KIỂU CHUỖI - KIỂU DỮ LIỆU CẤU TRÚC

CÁC BÀI TẬP CƠ BẢN

Bài 1: Nhập một chuỗi S từ bàn phím. Kiểm tra xem chuỗi có phải là chuỗi đối xứng.

```
#include <iostream>
using namespace std;
// Nên đặt prototype ở đây để biết hàm nhận vào cái gì và trả về cái gì
// Đầu vào: Chuỗi cần kiểm tra
// Đầu ra: 1 nếu là chuỗi đối xứng, 0 nếu là chuỗi không đối xứng
int LaChuoiDoiXung(char[]);

int main()
{
    char str[128];
    cout << "Nhap 1 chuoi: ";
   // hàm cin.getline lấy các giá trị người dùng nhập vào và lưu vào chuỗi kí tự,
  // việc lấy các giá trị này kết thúc khi người dùng nhập kí tự newline như \n
 // tham số 1: chuỗi ký tự
 // tham số 2: độ lớn chuỗi ký tự
    cin.getline(str, 128);

    int kq = LaChuoiDoiXung(str);
    if (kq == 1)
        cout << "Chuoi " << str << " la chuoi doi xung.\n"
    else
        cout << "Chuoi " << str << " khong la chuoi doi
xung.\n"

    return 0;
}
int LaChuoiDoiXung(char str[])
{
    // strlen để lấy số lượng kí tự trong chuỗi. Nên tính 1 lần để sử dụng lại
    int l = strlen(str);
    for (int i = 0; i < l/2; i++)
    {
        if (str[i] != str[l - i - 1])
        {
            return 0;
        }
    }
    return 1;

}
```

Bài 2: Nhập một chuỗi S từ bàn phím. Tìm ký tự xuất hiện nhiều nhất trong chuỗi đó và số lần xuất hiện.

```
#include <iostream>
using namespace std;
/*
Đầu vào: Chuỗi cần kiểm tra và biến chứa ký tự xuất hiện nhiều nhất tìm được
Đầu ra: Số lần xuất hiện nhiều nhất của ký tư.
Bằng 0 nếu không có ký tự nào
*/
int KyTuXuatHienNhieuNhat(char[], char &);

int main()
{
    char str[128];
    cout << "Nhap 1 chuoi: ";

    cin.getline(str, 128);

    char chr;
    int max = KyTuXuatHienNhieuNhat(str, chr);

    if (max != 0) // Trường hợp max khác 0 - Chuỗi không rỗng
        cout << "Ky tu " << chr << " xuat hien nhieu nhat la "
             << max << " lan \n";
    else
        cout << "Chuoi rong! Khong co ky tu xuat hien nhieu
nhat \n";

    return 0;
}
```

```
int KyTuXuatHienNhieuNhat(char str[], char &chr)
{
    int i, j, length;
    length = strlen(str);
    char curchr;     // Kí tự đang xét
    int curcount;    // Số lần xuất hiện của kí tự đang xét
    int max = 0;     // Số lần xuất hiện nhiều nhất ban đầu là 0

    for (i = 0; i<length; i++)
    {
        curchr = str[i];
        curcount = 1;
        // Lấy ký tự thứ i ra kiểm tra với các kí tự sau i

        for (j = i + 1; j<length; j++)
            if (str[j] == str[i])
                curcount++;
        /* Tìm được số lần xuất hiện nhiều hơn max thì cập nhật lại số lần và
kí tự*/
        if (max < curcount)
        {
            max = curcount;
            chr = curchr;

        }
    }
    return max;
}
```

Bài 3: Cho kiểu cấu trúc sinh viên gồm các thuộc tính: Họ tên, MSSV, Lớp
Viết chương trình nhập xuất dữ liệu cho 1 danh sách sinh viên

```
#include <iostream>

using namespace std;

typedef struct sinhvien
{
    char ten[25];
    char mssv[10];
    char lop[5];
}SV;
```

```cpp
void NhapSV(SV &sinhvien)
{
    cout << "Nhap ten sinh vien : ";
    cin.getline(sinhvien.ten, 25);

    cout <<"Nhap ma so sinh vien : ";
    cin.getline(sinhvien.mssv, 10);

    cout <<"Nhap lop cua sinh vien : ";
    cin.getline(sinhvien.lop, 5);
}

void XuatSV(SV sinhvien)
{
    cout <<"Ten sinh vien : ";
    cout << sinhvien.ten << endl;

    cout <<"Ma so sinh vien : ";
    cout << sinhvien.mssv << endl;

    cout <<"Lop cua sinh vien : ";
    cout << sinhvien.lop << endl;
}

int main(){
    SV dssv[10];
    int n = 0;

    cout << "Nhap so luong sinh vien : ";
    cin >> n;
    //Bỏ qua kí tự newline để hàm cin.getline lấy các kí tự từ sau kí tự newline
    cin.ignore();

    for (int i = 0; i < n; i++)
    {
        cout << "Nhap thong tin cho sinh vien thu " << i + 1
             << endl;

        NhapSV(dssv[i]);
    }

    for (int i = 0; i < n; i++)
    {
        cout << "Thong tin sinh vien thu " << i + 1 << endl;
        XuatSV(dssv[i]);
    }

    return 0;
```

CÁC BÀI TẬP THÊM CÓ ĐỘ KHÓ TRUNG BÌNH

KIỂU CHUỖI

1. Nhập họ tên của một người từ bàn phím. Hãy chuẩn hóa chuỗi họ tên này. (Xóa các khoảng trắng thừa và ký tự đầu tiên của họ, chữ lót và tên phải viết hoa, các ký tự còn lại viết thường).

Ví dụ:

Nhập: " NgUyen VaN A "

Xuất: "Nguyen Van A"

Gợi ý: Viết các hàm con và gọi theo thứ tự:

1.xoá khoảng trắng đầu chuỗi

2. xoá khoảng trắng cuối chuỗi

3. xoá khoảng dư ở giữa

4. đổi chữ thường

5. đổi chữ hoa

2. Không sử dụng các hàm có sẵn. Viết chương trình xóa N ký tự tại vị trí i trong chuỗi S.

Ví dụ:

Nhập: S = "Nguyen Van A"; i = 2; N = 3

(Xóa 3 ký tự tại ký tự 2 trong chuỗi S)

Xuất: S = "Nen Van A"

Gợi ý: Viết các hàm con xoá 1 phần tử trong chuỗi và tiến hành duyệt chuỗi từ vị trí i đến vị trí i + 2, sau đó gọi hàm xoá 1 phần tử đã có.

3. Viết chương trình nhập một số nguyên, xuất lại số đó ở dạng chuỗi nhưng có dấu "," ngăn cách hàng triệu, ngàn.

Ví dụ:

Nhập: N = 123456789

Xuất: S = "123,456,789"

Gợi ý: Duyệt chuỗi ngược

4. Nhập một chuỗi S từ bàn phím và một ký tự C. Đếm xem ký tự C xuất hiện bao nhiêu lần trong chuỗi S đó.

Nhập: "Nguyen Van A" ; C = "u"; => **Xuất**: 1 lần

5. Lập trình nhập vào từ bàn phím danh sách học sinh một lớp, sắp xếp lại danh sách theo thứ tự abc của Tên, nếu trùng Tên thì sắp xếp theo thứ tự abc của Họ.

Gợi ý: Sử dụng hàm so sánh strcmp

6. Viết chương trình nhập từ bàn phím 2 xâu ký tự S1 và S2. Hãy xét xem S1 có xuất hiện bao nhiêu lần trong S2 (hoặc ngược lại S2 xuất hiện bao nhiêu lần trong S1) và tại những vị trí nào?

7. Viết chương trình nhập một xâu S chỉ gồm các chữ cái thường. Hãy lập xâu S1 nhận được từ xâu S bằng cách sắp xếp lại các ký tự theo thứ tự abc.

8. Lập trình tính giá trị của một số viết dưới dạng LA MÃ.

Ví dụ: MDCLXVI = 1666.

M:1000; D:500; C:100; L:50; X :10; V:5 ; I :1

KIỂU CẤU TRÚC

1. Khai báo kiểu dữ liệu biểu diễn khái niệm phân số trong toán học và định nghĩa hàm nhập, hàm xuất cho kiểu dữ liệu này.
   Gợi ý:

```
typedef struct phanso
{
    int tu;
    int mau;
}PS;

void NhapPS(PS &ps)
{
    cin >> ps.tu;
    cin >> ps.mau;
}

void XuatPS (PS ps)
{
    cout << ps.tu << "/" << ps.mau;
}
```

2. Khai báo kiểu dữ liệu biểu diễn khái niệm hỗn số trong toán học và định nghĩa hàm nhập, hàm xuất cho kiểu dữ liệu này.

Gợi ý:

```
typedef struct honso
{
    int nguyen;
    int tu;
    int mau;
}HS;
```

3. Khai báo kiểu dữ liệu biểu diễn khái niệm điểm trong mặt phẳng Oxy và định nghĩa hàm nhập, hàm xuất cho kiểu dữ liệu này.

Gợi ý:

```
typedef struct DiemXY
{
    float x;
    float y;
}DIEMXY;

void NhapDIEMXY(DIEMXY &diem)
{
    float temp;

    cin >> temp;
    diem.x = temp;

    cin >> temp;
    diem.y = temp;
}

void XuatDIEMXY(DIEMXY diem)
{
    cout << "(" << diem.x << "," << diem.y << ")";
}
```

4. Khai báo kiểu dữ liệu biểu diễn khái niệm điểm trong mặt phẳng Oxyz và định nghĩa hàm nhập, hàm xuất cho kiểu dữ liệu này.

Gợi ý:

```
typedef struct DiemXYZ
{
    float x;
    float y;
    float z;
}DIEMXYZ;
```

5. Khai báo kiểu dữ liệu biểu diễn khái niệm đơn thức $P(x) = ax^n$ trong toán học và định nghĩa hàm nhập, hàm xuất cho kiểu dữ liệu này.

Gợi ý:

```
typedef struct DonThuc
{
    int a;
    int mu;
    float x;
}DT;

void NhapDonThuc(DT &donthuc)
{
    cin >> donthuc.a;
    cin >> donthuc.mu;
    float temp;
    cin >> temp;
    donthuc.x = temp;
}

void XuatDonThuc(DT donthuc)
{
    cout << donthuc.a << "*" << donthuc.x << "^" << donthuc.mu;
}
```

6. Khai báo kiểu dữ liệu biểu diễn khái niệm đa thức một biến $P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$ trong toán học và định nghĩa hàm nhập, hàm xuất cho kiểu dữ liệu này.

Gợi ý:

```
typedef struct DaThuc
{
    int soluong;
    DonThuc mangDonThuc[];
}DATHUC;


void NhapDaThuc (DATHUC &dathuc)
{
    cin >> dathuc.soluong;
    for (int i = 0; i < dathuc.soluong; i++)
        NhapDonThuc(dathuc.mangDonThuc[i]);
}
```

```
void XuatDaThuc (DATHUC dathuc)
{
    for (int i = 0; i < dathuc.soluong - 1; i++)
    {
        cout << "(";
        XuatDonThuc(dathuc.mangDonThuc[i]);
        cout << ") + ";
    }

    cout << "(";
    XuatDonThuc(dathuc.mangDonThuc[dathuc.soluong - 1]);
    cout << ")";

}
```

7. Khai báo kiểu dữ liệu biểu diễn khái niệm ngày trong thế giới thực và định nghĩa hàm nhập, hàm xuất cho kiểu dữ liệu này.
8. Khai báo kiểu dữ liệu biểu diễn khái niệm đường thẳng trong mặt phẳng Oxy và định nghĩa hàm nhập, hàm xuất cho kiểu dữ liệu này.
9. Khai báo kiểu dữ liệu biểu diễn khái niệm đường tròn trong mặt phẳng Oxy và định nghĩa hàm nhập, hàm xuất cho kiểu dữ liệu này.

Gợi ý:

```
typedef struct DuongTron
{
    DIEMXY tam;
    float bankinh;
}DUONGTRON;
```