http://Tuhoc.cc

LẬP TRÌNH C++ Dễ hiểu

LESSON 21.1

Mảng 1 chiều C++

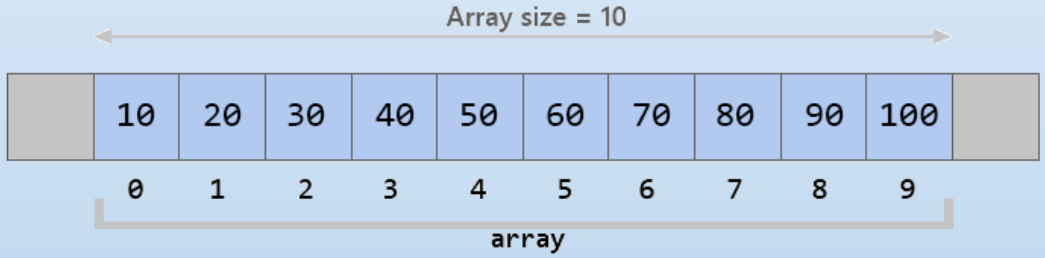
Part 1

@galailaptrinh

1 Mảng – array C++

❑ 1. Khái niệm : *Mảng là tập hợp các biến có cùng kiểu dữ liệu*

- ✓ *Mảng có kích thước cố định không thể thay đổi*
- ✓ *Mảng có index bắt đầu từ 0*



❑ 2 . Tại sao phải dùng mảng:

Ví dụ : Chúng ta có khoảng 50 điểm của học sinh cần lưu, nếu không dùng mảng thì chúng ta phải khai báo 50 biến float

=> Gom nhóm các đối tượng có chung tính chất lại với nhau

=> Giúp code gọn gàng hơn.

1 Mảng – array C++

❑ 3. Khai báo / khởi tạo mảng :

- ✓ *Khai báo:* Kiểu_dữ_liệu <tên mảng> [];
- ✓ *Khởi tạo:* Kiểu_dữ_liệu <tên mảng> [] = { gt1,gt2,gt3...};

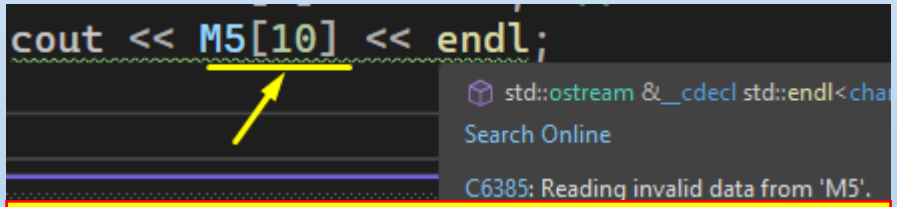
```
//1. khai báo mảng
int M[3]; // mảng nguyên M có 3 phần tử
string M2[7]; //mảng string M2 có 7 phần tử

//2. Khởi tạo (có gán giá trị)
int M3[] = { 1,3,5,7,9 };
string M4[] = { "mot", "hai", "ba", "bon" };
```

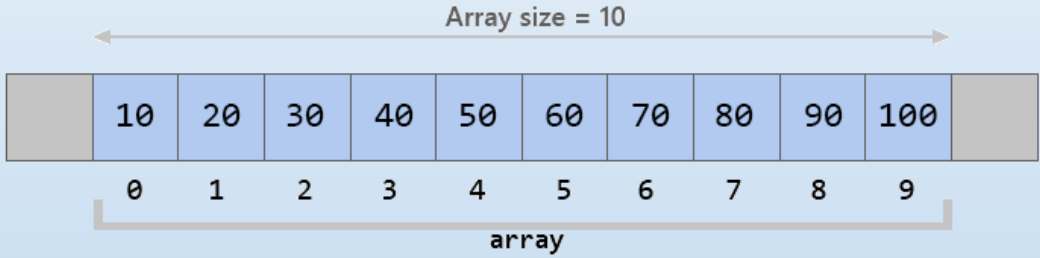
1 Mảng – array C++

❑ 4. Truy xuất phần tử của mảng :

```
//3. Truy xuất các phần tử theo vị trí index
int M5[] = { 10,20,30,40,50,60,70,80,90,100 };
cout << M5[0] << endl;   // 10
cout << M5[1] << endl;   // 20
cout << M5[2] << endl;   // 30
cout << M5[9] << endl;   // 100
```



Lỗi khi truy xuất vị trí index không tồn tại



❑5. length : trả về số phần tử của mảng (chiều dài mảng, bắt đầu từ 1)

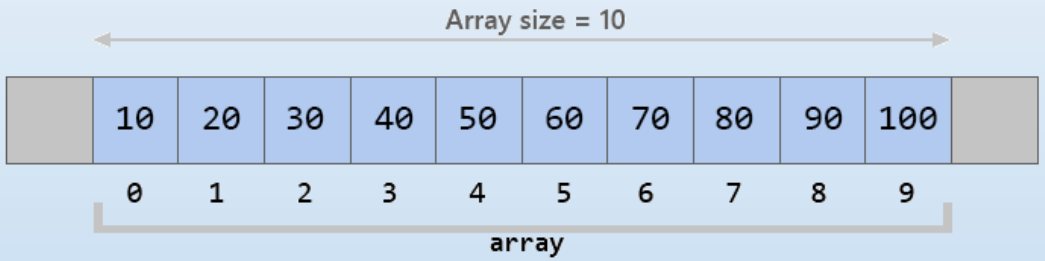
```
//4. kiểm tra chiều dài mảng
int M6[] = { 1, 2, 3, 4, 5 };
int arraySize = size(M6);
cout << arraySize << endl; // 5 p tử
```

1 Mảng – array C++

❑ 6. Duyệt mảng :

```
//5. Duyệt mảng, dùng để xuất các pt trong mảng
cout << "Mang M7 la: ";
int M7[] = { 10,20,30,40,50,60,70,80,90,100 };
for (int pt : M7) {
    cout << pt << " ";
}
cout << endl;

//6. duyệt mảng theo vị trí index
for (int i = 0; i < size(M7);i++) {
    //in ra vị trí index
    cout << i << endl;
    //in ra giá trị tại vị trí i
    cout << M7[i] << endl;
}
```



1 Mảng – array C++

❑ 7. Thay đổi giá trị cho mảng :

```
//cách 1 : gọi trực tiếp và gán giá trị
int M8[] = { 10,20,30,40,50,60,70,80,90,100 };
cout << M8[0] << endl; // xuất M8 tại index 0 => 10
M8[0] = 88; // thay giá trị tại index 0
cout << M8[0] << endl;  // kqua =>> 88
```

```
//cách 2 : dùng for để update toàn bộ các phần tử
int M9[] = { 10,20,30,40,50,60,70,80,90,100 };
for (int i = 0; i < size(M9);i++) {
    M9[i] += 2;
}
//xuất mảng sau khi thay đổi
cout << "Mang M9 sau doi la: ";
for (int pt : M9) {
    cout << pt << " ";
}
```

```
Mang M9 sau doi la: 12 22 32 42 52 62 72 82 92 102
```

http://Tuhoc.cc

LẬP TRÌNH C++ Dễ hiểu

LESSON 21.2

Mảng 1 chiều C++

Part 2

@galailaptrinh

1 Mảng – array C++

❑ **8. Ví dụ :** *Viết chương trình tạo 1 mảng 1 chiều gồm các phần tử là số nguyên có 4 phần tử, các phần tử do người dùng nhập từ bàn phím*

```
int mang[4];;

for (int i = 0; i < size(mang); i++) {
    cout << "mang[" << i << "]= " <<endl;
    cin >> mang[i];

}

cout << "Mang ban vua nhap la: ";
for (int i = 0; i < size(mang); i++) {
    cout << mang[i] << " ";
}
```

```
mang[0]=
5
mang[1]=
4
mang[2]=
7
mang[3]=
8
Mang ban vua nhap la: 5 4 7 8
```

1 Mảng – array C++

9. Sắp xếp mảng :

```cpp
//sử dụng thư viện #include <algorithm>
int mang2[5] = { 15, 2, 7, 6, 4 };

//sort(begin(arr), end(arr));
//begin(arr) trả về con trỏ đến phần tử đầu tiên của mảng,
//end(arr) trả về con trỏ đến vị trí sau phần tử cuối cùng của mảng.
sort(mang2, mang2 + size(mang2));
cout << "Mang sau sap xep tang : ";
for (int i = 0; i < 5; i++) {
    cout << mang2[i] << " ";
}
```

```
Mang sau sap xep tang : 2 4 6 7 15
```

10. Đảo ngược mảng :

```cpp
//10. đảo ngược mảng
reverse(mang2, mang2 + size(mang2));
cout << "\nMang sau dao nguoc: ";
for (int i = 0; i < 5; i++) {
    cout << mang2[i] << " ";
}
```

```
Mang sau dao nguoc: 15 7 6 4 2
```

1 Mảng – array C++

❑ 11. Tạo mảng với các phần tử ngẫu nhiên :

```cpp
#include <iostream>
#include <random>
using namespace std;

int main() {
    random_device rd;
    mt19937 gen(rd());
    uniform_int_distribution<> dis(0, 99); // set khoảng ngẫu nhiên


    int mang3[10]; // mảng có 10 phần tử
    for (int i = 0; i < size(mang3); i++) {
        mang3[i] = dis(gen); // gán số ngẫu nhiên với dis đã set
    }
    // xuất mảng
    for (int pt : mang3) {
        cout << pt << " ";
    }
}
```

1 Mảng – array C++

❑ Mersenne Twister

3.3 Mersenne Twister

Mersenne Twister là một thuật toán PRNG được *Makoto Matsumoto* và *Takuji Nishimura* phát triển vào năm 1997. Đây là một thuật toán thực sự tuyệt vời. Rất nhanh và tạo ra được dãy số với chất lượng ngẫu nhiên rất cao.

Mersenne Twister được sử dụng như là built-in PRNG cho *Python, Ruby, PHP và R*.

Cái tên Mersenne Twister được chọn vì chu kì của số ngẫu nhiên tạo ra bởi thuật toán này luôn là một số nguyên tố Mersenne

> FYI: Số nguyên tố Mersenne có dạng $M_n = 2^n - 1$, ví dụ 31.

Số nguyên tố Mersenne thường được sử dụng trong thuật toán sinh random là $2^{19937} - 1$, đó cũng là nguồn gốc của cái tên *MT19937* - standard implement của Mersene Twister. Kết quả đưa ra số tự nhiên 32 bits.

Ưu điểm:

- Đưa ra được dải số rất lớn $2^{19937} - 1$
- Pass rất nhiều các bài kiểm tra về tính ngẫu nhiên, có thể nói MT là một thuật toán vô cùng tốt.

2 Bài tập C++ 21

✓ *1. Viết chương trình tạo 1 mảng 1 chiều gồm các phần tử là số nguyên, có 7 phần tử ngẫu nhiên.*

✓*2. Xuất các giá trị trong mảng*

✓*3. Đảo ngược mảng, và xuất mảng sau khi đảo ngược*

✓*4. Sắp xếp mảng tăng dần*

✓*5. Tính tổng các phần tử trong mảng*

✓*6. Cho người dùng nhập 1 số bất kỳ, kiểm tra số đó có tồn tại trong mảng hay không, nếu có thì có bao nhiêu số? ?*

```
32 50 36 81 9 86 81 57 45 29
Mang sau dao nguoc la:
29 45 57 81 86 9 81 36 50 32
Mang sau sx tang la:
9 29 32 36 45 50 57 81 81 86
Tong cac pt trong mang= 506
Moi nhap so: 81
co 2 so 81 trong mang
```

http://Tuhoc.cc

LẬP TRÌNH C++ Dễ hiểu

LESSON 21.3

Giải bài tập C++ 21

Mảng 1 chiều C++

@galailaptrinh

5 Giải Bài tập 21

```
/*
1. Viết chương trình tạo 1 mảng 1 chiều gồm các phần tử là số nguyên,
có 7 phần tử ngẫu nhiên.
2. Xuất các giá trị trong mảng
3. Đảo ngược mảng, và xuất mảng sau khi đảo ngược
4. Sắp xếp mảng tăng dần
5. Tính tổng các phần tử trong mảng
6. Cho người dùng nhập 1 số bất kỳ,
kiểm tra số đó có tồn tại trong mảng hay không, nếu có thì có bao nhiêu số? ?
*/
```

```
#include <iostream>
#include <random>
#include <algorithm>
using namespace std;

int main()
{
    random_device rd;
    mt19937 gen(rd());
    uniform_int_distribution<> dis(0, 99);
}
```

5 Giải Bài tập 21

```cpp
int mang[10]; // mảng có 10 phần tử
for (int i = 0; i < size(mang); i++) {
    // gán số ngẫu nhiên với dis đã set
    mang[i] = dis(gen);
}
// 2. xuất mảng
for (int pt : mang) {
    cout << pt << " ";
}
cout << endl;
//3. đảo ngược mảng
reverse(mang, mang + size(mang));
//xuất mảng
cout << "Mang sau dao nguoc là: "<<endl;
for (int pt : mang) {
    cout << pt << " ";
}
cout << endl;
```

```cpp
//4.sx tăng dần
sort(mang, mang + size(mang));
//xuất mảng
cout << "Mang sau sx tang là: " << endl;
for (int pt : mang) {
    cout << pt << " ";
}
cout << endl;

//5. tính tổng các pt trong mảng
int tong = 0;
for (int i = 0; i < size(mang); i++) {
    tong += mang[i];
}
cout << "Tong cac pt trong mang= "<< tong << endl;

//6. nhập vào 1 số bất kỳ,
//kiểm tra xem có tồn tại trong mảng hay không,
//nếu có thì có bao nhiêu số?
int so;
cout << "Moi nhap so: ";
cin >> so;
string vitri = "";
int dem = 0;
for (int i = 0; i < size(mang); i++) {
    if (so == mang[i])
        dem++;
}
if(dem==0)
    cout << "khong co so " << so << " trong mang";
else
    cout << "co "<<dem <<" so " << so << " trong mang";
```