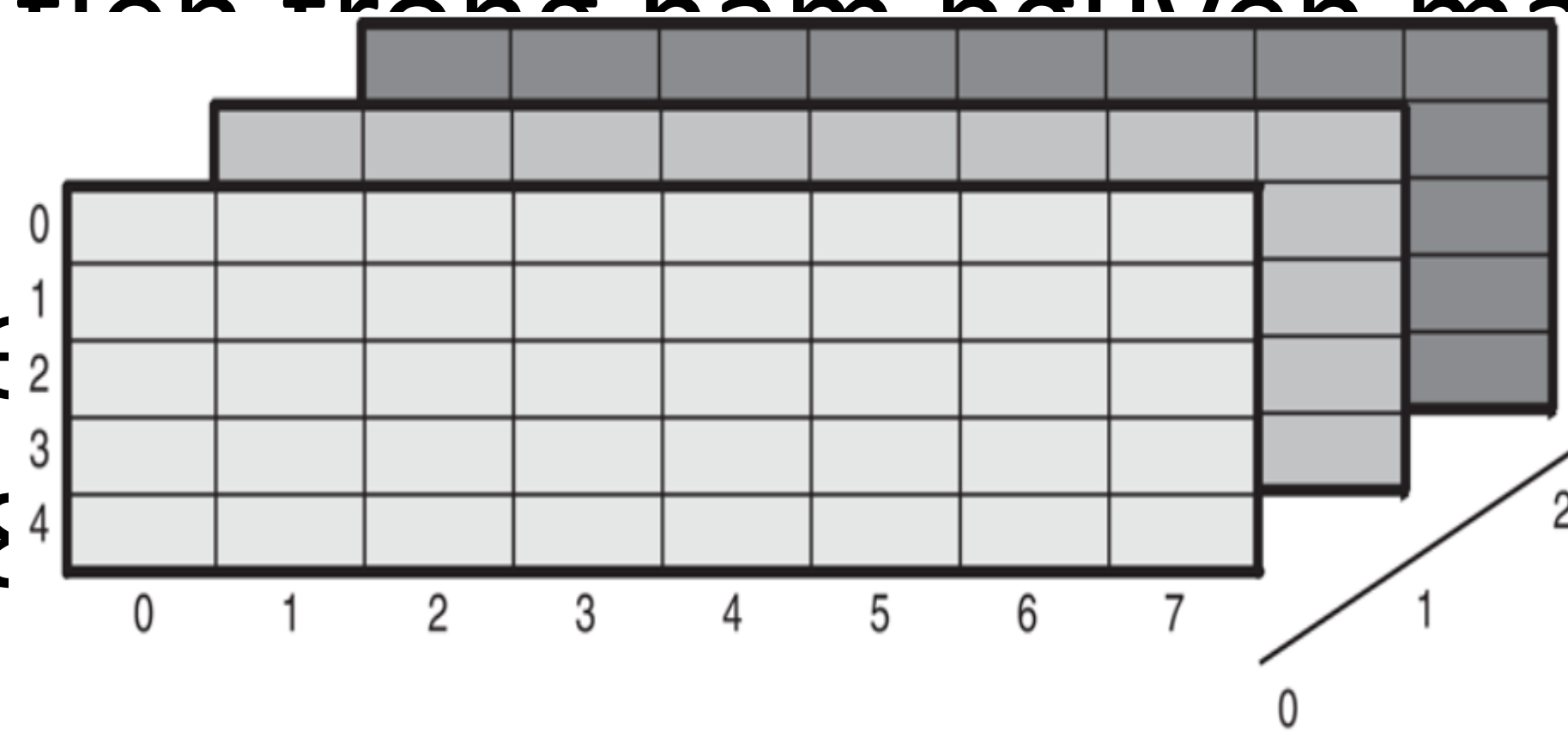
Tổng tất cả các phần tử theo cột và hiển thị trung bình theo cột

```
const int NUM_STUDENTS = 3;
const int NUM_SCORES = 5;
double total;    // Accumulator
double average; // To hold average scores
double scores[NUM_STUDENTS][NUM_SCORES] =
    {{88, 97, 79, 86, 94},
     {86, 91, 78, 79, 84},
     {82, 73, 77, 82, 89}};
```

```
// Get the class average for each score.
for (int col = 0; col < NUM_SCORES; col++)
{
    // Reset the accumulator.
    total = 0;
    // Sum a column
    for (int row = 0; row < NUM_STUDENTS; row++)
        total += scores[row][col];
    // Get the average
    average = total / NUM_STUDENTS;
    // Display the class average.
    cout << "Class average for test " << (col + 1)
          << " is " << average << endl;
}
```

# Mảng 3 chiều hoặc nhiều hơn

- Có thể định nghĩa với một vài chiều như sau:
    - `short rectSolid[2][3][5];`
    - `double timeGrid[3][4][3][4];`
  - C++ không giới hạn số chiều
  - Khi sử dụng mảng làm tham số cho hàm thì cần chỉ kích thước, trừ chiều đầu tiên trong hàm nguyên mẫu và tiêu đề hàm.
    - `void getRectSolid(short [] [3] [5]);`
  - Ví dụ: `double seats[3][5][8];`
- 



# Kiểm tra



- 6.19. Định nghĩa mảng 2 chiều kiểu `int` có tên là `grade`. Gồm 30 dòng và 10 cột.
- 6.20. Có bao nhiêu phần tử trong mảng `double sales[6][4];`?
- 6.21. Viết câu lệnh gán `56893.12` cho cột và dòng đầu tiên cho mảng `double sales[6][4];`
- 6.22. Viết câu lệnh hiển thị phần tử cuối cùng của mảng `double sales[6][4];`

# Giới thiệu vector

- Một kiểu dữ liệu định nghĩa trong STL (Standard Template Library)
- Có thể nắm giữ các giá trị của bất kỳ loại nào:
  - `vector<int> scores;`
- Tự động thêm không gian khi cần thiết – Không cần xác định kích thước khi định nghĩa.
- Có thể truy cập các phần tử bằng `[]` – giống mảng.
- Cần thêm `#include<vector>`
- Một số ví dụ:
  - `vector<int> scores; // Khai báo vector kiểu int`
  - `vector<int> scores(30); // Khai báo vector kiểu int, khởi tạo kích thước là 30`
  - `vector<int> scores(30,0); // Khai báo vector kiểu int, khởi tạo kích thước là 30 và tất cả các giá trị là 0.`
  - `vector<int> finals(scores); // Khai báo và khởi tạo vector finals như vector scores`

# Thêm/Loại bỏ 1 phần tử trong vector

- Thêm 1 phần tử vào 1 vector:
  - `scores.push_back(75);`//Thêm 1 phần tử có giá trị 75 vào vector scores.
- Để biết kích thước của 1 vector:
  - `int kichthuoc = scores.size();`
- Loại bỏ 1 phần tử cuối cùng trong vector:
  - `scores.pop_back();`//Loại bỏ phần tử cuối cùng trong vector scores
- Loại bỏ tất cả phần tử trong vector:
  - `scores.clear();`
- Để xác định một vector là rỗng:
  - `while(!scores.empty())`

# HỎI ĐÁP



# Bài tập

- **Bài tập chương 6:**
  - Chapter6\_Programming Challenges.pdf



**Trân trọng cảm ơn!**

**ĐẬU HẢI PHONG**

*Giảng viên*

*dauhaiphong@dainam.edu.vn*

*0912441435*