

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Yêu cầu công việc:

Lập một chương trình tính so sánh giữa 2 số tự nhiên (a,b).

Tính diện tích một hình vuông có cạnh = a.

Tính tiền điện tiêu thụ với số điện tiêu thụ là a.

STT	Mức sử dụng của một hộ trong một tháng	Giá bán điện (đồng/kWh)
1	Bậc 1: Cho kWh từ 0-50	1.388
2	Bậc 2: Cho kWh từ 51-100	1.433
3	Bậc 3: Cho kWh từ 101-200	1.660
4	Bậc 4: Cho kWh từ 201-300	2.082
5	Bậc 5: Cho kWh từ 301-400	2.324
6	Bậc 6: Cho kWh từ 401 trở lên	2.399

Theo Quyết định số 4887/QĐ-BCT ngày 30/5/2014

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Dữ liệu kiểu mảng (Array)

Định nghĩa. Mảng là dãy các phần tử (biến) có cùng chung một kiểu dữ liệu được tổ chức liên tục nhau trong bộ nhớ.

Khai báo mảng một chiều:

Tên-kiểu Tên-mảng [số lượng phần tử];

Ví dụ: Khai báo mảng một chiều

int A[10]; //khai báo mảng A kiểu int gồm 10 phần tử A[0], A[1], ..., A[9]

float X[10]; //khai báo mảng X kiểu foat gồm 10 phần tử X[0], X[1], ..., X[9]

int A[] = { 9, 7, 12, 8, 6, 5 };// vừa khai báo vừa khởi đầu cho mảng

Khai báo mảng nhiều chiều:

Tên-kiểu Tên-mảng [chiều 1][chiều 2]...[chiều k];

Ví dụ: Khai báo mảng nhiều chiều (2 chiều):

int A[3][3]; //Khai báo ma trận vuông cấp 3x3 gồm 9 phần tử

// Các phần tử A[0][0],..., A[0,2], ..., A[2][0], A[2][1], A[2][2]

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Ví dụ: Khai báo và khởi đầu cho mảng nhiều chiều (2 chiều):

```
int A[3][3] = { //Khai báo mảng gồm 9 phần tử
    { 1, 2, 3}, // khởi đầu cho hàng 0
    { 4, 5, 6}, // khởi đầu cho hàng 1
    { 7, 8, 9} // khởi đầu cho hàng 2. Chú ý không có dấu ','.
};
```

Hoặc ta có thể vừa khai báo và khởi đầu thế này cũng được:

```
int A[3][3] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
```

Truy cập phần tử của mảng: thông qua chỉ số phần tử trong mảng.

Ví dụ:

```
int A[10]; //Khai báo mảng gồm 10 phần tử.
```

```
A[5] = 12; //Khởi tạo cho A[5] giá trị là 12.
```

```
int B[3][3]; // Khai báo mảng 2 chiều gồm 9 phần tử
```

```
B[1][2] = 9; //Khởi tạo cho B[1][2] giá trị là 9.
```


Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Ví dụ: Khởi tạo giá trị và kiểm tra tổ chức lưu trữ của mảng một chiều.

```
#include <iostream>
#include <iomanip>
using namespace std;
void Init ( int A[], int n ){
    for (int i=0; i<n; i++) { cout<<"Nhap A["<<i<<"]="; cin>>A[i]; }
}
void Address_Array ( int A[], int n) {
    cout<<"\n Dia chi cac phan tu:"<<endl;
    for(int i=0; i<n; i++) cout<<"Dia chi A["<<i<<"]="<<&A[i]<<endl;
}
void Result( int A[], int n) { cout<<"Noi dung mang:";
    for (int i=0; i<n; i++) cout<<A[i]<<setw(4);
}
int main(void ) {
    int A[10], n; // Khai báo mảng A[10] và số phần tử của mảng là n.
    cout<<"Nhap n="; cin>>n; //nhập n
    Init( A, n); // Khởi tạo giá trị cho mảng A gồm n phần tử
    Result( A, n); // Giá trị các phần tử của mảng A
    Address_Array( A, n); //Địa chỉ các phần tử của mảng A
    system("PAUSE");
    return 0;
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Ví dụ: Khởi tạo giá trị và kiểm tra tổ chức lưu trữ của mảng hai chiều.

```
#include <iostream>
using namespace std;
void Init(int A[][10], int n, int m){
    for(int i=0; i<n; i++)
        for (int j=0; j<m; j++){
            cout<<"Nhap A["<<i<<"]["<<j<<"]=""; cin>>A[i][j];
        }
}
void Result(int A[][10], int n, int m){
    cout<<"Dia chi va gia tri cac phan tu"<<endl;
    for (int i=0; i<n; i++)
        for (int j=0; j<m; j++){
            cout<<"Gia tri A["<<i<<"]["<<j<<"]="<<A[i][j]<<endl;
            cout<<"Dia chi A["<<i<<"]["<<j<<"]="<<&A[i][j]<<endl;
        }
}
int main(void ) { int A[10][10], n, m;
    cout<<"Nhap n, m:"; cin>>n>>m;
    Init(A, n, m); Result(A,n,m); system("PAUSE");
    return 0;
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

```
#include <iostream>
#include <iomanip>
using namespace std;

int main ()
{
    int n[ 10 ]; // n is an array of 10 integers

    // initialize elements of array n to 0
    for ( int i = 0; i < 10; i++ )
    {
        n[ i ] = i + 100; // set element at location i to i + 100
    }
    cout << "Element" << setw( 13 ) << "Value" << endl;

    // output each array element's value
    for ( int j = 0; j < 10; j++ )
    {
        cout << setw( 7 )<< j << setw( 13 ) << n[ j ] << endl;
    }

    return 0;
}
```


Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Bài tập: Cấu trúc dữ liệu mảng.

Viết chương trình sắp xếp mảng theo thứ tự giảm dần/tăng dần

Nhập một số x chèn vào mảng sao cho vẫn đảm bảo tính sắp xếp giảm dần/tăng dần

Tính tổng các số chẵn/lẻ có trong mảng

Tính tổng các số trong các ô có thứ tử lẻ có trong mảng

Viết chương trình xây dựng các thao tác trên đa thức.

a) Khởi tạo đa thức $P_n(x)$, $Q_m(x)$;

b) Tìm $P_n(x_0)$;

c) Tìm đạo hàm cấp I của đa thức;

d) Tìm $R(x) = P_n(x) + Q_m(x)$;

e) Tìm $R(x) = P_n(x) - Q_m(x)$;

f) Tìm $R(x) = P_n(x) * Q_m(x)$;

g) Tìm $R(x) = P_n(x) / Q_m(x)$ và đa thức dư.

Viết chương trình xây dựng các thao tác trên ma trận.

a) Tạo lập ma trận A cấp N, B cấp M;

b) Nhân hai ma trận.

c) Tìm hạng của ma trận.

d) Tìm vector riêng và giá trị riêng.

e) Tính định thức.

f) Tính nghịch đảo;

g) Giải hệ PTTT thuần nhất bằng phương pháp Cramer..

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Xâu ký tự (String)

Định nghĩa. Xâu ký tự là một mảng, mỗi phần tử của nó là một ký tự, ký tự cuối cùng là ký tự *null* ('\0') chỉ rõ điểm kết thúc của xâu ký tự. Đây là định nghĩa được mô tả trong thư viện chuẩn của C. Muốn sử dụng các hàm này, ta chỉ cần khai báo sử dụng thư viện <cstring> thay cho <string.h>.

Khai báo:

`char str[20];` //Khai báo xâu ký tự độ dài không quá 20.

`char str[] = "Hello";` //Vừa khai báo vừa khởi đầu cho xâu ký tự.

Tổ chức lưu trữ xâu ký tự được thể hiện như bảng một chiều dưới đây:

Phần tử:	str[0]	str[1]	str[2]	str[3]	str[4]	str[5]
Giá trị:	H	e	l	l	o	'\0'
Địa chỉ:	0x2341	0x2342	0x2343	0x2344	0x2345	0x2346

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Ví dụ: Kiểm tra giá trị phần tử trong chuỗi ký tự.

```
#include <iostream>
#include <cstring>
using namespace std;

int main(void) {
    char str[] = "Hello";
    cout<<"Xau str:"<<str<<endl;
    int n = strlen(str); //tính độ dài xau str[]
    for (int i=0; i<=n; i++){
        cout<<"Ky tu str["<<i<<"]="<<str[i];
    }
    system("PAUSE");
    return(0);
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Một số hàm thông dụng trong cstring:

STT	Tên hàm và ý nghĩa của hàm
1	strcpy(s1, s2): copy chuỗi s2 vào s1.
2	strcat(s1, s2): nối chuỗi s2 vào sau chuỗi s1
3	strlen(s) : tính độ dài chuỗi s.
4	strchr(s, c): tìm vị trí đầu tiên của ký tự c trong s
5	strstr(s1, s2): tìm vị trí đầu tiên của s2 trong s1.
6	strrev(s) : đảo ngược chuỗi s.
7	strcmp (s1, s2) : so sánh hai chuỗi s1 và s2 theo thứ tự từ điển. Hàm trả lại giá trị lớn hơn 0 khi s1>s2, nhỏ hơn 0 khi s1<2, bằng 0 khi s1=s2.

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Ví dụ: Thực hiện các hàm xử lý chuỗi ký tự.

```
#include <iostream>
#include <cstring> //Nhớ chỗ này
using namespace std;

int main(void ) {
    char s1[] = "Hello"; //xau s1 khai dau la Hello
    char s2[] = "World"; //xau s2 khai dau la World
    char s3[10]; //khai bao s3 do dai khong qua 10
    strcpy(s3, s1); //copy s1 vao s3
    cout<<"s3="<<s3<<endl; //s3 chinh la "Hello"
    int k = strcmp(s1, s2); //so sanh s1 va s2
    cout<<"k ="<<k<<endl;
    strcat(s1,s2); //noi s2 vao sau s1
    cout<<"s1="<<s1<<endl; //s1 chinh la "HelloWorld"
    k = strlen(s1); //tinh do dai xau s1
    cout<<"Do dai s1:"<<k<<endl;
    system("PAUSE");
    return 0;
}
```


Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Xâu ký tự (String)

Thư viện chuẩn của C++ cung cấp một tập các phép toán và thao tác (phương thức) với chuỗi ký tự được khai báo trong lớp **string**. Từ bây giờ ta hiểu một mảng các ký tự, một string đều là chuỗi ký tự và chỉ khác biệt về cú pháp khai báo.

Khai báo:

```
char    str[20] ; // Khai báo kiểu char ta phải đưa vào số lượng phần tử
string  str; //Khai báo kiểu string không cần đưa vào số lượng phần tử
string str = "Hello World"; // Vừa khai báo vừa khởi đầu
string str ("Hello World"); // Vừa khai báo vừa khởi đầu thế này cũng được
```

Các phép toán đối với string: giả sử ta có các string s1, s2. Khi đó:

STT	Phép toán	Ý nghĩa
1	s1 = s2	Copy s2 vào s1
2	s1 = s1 + s2	Nối s2 vào sau s1
3	(s1==s2)	Hỏi s1 và s2 có đúng bằng nhau hay không?
4	(s1>s2)	Hỏi s1 có lớn hơn s2 hay không?
5	(s1<s2)	Hỏi s1 có bé hơn s2 hay không?
6	(s1<=s2)	Hỏi s1 có bé hơn hoặc bằng s2 hay không?
7	(s1!=s2)	Hỏi s1 và s2 có giống nhau hay không?

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Ví dụ: Kiểm tra các phép toán với string.

```
#include <iostream>
#include <string>
using namespace std;
int main(void ){
    string s1("Hello"), s2 = "World", s3;
    cout<<"Gia tri s1="<<s1<<endl;
    cout<<"Gia tri s2="<<s2<<endl;
    s3 = s1; //copy hai string
    cout<<"Gia tri s3="<<s3<<endl;
    s3 = s1+s2; //noi hai xau
    cout<<"Gia tri s3="<<s3<<endl;
    cout<<"Kiem tra s1==s2:"<<(s1==s2)<<endl;
    cout<<"Kiem tra s1>s2:"<<(s1>s2)<<endl;
    cout<<"Kiem tra s1<s2:"<<(s1<s2)<<endl;
    cout<<"Kiem tra s1>=s2:"<<(s1>=s2)<<endl;
    cout<<"Kiem tra s1<=s2:"<<(s1<=s2)<<endl;
    cout<<"Kiem tra s1!=s2:"<<(s1!=s2)<<endl;
    system("PAUSE"); return(0);
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Một số hàm (phương thức) trong lớp string của C++: Giả sử s, s1, s2 là các string. Khi đó:

STT	Tên hàm (Phương thức)	Ý nghĩa
1	s.size()	Trả lại độ dài string s
2	s.length()	Trả lại độ dài string s
3	getline(cin, s)	Nhập một dòng từ bàn phím cho string s
4	s.erase(n, k)	Xóa k ký tự trong s kể từ vị trí thứ n
5	s.insert(n, s1)	Chèn s1 vào s kể từ vị trí thứ n.
6	s.insert(n, s1, k, m)	Chèn m ký tự kể từ ký tự thứ k trong s1 vào s kể từ vị trí thứ n.
7	s.replace(n, k, s1)	Thay thế k ký tự trong s kể từ vị trí thứ k bằng xâu s1.
8	s.find(s1)	Trả lại vị trí xuất hiện đầu tiên của s1 trong s.
9	s.rfind(s1)	Trả lại vị trí xuất hiện tiếp theo của s1 trong s.
10	s.at(int i)	Truy nhập đến phần tử thứ i trong string

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Ví dụ: Kiểm tra các phép toán với string.

```
#include <iostream>
#include <string>
using namespace std;
int main(void) {
    string s("ABCDEFGHGIJK"), s1("EF");
    cout<<"Do dai s:"<<s.length()<<"Do dai s1:"<<s1.size()<<endl;
    s.erase(4, 2); //xoa "EF"
    cout<<"Gia tri moi cua s:"<<s<<endl;
    s.insert(4,s1); //chen s1 vao s ke tu vi tri 4
    cout<<"Gia tri moi cua s:"<<s<<endl;
    s.replace(s.length(),0,s1); //thay the s1 vao cuoi
    cout<<"Gia tri moi cua s:"<<s<<endl;
    int k = s.find(s1); //tim vi tri dau tien cua s2 trong s
    cout <<"Vi tri dau tien "<<s1 <<" : " <<k<<endl;
    k = s.rfind(s1);
    cout <<"Vi tri ke tiep "<<s1 <<" : " <<k<<endl;
    system("PAUSE");return(0);
}
```

Lịch sử và các phương pháp lập trình

Các phương pháp lập trình - Lập trình cấu trúc

Viết chương trình đoán 1 chữ đã có sẵn trong một mảng

Viết chương trình đếm số kí tự có trong một mảng (không sử dụng strlen())

Viết chương trình đổi những kí tự thường thành kí tự hoa trong một mảng
 $t[i] = t[i] + ('A' - 'a')$

Viết chương trình xoá tất cả các nguyên âm có trong mảng