

CHƯƠNG 1: TỔNG QUAN VỀ NGÔN NGỮ C++

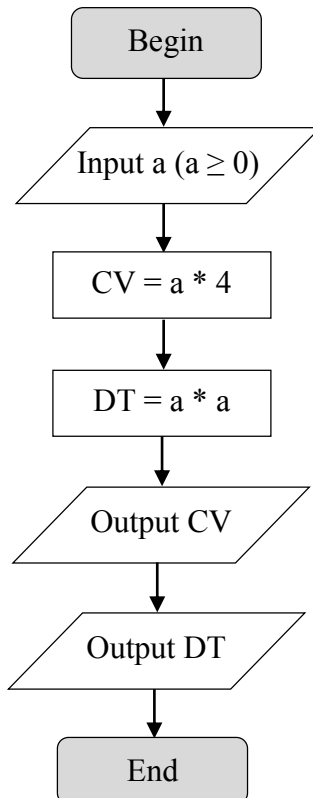


1. Viết chương trình in nội dung một bài thơ mà bạn thích.

Sinh viên tự làm

2. Viết chương trình nhập vào cạnh của hình vuông, in ra chu vi và diện tích.

Giải thuật:



3. Viết chương trình nhập vào 4 giá trị lần lượt là số thực, nguyên, nguyên dài và ký tự. In ra màn hình các giá trị này.

Sinh viên tự làm

4. Viết chương trình nhập vào một ký tự, in ra ký tự đó và mã ASCII của nó.

Sinh viên tự làm

5. Viết chương trình nhập 3 số thực a, b, c, in ra màn hình hàng chữ phương trình có dạng $ax^2 + bx + c = 0$, trong đó các giá trị a, b, c chỉ in 2 số lẻ (ví dụ với a = 5.141, b = -2, c = 0.8 in ra $5.14 x^2 - 2.00 x + 0.80$).

Chương trình:

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

```
main()
{
    float a,b,c;
    cout<<"Nhập 3 hệ số:";
    cin>>a>>b>>c;
    cout<<fixed<<setprecision(2);
    cout<<a<<"x^2 ";
    cout<<showpos<<fixed<<setprecision(2);
    cout<<b<<"x " <<c<<endl;
}
```

6. Viết chương trình in ra tổng, tích, hiệu và thương của 2 số thực được nhập vào từ bàn phím.

Sinh viên tự làm

7. Viết chương trình in ra trung bình cộng, trung bình nhân của 3 số nguyên được nhập vào từ bàn phím.

Sinh viên tự làm

8. Viết chương trình nhập vào 4 chữ số. In ra tổng của 4 chữ số này và chữ số hàng chục, hàng đơn vị của tổng đó (ví dụ 4 chữ số 3, 1, 8, 5 có tổng là 17 và chữ số hàng chục là 1 và hàng đơn vị là 7, **in ra** 17, 1, 7).

Giải thuật:

Begin

```
Input (a, b, c, d);
Tong = a + b + c + d - 4 * 48;
Output ("Tổng là: ", Tong);
Output ("Chữ số hàng chục là: ", Tong div 10);
Output ("Chữ số hàng đơn vị là: ", Tong mod 10);
```

End

9. Viết chương trình nhập vào bán kính của hình tròn, in ra diện tích, chu vi của nó.

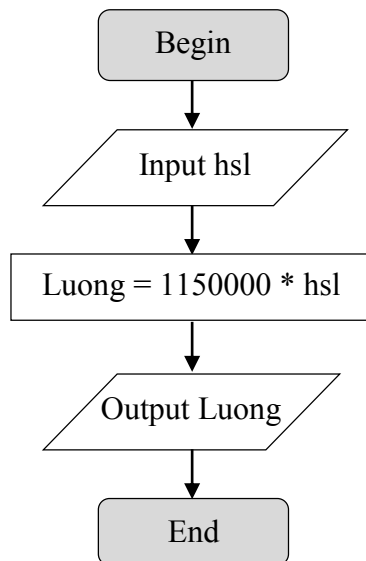
Sinh viên tự làm

10. Viết chương trình nhập vào một số nguyên (có 4 chữ số). In ra tổng của 4 chữ số này và chữ số đầu, chữ số cuối (ví dụ số 3185 có tổng các chữ số là 17, chữ số đầu và cuối là 3 và 5, **in ra** 17, 3, 5).

Sinh viên tự làm

11. Viết chương trình nhập vào hệ số lương (là số thực có 2 chữ số lẻ) của 1 nhân viên. In ra lương của nhân viên đó, với $\text{lương} = 1.150.000 * \text{hệ số lương}$.

Giải thuật:



12. Viết chương trình nhập 2 số nguyên a và b, hãy đổi giá trị của a và b theo 2 cách:

- Dùng biến phụ t: $t = a; a = b; b = t;$
- Không dùng biến phụ: $a = a + b; b = a - b; a = a - b;$

In kết quả ra màn hình giá trị của a và b trước và sau khi đổi.

Sinh viên tự làm

13. Viết chương trình nhập vào năm sinh của 1 người. Cho biết người đó năm nay được bao nhiêu tuổi.

Chương trình:

```
#include <iostream>
#include <time.h>
using namespace std;
main()
{
    int ns;
    cout<<"Nhập vào năm sinh của bạn: ";
    cin>>ns;
    time_t t = time(0);
    struct tm *Now = localtime(&t);
    int tuoi = Now->tm_year + 1900 - ns;
    cout<<"Bây giờ bạn đã "<<tuoi<<" tuổi."<<endl;
}
```

14. Viết chương trình tính chỉ số pignet của 1 người khi biết chiều cao (cm), vòng ngực trung bình (cm) và trọng lượng (kg).

Với: chỉ số pignet = chiều cao - (vòng ngực trung bình + trọng lượng)

Sinh viên tự làm

15. Viết chương trình nhập vào chiều dài 3 cạnh a, b, c của 1 tam giác. Tính chu vi và diện tích của tam giác đó theo công thức sau:

Chu vi : $CV = a + b + c$

Diện tích : $DT = \sqrt{p * (p - a) * (p - b) * (p - c)}$ với $p = CV/2$.

Giải thuật:

Begin

Input (a, b, c);

$CV = a + b + c$;

$p = CV/2$;

$DT = \text{sqrt}(p * (p - a) * (p - b) * (p - c))$;

Output ("Chu vi = ", CV);

Output ("Diện tích = ", DT);

End

CHƯƠNG 2: CÁC CẤU TRÚC ĐIỀU KHIỂN

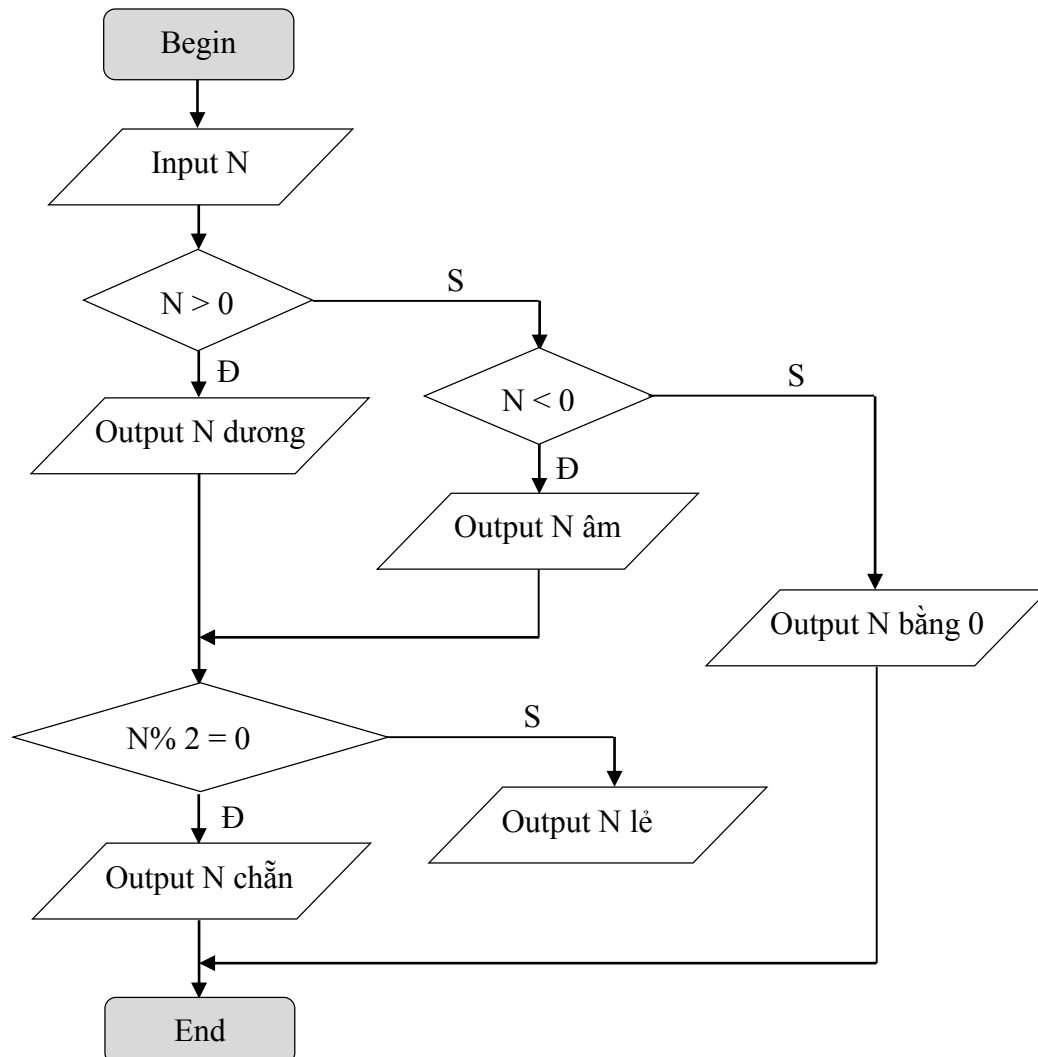


1. Viết chương trình nhập một ký tự, cho biết ký tự đó có phải là chữ cái hay không.

Sinh viên tự làm

2. Viết chương trình nhập vào một số nguyên, hãy cho biết số đó là âm hay dương, chẵn hay lẻ.

Giải thuật:



3. Viết chương trình nhập 3 số a, b, c in ra max, min của 3 số đó.

Sinh viên tự làm

4. Viết chương trình nhập 3 số a, b, c, hãy cho biết 3 số trên có thể là độ dài 3 cạnh của một tam giác.

Giải thuật:

Begin

Input (a, b, c);

If(a + b > c and a + c > b and b + c > a)

Output ("a, b, c là 3 cạnh của một tam giác");

Else

Output ("a, b, c không là 3 cạnh của một tam giác");

End

5. Viết chương trình nhập vào điểm cơ bản (đcb) và điểm nâng cao (đnc) cho 1 học viên. Cho biết học viên này được xếp loại gì, với cách xếp loại dựa vào điểm trung bình (đtb) như sau:

- Nếu đtb ≥ 9 và không có điểm nào dưới 8 thì được xếp loại xuất sắc
- Nếu đtb ≥ 8 và không có điểm nào dưới 7 thì được xếp loại giỏi
- Nếu đtb ≥ 7 và không có điểm nào dưới 6 thì được xếp loại khá
- Nếu đtb ≥ 5 và không có điểm nào dưới 3 thì được xếp loại trung bình
- Còn lại thì ghi không đạt

Sinh viên tự làm

6. Viết chương trình làm việc như 1 máy tính bỏ túi:

- Nhập vào 2 số
- Hỏi toán tử là +, -, * hay /, tương ứng in ra tổng, hiệu, tích, thương.
- Nếu không phải là các toán tử trên thì kết thúc chương trình.

Chương trình:

```
#include <iostream>
using namespace std;
main()
{
    float a, b, kq;
    char pt;
    cout<<"Hay nhap 2 so thuc a, b: ";
    cin>>a>>b;
    cout<<"Hay nhap 1 phep toan: ";
    cin>>pt;
    switch (pt)
    {
```

```
        case '+': kq = a + b; cout<<"Ket qua la: "<<kq;
                break;
        case '-': kq = a - b; cout<<"Ket qua la: "<<kq;
                break;
        case 'x': case '.': case '*': kq = a * b ;
                cout<<"Ket qua la: "<<kq; break ;
        case ':': case '/':
            if (b !=0)
            {
                kq = a / b;
                cout<<"Ket qua la: "<<kq;
            }
            else
                cout<<"Khong thuc hien duoc phép chia";
            break;
        default:
            cout<<"Nhap phép toan sai";
    }
}
```

7. Viết chương trình tính tích N số nguyên dương đầu tiên, với N được nhập từ bàn phím.

Sinh viên tự làm

8. Viết chương trình tính tổng $S = 1/2 + 2/3 + 3/4 + \dots + n/(n+1)$, với n là số nguyên dương được nhập từ bàn phím.

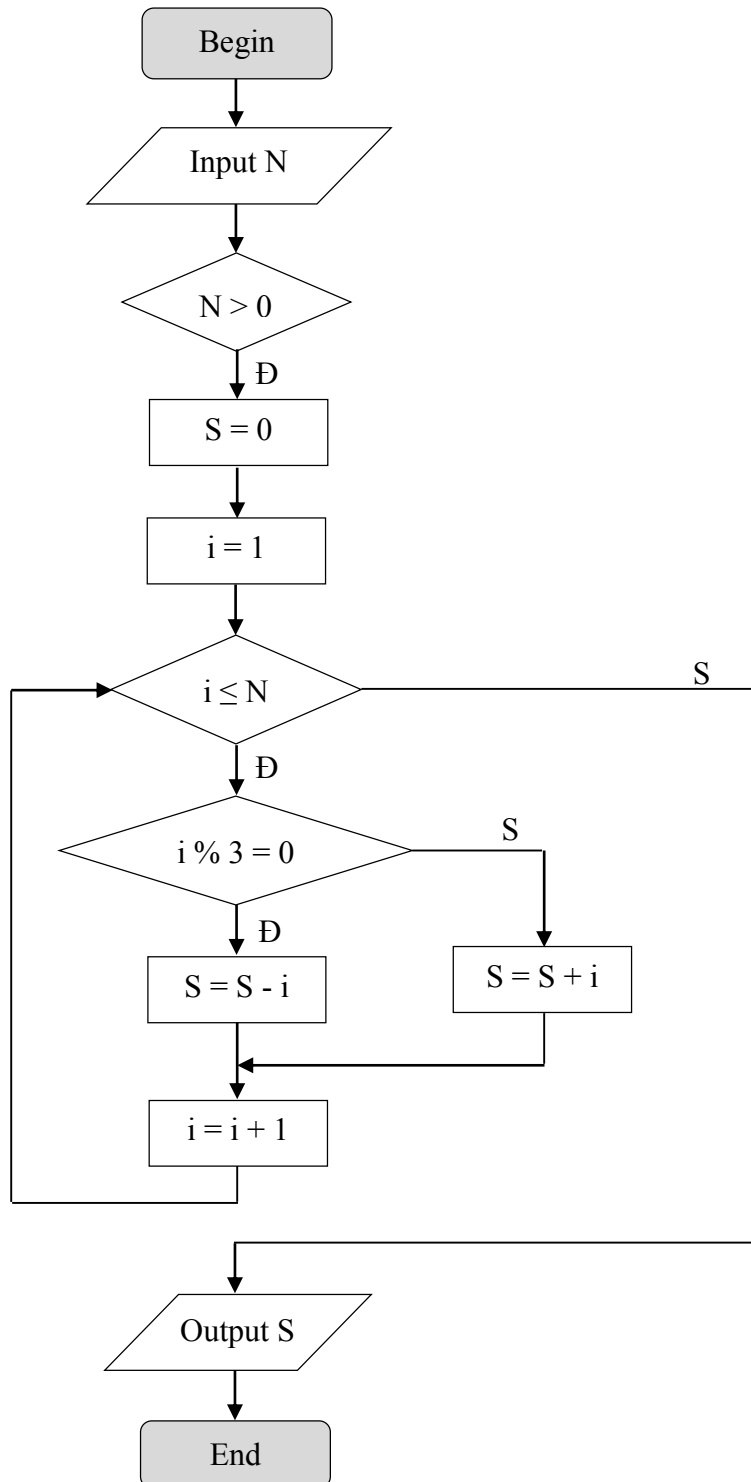
Chương trình:

```
#include <iostream>
using namespace std;
main()
{
    int n, i;
    float s=0;
    cout<<"Nhap vao so n: ";
    cin>>n;
```

```
for(i=1; i<=n; i++)  
    s=s+(float)i/(i+1);  
cout<<"Tong s = "<<s<<"\n";  
}
```

9. Viết chương trình tính tổng $S = 1 + 2 - 3 + 4 + 5 - 6 + 7 + 8 - 9 + \dots + /- n$, với n là số nguyên dương được nhập từ bàn phím.

Giải thuật:



10. Viết chương trình tính tổng $S = \sum_{i=1}^n \sum_{j=1}^m i + 2j$, với n, m là số nguyên dương được nhập từ bàn phím.

Sinh viên tự làm

11. Viết chương trình tính $e = 1 + \frac{x}{1!} - \frac{x^2}{2!} + \frac{x^3}{3!} - \frac{x^4}{4!} + \dots + (-1)^{n+1} \frac{x^n}{n!}$, với n là số nguyên dương, x là số thực được nhập từ bàn phím.

Chương trình:

```
#include <iostream>
using namespace std;
main()
{
    int n, i;
    float x, e = 1, lt = 1;
    long gt = 1;
    cout<<"Nhap n = ";
    cin>>n;
    cout<<"Nhap x = ";
    cin>>x;
    for(i=1; i<=n; i++)
    {
        lt = lt * x;
        gt = gt * i;
        e = e + lt/gt;
    }
    cout<<"Tong e = "<<e<<'\n';
}
```

12. Viết chương trình tính $e = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/n!$, cho đến khi $1/(n+1)!$ nhỏ hơn Epsilon, với Epsilon là 1 lượng khá nhỏ được nhập từ bàn phím.

Sinh viên tự làm

13. Viết chương trình nhập vào số nguyên dương n .

In ra tam giác có dạng sau:

@

@ @

@ @ @

@ @ @ @

(Nếu n nhập vào là 4)

Giải thuật:

Begin

Input (n);

For (i = 1 .. n) Do

For (j = 1 .. i) Do

Output ("@");

End

14. Viết chương trình nhập vào chiều dài và chiều rộng cho 1 hình chữ nhật. In ra hình chữ nhật đó bằng các dấu *. Giả sử ta nhập $cd = 6$, $cr = 4$ thì in:

```

* * * * *
* * * * *
* * * * *
* * * * *

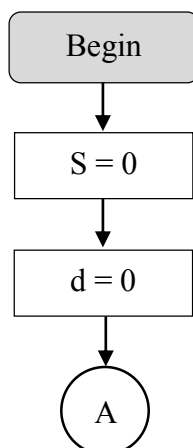
```

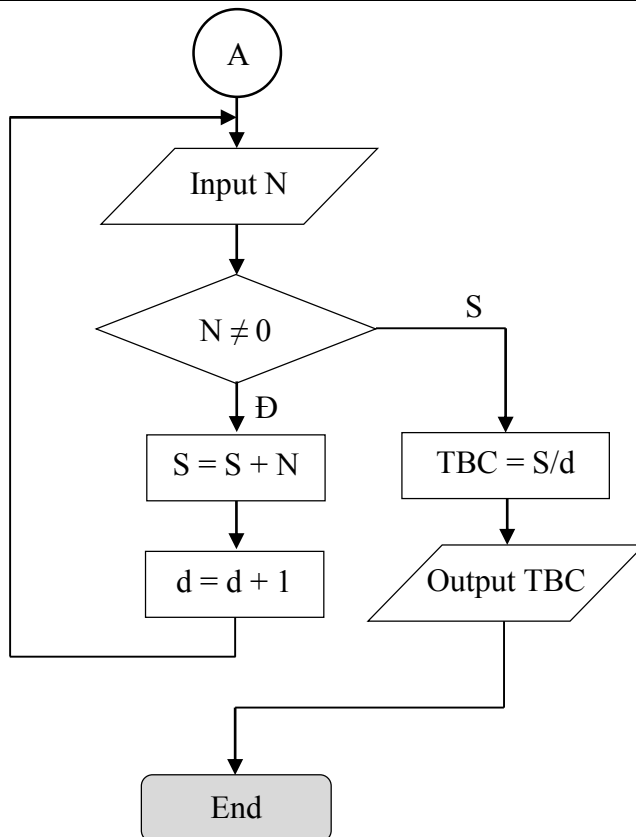
Sinh viên tự làm

15. Viết chương trình nhập vào lần lượt các số nguyên, quá trình nhập kết thúc khi nhập số nguyên là 0. Cho biết có bao nhiêu số dương, bao nhiêu số âm trong các số nguyên đã nhập.

Sinh viên tự làm

16. Viết chương trình nhập vào lần lượt các số thực, quá trình nhập kết thúc khi nhập số thực là 0. Cho biết trị trung bình của các số đã nhập.

Giải thuật:



17. Viết chương trình làm nhiều lần công việc sau: Nhập 1 số thực, in số đối của số thực đó. Chương trình kết thúc khi nào số thực nhập vào là 0.

Sinh viên tự làm

18. Viết chương trình in ra tất cả các số nguyên tố $< N$, N là trị nhập.

Sinh viên tự làm

19. Viết chương trình nhập vào 1 số nguyên dương, in ra ước số lẻ lớn nhất của nó. Ví dụ ta nhập số 60 thì in ra là 15.

Chương trình:

```

#include <iostream>
using namespace std;
main()
{
    int n;
    cout<<"Nhập vào số nguyên: ";
    cin>>n;
    int i=n-1;
    while(n%i!=0 || i%2==0)
        i--;
  
```

```
cout<<i;  
}
```

20. Viết chương trình tính số hạng thứ n của dãy Fibonacci. Hướng dẫn: Dãy Fibonacci là dãy số gồm các số hạng $F(n)$ với $F(n) = F(n - 1) + F(n - 2)$ và $F(1) = F(2) = 1$

Ví dụ: Dãy Fibonacci gồm 10 số hạng đầu tiên là: 1 1 2 3 5 8 13 21 34

Sinh viên tự làm

21. Viết chương trình phân tích 1 số nguyên dương n thành tích của các thừa số nguyên tố.

Sinh viên tự làm

22. Viết lại chương trình của bài 8 bằng lệnh nhảy goto.

Chương trình:

```
#include <iostream>  
using namespace std;  
main()  
{  
    int n, i;  
    float s=0;  
    cout<<"Nhap vao so n: ";  
    cin>>n;  
    i=0;  
lap:i=i+1;  
    if (i<=n)  
    {  
        s=s+(float)i/(i+1);  
        goto lap;  
    }  
    cout<<"Tong s = "<<s<<'\n';  
}
```

23. Viết lại chương trình của bài 12 bằng lệnh nhảy goto.

Sinh viên tự làm

24. Viết lại chương trình của bài 16 bằng lệnh nhảy goto.

Sinh viên tự làm

CHƯƠNG 3: DỮ LIỆU KIỂU MẢNG



1. Viết chương trình nhập vào mảng 1 chiều gồm n phần tử kiểu số nguyên. In ra mảng vừa nhập theo thứ tự ngược. Cho biết có bao nhiêu phần tử có nội dung là số nguyên tố. Tính tích các phần tử là ước số của k , với k được nhập từ bàn phím. Cho biết phần tử X xuất hiện ở lần thứ m tại vị trí thứ mấy, với X và m được nhập từ bàn phím. Sắp xếp mảng theo thứ tự giảm dần. In ra mảng sau khi sắp xếp.

Chương trình:

```
#include <iostream>

using namespace std;

main()
{
    int a[20];
    int n, i, j, d, k, X, m, tam;
    do
    {
        cout<<"Nhập số phần tử n = ";
        cin>>n;
    } while (n <= 0 || n > 20);
    //Nhập mảng
    for(i=0; i<n; i++)
    {
        cout<<"Nhập phần tử thu "<<i+1<<" : ";
        cin>>a[i];
    }
    //In mảng theo thứ tự ngược
    cout<<"Mảng vừa nhập là : "<<endl;
    for(i=n-1; i>=0; i--)
        cout<<a[i]<<"\t";
    cout<<endl;
```

```
//Dem so phan tu la so nguyen to

d = 0;
for(i=0; i<n; i++)
{
    j=1;
    do
    {
        j++;
    } while(j <= a[i]/2 && a[i]%j != 0);
    if(a[i] > 1 && j > a[i]/2) d++;
}

cout<<"Co "<<d<<" phan tu la so nguyen to"<<endl;

//Tinh tich cac phan tu la uoc so cua k

cout<<"Nhap k = ";
cin>>k;
d = 0;
long tich=1;
for(i=0; i<n; i++)
    if(a[i]!=0 && k%a[i] == 0) tich*=a[i];
cout<<"Tich cac p.tu la uoc so cua "<<k<<" = "<<tich;
cout<<"Nhap X = ";
cin>>X;
do
{
    cout<<"Nhap m = ";
    cin>>m;
}while (m<=0 || m>n);
i=0;
d=0;
while(i<=n && d<m)
```

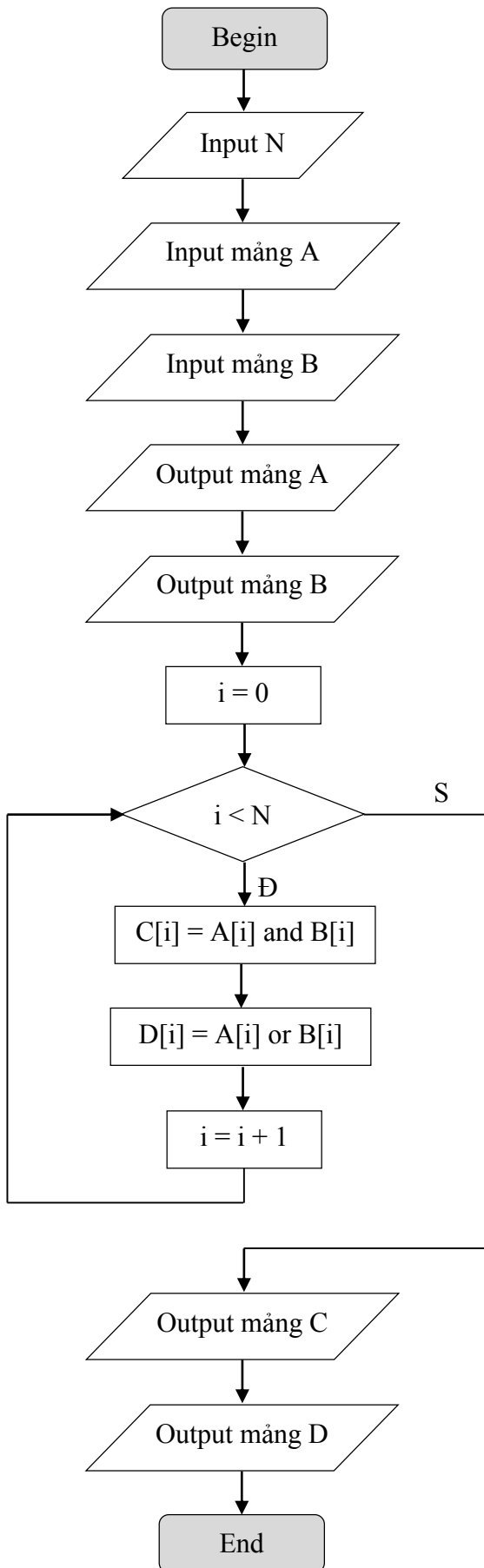
```
{
    if (a[i]==X) d++;
    i++;
}
if (i<=n)
    cout<<X<<" x.hien lan thu "<<m<<" tai v.tri "<<i-1;
else
    cout<<"Khong co "<<X<<" xuat hien o lan thu "<<m;
//Sap xep mang theo thu tu giam dan
for (i = 0; i < n-1; i++)
    for (j = i+1; j < n; j++)
        if (a[i] < a[j])
        {
            tam = a[i];
            a[i] = a[j];
            a[j] = tam;
        }
//In mang sau khi sap xep
cout<<"Mang sau khi sap xep tang dan la :"<<endl;
for (i = 0; i < n; i++)
    cout<<a[i]<<'\\t';
}
```

2. Viết chương trình nhập vào 2 mảng 1 chiều có cùng số phần tử và nội dung các phần tử kiểu số thực. Tạo ra mảng thứ 3 bằng tổng của 2 dãy đó ($c[i] = a[i] + b[i]$) in ra dãy vừa tạo.

Sinh viên tự làm

3. Viết chương trình nhập vào 2 mảng 1 chiều có cùng số phần tử và nội dung các phần tử kiểu bool. In ra 2 mảng vừa nhập trên 2 hàng khác nhau. Tạo ra mảng thứ 3 là kết quả của việc thực hiện phép toán AND trên từng phần tử của 2 mảng đã nhập. Tạo ra mảng thứ 4 là kết quả của việc thực hiện phép toán OR trên từng phần tử của 2 mảng đã nhập. In ra 2 mảng vừa tạo trên 2 hàng khác nhau.

Giải thuật:



4. Viết chương trình thực hiện các công việc sau:

- Nhập mảng 1 chiều gồm n phần tử kiểu số nguyên.
- In ra mảng vừa nhập.
- In ra vị trí của các phần tử lớn nhất có trong dãy.
- Tính trị trung bình của các phần tử dương có trong dãy.
- Đếm số phần tử là lũy thừa của K, với K nhập từ bàn phím.

Sinh viên tự làm

5. Viết chương trình thực hiện các công việc sau:

- Nhập mảng 1 chiều gồm n phần tử kiểu số nguyên.
- In ra các phần tử nằm ở các vị trí chẵn.
- Kiểm tra xem dãy có thứ tự hay không?
- Kiểm tra xem dãy có đối xứng không?
- Tạo ra 1 mảng mới được copy từ mảng nhập gồm M phần tử bắt đầu từ phần tử thứ K, với M và K được nhập từ bàn phím.
- In ra mảng vừa tạo.

Giải thuật:

Bước 1: Nhập mảng 1 chiều

Bước 1.1: Khai báo mảng (a) thuộc kiểu int

Bước 1.2: Nhập số phần tử của mảng (N)

Bước 1.3: Nhập mảng (a)

Bước 2: In ra các phần tử nằm ở các vị trí chẵn

For (i = 0 .. N-1) Do

```
{  
    cout<<a[i];  
    i = i + 2;  
}
```

Bước 3: Kiểm tra xem dãy có thứ tự không

Bước 3.1: kt = true;

Bước 3.2: i = 0;

Bước 3.3:

```
while(kt = true and i < N-1)  
{
```

```
j = i + 1;
while(kt = true and j < N)
    if(a[i] > a[j]) kt = false;
    else j++;
i++;
}
```

Bước 3.4:

```
if(kt = true) cout<<"Dãy tăng dần";
else
```

Bước 3.4.1: kt = true;

Bước 3.4.2: i = 0;

Bước 3.4.3:

```
while(kt = true and i < N-1)
{
    j = i + 1;
    while(kt = true and j < N)
        if(a[i] < a[j]) kt = false;
        else j++;
    i++;
}
```

Bước 3.4.4:

```
if(kt = true) cout<<"Dãy giảm dần";
else cout<<"Dãy không có thứ tự";
```

Bước 4: Kiểm tra xem dãy có đối xứng không

Bước 4.1: kt = true;

Bước 4.2: i = 0;

Bước 4.3:

```
while(kt = true and i <= N/2 - 1)
    if(a[i] == a[N - i - 1]) i++;
    else kt = false;
```

Bước 4.4:

```
if(kt = true) cout<<"Dãy đối xứng";
```

```
else cout<<"Dãy không đối xứng";
```

Bước 5: Tạo ra 1 mảng mới được copy từ mảng nhập gồm M phần tử bắt đầu từ phần tử thứ K, với M và K được nhập từ bàn phím

Bước 5.1: Nhập và kiểm tra K

Bước 5.2: Nhập và kiểm tra M

Bước 5.3: Khai báo mảng mới acopy

Bước 5.4:

```
For (i = 0 .. M) Do
```

```
    acopy[i] = a[i + K];
```

Bước 6: In mảng vừa tạo (acopy)

```
For (i = 0 .. M) Do
```

```
    cout<<acopy[i];
```

Bước 7: Kết thúc

6. Viết chương trình nhập vào ma trận a gồm m hàng, n cột, các phần tử kiểu số thực. In ra ma trận vừa nhập. Cho biết trong ma trận có bao nhiêu phần tử có phần nguyên là chẵn. Tính tích các phần tử dương nằm trên hàng h, với h nhập từ bàn phím. Sắp xếp các phần tử nằm trên cột c theo thứ tự tăng dần, với c nhập từ bàn phím. In ra ma trận sau khi sắp xếp.

Sinh viên tự làm

7. Viết chương trình nhập vào ma trận vuông cấp n, các phần tử kiểu ký tự. In ra ma trận vừa nhập. Cho biết trong ma trận có bao nhiêu ký tự 'T'. In ra các phần tử nằm trên đường chéo phụ. Cho biết ký tự lớn nhất nằm trên đường chéo chính là gì. Cho biết ma trận có hàng nào có thứ tự tăng dần không?

Chương trình:

```
#include <iostream>

using namespace std;

main()
{
    char a[11][11];
    int n, i, j, d;
    do
    {
        cout<<"Nhập cấp của ma trận : ";
```

```
    cin>>n;
}while (n<=0 || n>10);
//Nhap ma tran vuong
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
    {
        cout<<"Nhap a["<<i<<","<<j<<"]=" ";
        cin>>a[i][j];
    }
//In ma tran
cout<<"Ma tran vua nhap la:"<<endl;
for (i=0; i<n; i++)
{
    for (j=0; j<n; j++)
        cout<<a[i][j]<<" ";
    cout<<endl;
}
//Dem ky tu T co trong ma tran
d=0;
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
        if (a[i][j]=='T') d++;
cout<<"Co "<<d<<" ky tu T trong ma tran"<<endl;
//In cac phan tu nam tren duong cheo phu
cout<<"Cac phan tu nam tren duong cheo phu : ";
for (i=0; i<n; i++)
    cout<<a[i][n-1-i]<<" ";
cout<<endl;
//Tim ky tu lon nhat tren duong cheo chinh
char kt=a[0][0];
```

```
for (i=1; i<n; i++)
    if (a[i][i] > kt)
        kt = a[i][i];

cout<<"Ky tu lon nhat nam tren duong cheo chinh la :";
cout<<kt<<endl;

//Tim hang co thu tu tang dan
for(i=0; i<n; i++)
{
    j = 1;
    while(a[i][j]>=a[i][j-1]) j++;
    if(j>=n)
        cout<<"Hang "<<i<<" co thu tu tang dan"<<endl;
}
```

8. Viết chương trình nhập vào ma trận vuông cấp n , các phần tử kiểu số nguyên. In ra ma trận vừa nhập. In ra các phần tử nằm trên đường biên của ma trận. Cho biết phần tử nhỏ nhất nằm ở vị trí nào. Cho biết hàng nào có tổng các phần tử là lớn nhất. Tính giá trị trung bình của tất cả các phần tử dương có trong ma trận.

Sinh viên tự làm

9. Viết chương trình nhập vào 2 ma trận gồm m hàng, n cột, các phần tử kiểu số nguyên. Tạo ra ma trận thứ 3 là ma trận tổng của 2 ma trận vừa nhập. Tạo ra ma trận thứ 4 là ma trận hiệu của 2 ma trận vừa nhập.

Giải thuật:

Bước 1: Khai báo 2 ma trận (a và b) kiểu int

Bước 2: Nhập và kiểm tra số hàng (m) và số cột (n) cho 2 ma trận

Bước 3: Nhập ma trận (a)

```
For (i = 0 .. m-1) Do
    For (j = 0 .. n-1) Do
        cin>>a[i][j];
```

Bước 4: Nhập ma trận (b)

```
For (i = 0 .. m-1) Do
    For (j = 0 .. n-1) Do
```

```
cin>>b[i][j];
```

Bước 5: Tạo ma trận tổng (c)

```
For (i = 0 .. m-1) Do
```

```
    For (j = 0 .. n-1) Do
```

```
        c[i][j] = a[i][j] + b[i][j];
```

Bước 6: Tạo ma trận hiệu (d)

```
For (i = 0 .. m-1) Do
```

```
    For (j = 0 .. n-1) Do
```

```
        d[i][j] = a[i][j] - b[i][j];
```

Bước 7: In ma trận tổng (c)

```
For (i = 0 .. m-1) Do
```

```
{
```

```
    For (j = 0 .. n-1) Do
```

```
        cout<<c[i][j];
```

```
    cout<<endl;
```

```
}
```

Bước 8: In ma trận hiệu (d)

```
For (i = 0 .. m-1) Do
```

```
{
```

```
    For (j = 0 .. n-1) Do
```

```
        cout<<d[i][j];
```

```
    cout<<endl;
```

```
}
```

Bước 9: Kết thúc

10. Viết chương trình nhập vào 2 ma trận các phần tử kiểu số thực. Kiểm tra xem hai ma trận này có thể tính tích được không? Nếu được thì tính tích hai ma trận này và lưu thành một ma trận mới.

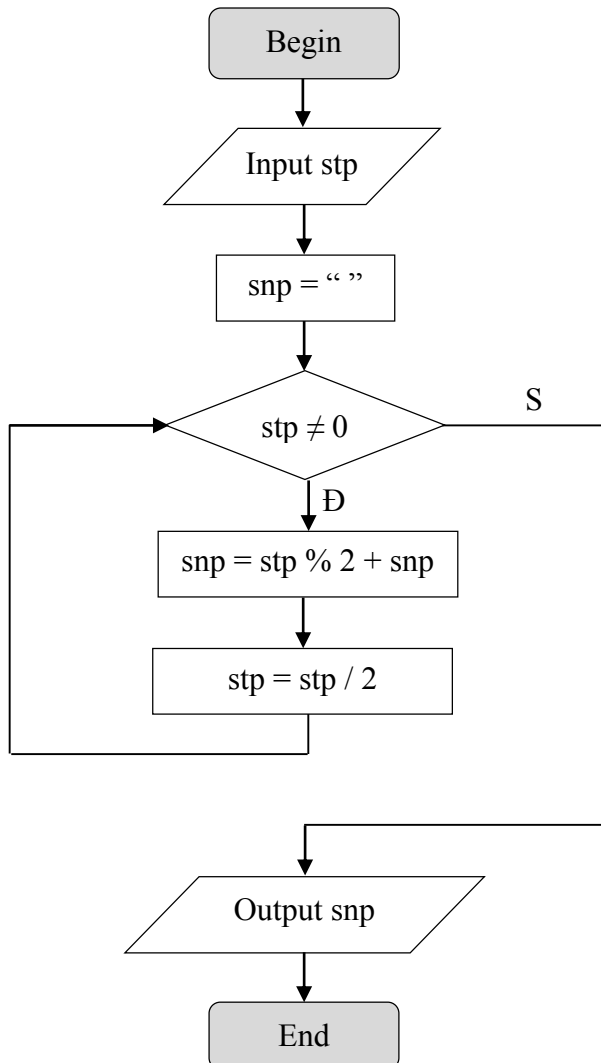
Sinh viên tự làm

CHƯƠNG 4: DỮ LIỆU KIỂU CHUỖI



1. Viết chương trình nhập vào một số nguyên hệ thập phân (cơ số 10), đổi số đó sang hệ nhị phân (cơ số 2).

Giải thuật:



2. Viết chương trình nhập vào một số nguyên hệ nhị phân (cơ số 2), đổi số đó sang hệ thập phân (cơ số 10).

Sinh viên tự làm

3. Viết chương trình nhập vào một số nguyên hệ thập phân (cơ số 10), đổi số đó sang hệ thập lục phân (cơ số 16).

Chương trình:

```
#include <iostream>
#include <string.h>
```

```
using namespace std;
main()
{
    char stlp[20];
    int tam;
    char kt;
    int stp, i=0;
    cout<<"Nhap so nguyen:";
    cin>>stp;
    strcpy(stlp, "");
    while(stp>0)
    {
        tam=stp%16;
        if(tam<=9)
            kt=char(tam+48);
        else
            switch (tam)
            {
                case 10: kt='A'; break;
                case 11: kt='B'; break;
                case 12: kt='C'; break;
                case 13: kt='D'; break;
                case 14: kt='E'; break;
                case 15: kt='F';
            }
        stlp[i]=kt;
        i++;
        stp=stp/16;
    }
    strrev(stlp);
    cout<<"So thap luc phan tuong ung la:"<<stlp<<endl;
}
```


4. Viết chương trình nhập vào một số nguyên hệ bát phân (cơ số 8), đổi số đó sang hệ thập lục phân (cơ số 16).

Sinh viên tự làm

5. Viết chương trình nhập vào một số nguyên hệ nhị phân (cơ số 2), đổi số đó sang hệ bát phân (cơ số 8).

Chương trình:

```
#include <iostream>
#include <string.h>
#include <math.h>
using namespace std;
main()
{
    char snp[20], sbp[20]="", t;
    int stp=0, i, cd;
    cout<<"Nhập số nhị phân:";
    cin>>snp;
    cd=strlen(snp);
    i=0;
    while(i<cd)
    {
        if(snp[i]=='1')
            stp=stp+(int)pow(2, cd-i-1);
        i++;
    }
    i=0;
    while(stp>0)
    {
        t=char(stp%8+48);
        sbp[i]=t;
        i++;
        stp=stp/8;
    }
}
```

```
    strrev(sbp) ;  
    cout<<"So bat phan tuong ung la:"<<sbp<<endl;  
}
```

6. Viết chương trình nhập vào một chuỗi, in ra chuỗi đảo ngược theo từng ký tự.

Ví dụ: Nhập chuỗi ‘ABCD’ \Rightarrow Chuỗi đảo là: ‘DCBA’

Sinh viên tự làm

7. Viết chương trình nhập vào một chuỗi, in ra chuỗi đảo ngược theo từng từ.

Ví dụ: Nhập chuỗi ‘Cấm Không Được Câu Cá’

\Rightarrow Chuỗi đảo là: ‘Cá Câu Được Không Cấm’

Giải thuật:

Bước 1: Khai báo biến chuỗi (st) chứa chuỗi nhập, biến (tam) chứa chuỗi tạm và biến (tu) chứa từng từ trong chuỗi

Bước 2: Nhập chuỗi (st)

Bước 3: Ghép 1 khoảng trắng vào đầu chuỗi (st)

Bước 4: Gán tam = chuỗi rỗng

Bước 5: j = vị trí của ký tự cuối cùng trong chuỗi (st)

Bước 6: Trong khi (j > 0 và st[j] != " ") thì j--

Bước 7: i = j

Bước 8: Trong khi (i > 0 và st[i] != " ") thì i--

Bước 9: Gán tu = chuỗi rỗng

Bước 10: Copy (j – i) ký tự trong chuỗi (st) từ vị trí thứ i + 1 vào biến (tu)

Bước 11: Ghép 1 khoảng trắng vào cuối biến (tu)

Bước 12: Ghép giá trị biến (tu) vào cuối chuỗi (tam)

Bước 13: Gán j = i - 1

Bước 14: Lặp lại bước 6 cho đến khi j = 0

Bước 15: In ra chuỗi (tam)

Bước 16: Kết thúc

8. Viết chương trình nhập vào một số nguyên dương nhỏ hơn 1000, đọc số đó ra chữ.

Ví dụ: Nhập số 125 \Rightarrow Đọc là ‘Một trăm hai mươi lăm’

Sinh viên tự làm

9. Viết chương trình nhập vào một chuỗi ký số (các ký tự từ 1 đến 9), hãy cho biết trong chuỗi có bao nhiêu dãy con tăng dần, in ra dãy con tăng dần dài nhất có trong chuỗi này.

Chương trình:

```
#include <iostream>
#include <string.h>
using namespace std;
main()
{
    char st[100], stc[100];
    int a[100], i, j, k, cd, max;
    cout<<"Nhap chuoi:";
    cin.getline(st, 100);
    cout<<"Chuoi nhap la:"<<st<<endl;
    for(i=0; i<100; i++)
        a[i]=0;
    cd=strlen(st);
    i=j=k=0;
    while(i<cd-1)
        if(st[i]<=st[i+1]) i++;
        else
        {
            a[k]=i-j+1;
            i++;
            j=i;
            k++;
        }
    a[k]=i-j+1;
    for(i=0; i<=k; i++)
        cout<<a[i]<<'\\t';
    cout<<"\\n Trong chuoi co "<<k+1<<" day con tang dan";
    max=a[0];
    for(i=1; i<=k; i++)
        if(max<a[i]) max=a[i];
    cout<<"\\n max = "<<max<<endl;
```

```
cout<<"Cac day con tang dan dai nhat la:"<<endl;
i=j=0;
while(i<cd-1)
    if(st[i]<=st[i+1]) i++;
    else
    {
        if(i-j+1==max)
        {
            strcpy(stc,"");
            strncpy(stc,st+j,i-j+1);
            stc[i-j+1]='\0';
            cout<<stc<<endl;
        }
        i++;
        j=i;
    }
if(i-j+1==max)
{
    strcpy(stc,"");
    strncpy(stc,st+j,i-j+1);
    stc[i-j+1]='\0';
    cout<<stc<<endl;
}
}
```

10. Viết chương trình nhập vào một chuỗi bất kỳ, cho biết từ nào xuất hiện nhiều nhất.

Sinh viên tự làm

CHƯƠNG 5: CON TRỎ VÀ HÀM



1. Viết chương trình nhập vào chuỗi ký tự st (dạng con trỏ), in ra màn hình chuỗi đảo ngược.

Chương trình:

```
#include <iostream>
#include <string.h>
using namespace std;
char *st=new char[100];
main()
{
    cout<<"Nhap chuoi bat ky: ";
    gets(st);
    cout<<"Chuoi dao nguoc la: ";
    for(int i=strlen(st)-1;i>=0;i--)
        cout<<st[i];
}
```

2. Viết chương trình nhập vào chuỗi ký tự s(dạng con trỏ), hãy copy từ chuỗi s sang chuỗi t một đoạn bắt đầu tại vị trí m với độ dài n.

Sinh viên tự làm

3. Xây dựng hàm tìm UCLN của 2 số nguyên. Áp dụng hàm này để tìm UCLN của 3 số nguyên được nhập từ bàn phím.

Chương trình:

```
#include <iostream>
using namespace std;
int UCLN(int a, int b)
{
    int t;
    if(a<b) t=a;
    else t=b;
    while(a%t!=0 || b%t!=0)
        t--;
}
```

```
        return t;
    }
main()
{
    int a,b,c;
    cout<<"Nhap 2 so nguyen a, b:";
    cin>>a>>b;
    cout<<"UCLN ("<<a<<" , "<<b<<" )="<<UCLN (a,b)<<endl;
    cout<<"Nhap so nguyen c:";
    cin>>c;
    cout<<"UCLN ("<<a<<" , "<<b<<" , "<<c<<" )=";
    cout<<UCLN (UCLN (a,b) , c)<<endl;
}
```

4. Xây dựng hàm tìm BCNN của 2 số nguyên. Áp dụng hàm này để tìm UCLN của 4 số nguyên được nhập từ bàn phím.

Sinh viên tự làm

5. Xây dựng hàm kiểm tra một số nguyên dương n có là số nguyên tố. Áp dụng hàm này để tính tổng của K số nguyên tố đầu tiên, với K được nhập từ bàn phím.

Chương trình:

```
#include <iostream>
using namespace std;
int NguyenTo(int n)
{
    int i=1;
    do{
        i++;
    }while(i<=n/2 && n%i!=0);
    if(i>n/2) return 1;
    else return 0;
}
main()
{
```

```
int k;
cout<<"Ban muon tinh tong bao nhieu SNT? ";
cin>>k;
int i=2, d=0;
long s=0;
while (d<k)
{
    if (NguyenTo(i))
    {
        s=s+i;
        d++;
    }
    i++;
}
cout<<"Tong "<<k<<" so nguyen to dau tien la: ";
}
```

6. Xây dựng hàm kiểm tra một năm có là năm nhuận không. Áp dụng hàm này in ra các năm nhuận từ năm 1000 đến 2035.

Sinh viên tự làm

7. Xây dựng hàm chuẩn hoá một chuỗi bất kỳ về dạng Proper (cắt bỏ các khoảng trắng ở 2 đầu, cắt bớt các dấu trắng thừa ở giữa các từ (chỉ để lại 1), viết hoa đầu từ).

Sinh viên tự làm

8. Xây dựng hàm đệ qui tính $n!$. Áp dụng hàm này để tính tổ hợp n chập k theo công thức truy hồi: $C(n, k) = n! / (k! (n-k)!)$. Viết hàm `main()` minh họa tính đúng đắn của các hàm đã xây dựng.

Chương trình:

```
#include <iostream>
using namespace std;
long GT(int n)
{
    if(n==0) return 1;
    else return n*GT(n-1);
}
```

```
float TH(int n, int k)
{
    if(k==0 || k==n) return 1;
    else return GT(n) / (GT(k)*GT(n-k));
}
main()
{
    int n,k;
    cout << "Nhap n va k: ";
    cin >> n >> k;
    cout << "To hop "<<n<<" chap "<<k<<" la: " <<TH(n,k);
}
```

9. Xây dựng các hàm sau bằng giải thuật không đệ qui:

a. Kiểm tra xem số n có phải là số chính phương không?

✧ **Xác định kiểu trả về, các tham số và cách truyền tham số của hàm:**

- Kiểu trả về: bool/int
- Các tham số: int n

✧ **Giải thuật:**

```
float cb2n = sqrt(n);
if(cb2n == (int) cb2n)
    return true;
else
    return false;
```

b. Tìm số lớn nhất trong 4 số thực.

Sinh viên tự làm

c. In ra n phần tử đầu tiên của dãy Fibonacci.

✧ **Xác định kiểu trả về, các tham số và cách truyền tham số của hàm:**

- Kiểu trả về: void
- Các tham số: int n

✧ **Giải thuật:**

```
if(n == 1) cout << 1;
else
```



```

{
    f1 = f2 = 1;
    cout << 1 << '\t' << 1;
    for(i = 3, i <= n; i++)
    {
        f = f1 + f2;
        cout << '\t' << f;
        f1 = f2; f2 = f;
    }
}

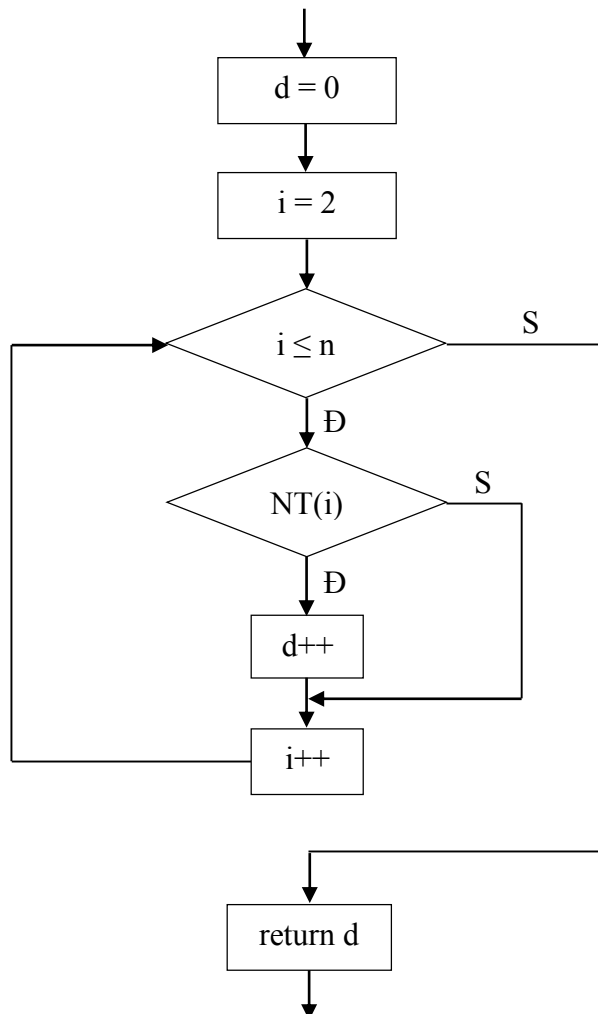
```

d. Đếm xem có bao nhiêu số nguyên tố trong n số nguyên dương đầu tiên.

✧ **Xác định kiểu trả về, các tham số và cách truyền tham số của hàm:**

- Kiểu trả về: int
- Các tham số: int n

✧ **Giải thuật:**



e. In ra các ước số của số nguyên dương n.

Sinh viên tự làm

Viết hàm main() minh họa tính đúng đắn của các hàm trên.

10. Xây dựng các hàm sau bằng giải thuật đệ qui:

a. Tính tổng N số nguyên dương đầu tiên.

Sinh viên tự làm

b. Tính x^n với x là số thực còn n là số nguyên.

```
double LuyThua(float x, int n)
{
    if(n == 0)
        return 1;
    else
        return x * LuyThua(x, n-1);
}
```

c. Tìm bội số chung nhỏ nhất của 2 số nguyên dương.

Cách 1:

```
long BCNN(int a, int b, int i)
{
    if(a%b==0) return a;
    else
        if(b%a==0) return b;
    else
        if(a>b) return BCNN(a+a/i,b,i+1);
        else BCNN(a,b+b/i,i+1);
}
```

Cách 2:

```
long BCNN2(int a, int b, int i)
{
    if(a==b) return a;
    else
    {
        long t=a*i;
```

```

        if (t%b==0) return t;
        else return BCNN2(a,b,i+1);
    }
}

```

d. In ra các ước số của số nguyên dương n.

Sinh viên tự làm

Viết hàm main() minh họa tính đúng đắn của các hàm trên.

11. Xây dựng các hàm sau cho dữ liệu kiểu mảng 1 chiều có n phần tử kiểu số nguyên:

- a. Nhập mảng.
- b. Xuất mảng.
- c. Đếm số phần tử là số nguyên tố có trong mảng.
- d. Tính tổng các phần tử là ước số của K.
- e. Tìm BCNN của 2 phần tử đầu và cuối mảng.
- f. Sắp xếp mảng theo thứ tự tăng dần.

Viết hàm main() minh họa tính đúng đắn của các hàm trên.

Chương trình:

```

#include <iostream>
using namespace std;

//a. Ham nhap mang
void NhapMang(int a[], int n)
{
    for(int i=0; i<n; i++)
    {
        cout<<"Nhap a["<<i<<"]= ";
        cin>>a[i];
    }
}

//b. Ham xuat mang
void XuatMang(int a[], int n)
{
    for(int i=0; i<n; i++)
        cout<<a[i]<<'\\t';

```

```
}
```

```
//c1. Ham kiem tra so nguyen to
```

```
int NguyenTo(int n)
```

```
{
```

```
    int i=1;
```

```
    do{
```

```
        i++;
```

```
    }while(i<=n/2 && n%i!=0);
```

```
    if(i>n/2)
```

```
        return 1;
```

```
    else
```

```
        return 0;
```

```
}
```

```
//c2. Ham dem so nguyen to co trong mang
```

```
int DemNT(int a[], int n)
```

```
{
```

```
    int dem=0;
```

```
    for(int i=0; i<n; i++)
```

```
        if(a[i]>1 && NguyenTo(a[i])) dem++;
```

```
    return dem;
```

```
}
```

```
//d. Ham tinh tong cac pt la uoc so cua k
```

```
long TongUocK(int a[], int n, int k)
```

```
{
```

```
    long tong=0;
```

```
    for(int i=0; i<n; i++)
```

```
        if(k%a[i]==0) tong=tong+a[i];
```

```
    return tong;
```

```
}
```

```
//e1. Ham tim boi chung nho nhat cua 2 so
```

```
long BCNN(int a, int b, int i)
```

```
{
```

```
    if(a==b) return a;
    else
    {
        long t=a*i;
        if(t%b==0) return t;
        else return BCNN(a,b,i+1);
    }
}

//e2. Ham tim boi chung nho nhat cua 2 pt dau va cuoi
long BCNN_DC(int a[], int n)
{
    return BCNN(a[0],a[n-1],1);
}

//f. Ham sap xep mang tang dan
void SxTangDq(int a[], int n, int i)
{
    int j, tam;
    if(i<n-1)
    {
        for(j=i+1; j<n; j++)
            if(a[i]>a[j])
            {
                tam=a[i]; a[i]=a[j]; a[j]=tam;
            }
        SxTangDq(a,n,i+1);
    }
}

//Ham main() minh hoa
main()
{
    int a[20], n, k;
    cout<<"Nhap n = ";
```

```

    cin>>n;
   NhapMang(a,n);
    cout<<"Mang vua nhap la:"<<endl;
    XuatMang(a,n);
    cout<<endl;
    cout<<"Trong mang co "<<DemNT(a,n);
    cout<<" la so nguyen to"<<endl;
    cout<<"Nhap k = ";
    cin>>k;
    cout<<"Tong cac pt la uoc so cua "<<k;
    cout<<" la: "<<TongUocK(a,n,k)<<endl;
    cout<<"BCNN cua pt dau va cuoi mang la: " ;
    cout<<BCNN_DC(a,n)<<endl;
    SxTangDq(a,n,0);
    cout<<"Mang sau khi sap xep tang dan la:"<<endl;
    XuatMang(a,n);
}

```

12. Xây dựng các hàm sau cho dữ liệu kiểu ma trận M hàng, N cột, phần tử kiểu số thực:

- Nhập ma trận.
- In ma trận.
- Đếm số phần tử có phần nguyên là 0.
- Tính tổng các phần tử nằm trên hàng H.
- Cho biết vị trí của phần tử nhỏ nhất.
- Sắp xếp các phần tử trên cột C theo thứ tự giảm dần.

Viết hàm main() minh họa tính đúng đắn của các hàm trên.

Sinh viên tự làm

13. Xây dựng các hàm thực hiện các chức năng sau:

- Trả về chiều dài của chuỗi.

✧ **Xác định kiểu trả về, các tham số và cách truyền tham số của hàm:**

- Kiểu trả về: int
- Các tham số: char* st

✧ **Giải thuật:**

```
int i = 0, d = 0;
while(st[i] != '\0')
{
    d++;
    i++;
}
return d;
```

b. Trả về vị trí của chuỗi stc trong chuỗi st.

✧ **Xác định kiểu trả về, các tham số và cách truyền tham số của hàm:**

- *Kiểu trả về:* int
- *Các tham số:* char* stc, char* st

✧ **Giải thuật:**

```
int i = 0, cdstc, cdst;
char* tam = new char;
cdstc = Chieudai(stc);
cdst = Chieudai(st);
bool kt = false;
while(kt == false && i <= cdst - cdstc)
{
    strncpy(tam, st + i, cdstc);
    tam[cdstc] = '\0';
    if(strcmp(tam, stc) == 0) kt = true;
    else i++;
}
if(kt == true) return i;
else return - 1;
```

c. Trả về chuỗi stc từ chuỗi st gồm num ký tự bắt đầu từ vị trí thứ pos.

✧ **Xác định kiểu trả về, các tham số và cách truyền tham số của hàm:**

- *Kiểu trả về:* char*
- *Các tham số:* char* st, int num, int pos

✧ ***Giải thuật:***

```
char* tam = new char;  
strcpy(tam, "");  
int n = Chieudai(st) - pos;  
if(num > n) num = n;  
for(int i = 0; i < num; i++)  
    tam[i] = st[pos + i];  
tam[num] = '\0';  
return tam;
```

d. Trả về chuỗi sau khi ghép 3 chuỗi lại với nhau.

✧ ***Xác định kiểu trả về, các tham số và cách truyền tham số của hàm:***

- *Kiểu trả về:* char*
- *Các tham số:* char* st1, char* st2, char* st3

✧ ***Giải thuật:***

```
char* tam = new char;  
strcpy(tam, "");  
int cd1 = Chieudai(st1);  
int cd2 = Chieudai(st2);  
int cd3 = Chieudai(st3);  
int i;  
for(i = 0; i < cd1; i++)  
    tam[i] = st1[i];  
for(i = 0; i < cd2; i++)  
    tam[cd1 + i] = st2[i];  
for(i = 0; i < cd3; i++)  
    tam[cd1 + cd2 + i] = st3[i];  
tam[cd1 + cd2 + cd3] = '\0';  
return tam;
```

Viết hàm main() minh họa tính đúng đắn của các hàm trên.

14. Xây dựng các hàm thực hiện các chức năng sau:

- Chèn chuỗi stc vào chuỗi st bắt đầu từ vị trí thứ pos.
- Xóa num ký tự trong chuỗi st bắt đầu từ vị trí thứ pos.

- c. Đổi số nguyên thành chuỗi số tương ứng.
- d. Đổi số thực thành chuỗi số tương ứng.
- e. Đổi chuỗi số nguyên thành số tương ứng.
- f. Đổi chuỗi số thực thành số tương ứng.

Viết hàm main() minh họa tính đúng đắn của các hàm trên.

Sinh viên tự làm

CHƯƠNG 6: DỮ LIỆU KIỂU CẤU TRÚC



1. Một phân số gồm 2 trường là tử số và mẫu số. Viết chương trình nhập vào 2 phân số. In ra kết quả của các phép toán cộng, trừ, nhân và chia trên 2 phân số đó.

Giải thuật:

Bước 1: Khai báo cấu trúc Phanso gồm 2 trường tu và mau

Bước 2: Xây dựng hàm tìm ước chung lớn nhất của 2 số nguyên

Bước 3: Xây dựng hàm rút gọn một phân số

Bước 4: Xây dựng hàm nhập một phân số

Bước 5: Xây dựng hàm xuất một phân số

Bước 6: Viết hàm main() có sử dụng khai báo và các hàm đã xây dựng

Bước 6.1: Khai báo các biến a, b, tong, hieu, tich, thuong, kq kiểu Phanso

Bước 6.2: Nhập 2 phân số a và b

Bước 6.3: Xuất 2 phân số a và b đã nhập

Bước 6.4: Tính tổng 2 phân số a và b

Bước 6.4.1: $tong.tu = a.tu * b.mau + a.mau * b.tu$

Bước 6.4.2: $tong.mau = a.mau * b.mau$

Bước 6.4.3: Rút gọn phân số tong và gán vào phân số kq

Bước 6.4.4: Xuất phân số kq chính là phân số tổng của a và b

Bước 6.5: Tính hiệu 2 phân số a và b

Bước 6.5.1: $hieu.tu = a.tu * b.mau - a.mau * b.tu$

Bước 6.5.2: $hieu.mau = a.mau * b.mau$

Bước 6.5.3: Rút gọn phân số hieu và gán vào phân số kq

Bước 6.5.4: Xuất phân số kq chính là phân số hiệu của a và b

Bước 6.6: Tính tích 2 phân số a và b

Bước 6.6.1: $tich.tu = a.tu * b.tu$

Bước 6.6.2: $tich.mau = a.mau * b.mau$

Bước 6.6.3: Rút gọn phân số tich và gán vào phân số kq

Bước 6.6.4: Xuất phân số kq chính là phân số tích của a và b

Bước 6.7: Tính thương 2 phân số a và b

Bước 6.7.1: $thuong.tu = a.tu * b.mau$

Bước 6.7.2: $thuong.mau = a.mau * b.tu$

Bước 6.7.3: Rút gọn phân số thương và gán vào phân số kq

Bước 6.7.4: Xuất phân số kq chính là phân số thương của a và b

Bước 6.8: Kết thúc

2. Giả sử để quản lý thông tin về họ tên, ngày sinh (phải bao gồm ngày, tháng, năm sinh) người ta định nghĩa 2 cấu trúc là NgaySinh và SinhVien. Từ các cấu trúc đó hãy viết chương trình nhập N sinh viên, in thông tin về các sinh viên, cho biết sinh viên nào sinh vào ngày giữa 15, đếm số sinh viên sinh vào mùa xuân (tháng 1, 2, 3), in họ tên sinh viên có tuổi đời trẻ nhất.

Sinh viên tự làm

3. Viết chương trình quản lý học sinh của một lớp. Mỗi học sinh gồm các thông tin: họ tên, năm sinh, điểm HK1, điểm HK2. In ra họ tên, năm sinh và điểm cả năm của tất cả các học sinh, với điểm cả năm = (điểm HK1 + điểm HK2)/2. In ra tuổi lớn nhất và nhỏ nhất của lớp. Cho biết học sinh nào có điểm cả năm lớn nhất. Tìm điểm trung bình cả năm của lớp. Sắp xếp danh sách giảm dần theo điểm cả năm, nếu trùng thì sắp xếp giảm dần theo điểm HK2. In ra danh sách sau khi sắp xếp.

Chương trình:

```
#include <iostream>
using namespace std;
struct hocsinh
{
    char ht[40];
    int ns;
    float dk1, dk2;
} hs[40];
main()
{
    int n,i,j;
    hocsinh tam;
    //Nhập và kiểm tra số lượng học sinh
    do
    {
        cout<<"Nhập số lượng học sinh : ";
```

```
    cin>>n;
}while (n <=0 || n > 40);
//Nhập sách học sinh
for (i=0; i<n; i++)
{
    cout<<"Nhập học sinh thu "<<i+1<<" : "<<endl;
    cin.ignore(1);
    cout<<"Nhập họ tên:"; cin.getline(hs[i].ht,40);
    cout<<"Nhập nam sinh:"; cin>>hs[i].ns;
    cout<<"Nhập điểm học kỳ 1:"; cin>>hs[i].dk1;
    cout<<"Nhập điểm học kỳ 2:"; cin>>hs[i].dk2;
}
//In danh sách học sinh đã nhập
for (i=0; i<n; i++)
{
    cout<<"Thông tin về học sinh thu "<<i+1;
    cout<<" là : "<<endl;
    cout<<"Họ tên : "<<hs[i].ht<<endl;
    cout<<"Nam sinh : "<<hs[i].ns<<endl;
    cout<<"Điểm cả năm : ";
    cout<<(hs[i].dk1+hs[i].dk2)/2<<endl;
}
//In ra tuổi lớn nhất và nhỏ nhất của lớp
int nsmax, nsmin;
nsmax=hs[0].ns;
nsmin=hs[0].ns;
for (i=1; i<n; i++)
{
    if (nsmax<hs[i].ns) nsmax=hs[i].ns;
    if (nsmin>hs[i].ns) nsmin=hs[i].ns;
```

```
}  
cout<<"Tuoi lon nhat cua lop la : "<<2016-nsmin<<endl;  
cout<<"Tuoi nho nhat cua lop la : "<<2016-nsmax<<endl;  
//Tim hoc sinh co diem ca nam lon nhat  
float dcnmax=(hs[0].dk1+hs[0].dk2)/2;  
for (i=1; i<n; i++)  
    if (dcnmax<(hs[i].dk1+hs[i].dk2)/2)  
        dcnmax=(hs[i].dk1+hs[i].dk2)/2;  
cout<<"Danh sach hoc sinh co diem ca nam lon nhat:\n";  
for (i=0; i<n; i++)  
    if ((hs[i].dk1+hs[i].dk2)/2==dcnmax)  
        cout<<hs[i].ht<<endl;  
//Tim diem trung binh ca nam cua lop  
float td=0;  
for (i=0; i<n; i++)  
    td=td+(hs[i].dk1+hs[i].dk2)/2;  
cout<<"Diem trung binh ca nam cua lop = "<<td/n;  
//Sap xep danh sach giam dam theo diem ca nam, diem hk2  
for (i=0; i<n-1; i++)  
    for (j=i+1; j<n; j++)  
        if(((hs[i].dk1+hs[i].dk2)/2<(hs[j].dk1+hs[j].dk2)/2)  
            ||(((hs[i].dk1+hs[i].dk2)/2==(hs[j].dk1+hs[j].dk2)/2)  
            &&(hs[i].dk2<hs[j].dk2)))  
        {  
            tam=hs[i];  
            hs[i]=hs[j];  
            hs[j]=tam;  
        }  
//In ra danh sach sau khi sap xep  
cout<<"Danh sach hs sau khi sap xep la : "<<endl;
```

```
for (i=0; i<n; i++)  
{  
    cout<<"Thong tin ve hoc sinh thu "<<i+1;  
    cout<<" la : "<<endl;  
    cout<<"Ho ten : "<<hs[i].ht<<endl;  
    cout<<"Nam sinh : "<<hs[i].ns<<endl;  
    cout<<"Diem ca nam : ";  
    cout<<(hs[i].dk1+hs[i].dk2)/2<<endl;  
}  
}
```

4. Viết chương trình nhập vào N hóa đơn bán hàng, N không quá 100. Mỗi hóa đơn gồm các thông tin: mã số hóa đơn, tên người mua, tên hàng, loại hàng, đơn giá và số lượng. Khi nhập cần kiểm tra các điều kiện sau: mã số phải đủ 4 ký tự trong đó 2 ký tự đầu là chữ cái và 2 ký tự cuối là chữ số, tên người mua chỉ gồm chữ cái và khoảng trắng nhưng không toàn là khoảng trắng và sau khi nhập đổi sang dạng chuẩn proper, tên hàng chỉ gồm chữ cái và chữ số nhưng phải có ít nhất là 2 ký tự và sau khi nhập đổi sang dạng chuẩn upper, loại hàng chỉ nhận một trong ba giá trị là 1, 2, hoặc 3, đơn giá và số lượng phải dương. Cho biết thứ tự của hóa đơn có thành tiền cao nhất với thành tiền = số lượng * đơn giá. Cho biết số lượng trung bình của tất cả các hóa đơn. Tính tổng thành tiền bán được theo từng loại hàng.

Sinh viên tự làm

5. Viết chương trình quản lý nhân viên trong một cơ quan, mỗi nhân viên thông tin: mã số, họ tên, quê quán, ngày sinh, chức vụ, phụ cấp chức vụ, hệ số lương, tạm ứng và thực lĩnh. Chương trình cần thực hiện được các chức năng sau:

- Nhập danh sách nhân viên có kiểm tra dữ liệu nhập
 - + Mã số gồm 6 ký số nhưng phải được tạo thành ít nhất từ 2 ký số khác nhau.
 - + Họ tên chỉ gồm chữ cái và khoảng trắng nhưng không toàn là khoảng trắng và sau khi nhập đổi sang dạng chuẩn upper.
 - + Hệ số lương từ 2.1 đến 6.8, tạm ứng phải không âm.
- In danh sách nhân viên.
- Tính thực lĩnh cho các nhân viên,
với thực lĩnh = hệ số lương * 1.210.000 + phụ cấp chức vụ - tạm ứng.
- In họ tên nhân viên có hệ số lương cao nhất.

- Đếm số nhân viên có thực lĩnh trên 5000000.
- Tính tổng tạm ứng của tất cả những nhân viên có chức vụ được nhập từ bàn phím.
- Sắp xếp giảm dần theo thực lĩnh.
- In ra danh sách nhân viên sau khi sắp xếp.

Phần khai báo:

- Khai báo kiểu cấu trúc Date
- Khai báo kiểu cấu trúc Nhanvien có trường Ngaysinh kiểu Date
- Khai báo mảng 1 chiều NV có kiểu phần tử là Nhanvien
- Khai báo các biến cần thiết

Phần chương trình:

- Nhập và kiểm tra số lượng nhân viên (N).
- Nhập danh sách nhân viên (NV): Sử dụng cấu trúc lặp for; trong quá trình nhập thông tin phải có kiểm tra dữ liệu theo yêu cầu.
- In danh sách nhân viên (NV): Sử dụng cấu trúc lặp for.
- Tính thực lĩnh:

```
for(i = 0; i < N; i++)
    NV[i].tl = NV[i].hsl * 1210000 + NV[i].pc - NV[i].tu;
```
- In họ tên nhân viên có hệ số lương cao nhất:

```
max = NV[0].hsl;
for(i = 1; i < N; i++)
    if(max < NV[i].hsl) max = NV[i].hsl;
for(i = 0; i < N; i++)
    if(NV[i].hsl == max)
        cout<<NV[i].ht<<endl;
```
- Đếm số nhân viên có thực lĩnh trên 5000000:

```
d = 0;
for(i = 0; i < N; i++)
    if(NV[i].tl > 5000000) d++;
cout<<"Co "<<d<<" nhan vien co thuc linh tren 5000000"<<endl;
```
- Tính tổng tạm ứng của tất cả những nhân viên có chức vụ được nhập từ bàn phím: Tương tự như đoạn đếm số nhân viên nhưng nếu thỏa điều kiện thì thực hiện lệnh: $t = t + NV[i].tu$;
- Sắp xếp giảm dần theo thực lĩnh:
 Sử dụng 2 cấu trúc lặp for lồng nhau; nếu 2 nhân viên thứ i và j không đúng vị trí thì đổi chỗ 2 nhân viên này.
- In ra danh sách nhân viên sau khi sắp xếp:
 Tương tự như đoạn in danh sách nhân viên sau khi nhập.

CHƯƠNG 7: DỮ LIỆU KIỂU TẬP TIN

1. Viết chương trình in nội dung của một file văn bản và cho biết file đó có bao nhiêu dòng.

Chương trình:

```
#include <iostream>
#include <fstream>
#include <conio.h>
using namespace std;
main()
{
    char fname[20];
    cout << "Nhap ten file can mo: ";
    cin.getline(fname, 20);
    ifstream f(fname);
    if (!f.good())
    {
        cout << "File " << fname << " chua ton tai!";
        getch();
        exit(1);
    }
    int d=0;
    char tam[80];
    while(!f.eof())
    {
        f.getline(tam, 80);
        cout<<tam<<endl;
        d++;
    }
    f.close();
    cout<<"\n Trong file "<<fname<<" co "<<d<<" dong.";
```


2. Viết chương trình đọc in từng kí tự của file văn bản ra màn hình, mỗi màn hình in 20 dòng.

Sinh viên tự làm

3. Viết chương trình tìm từ dài nhất trong một file văn bản.

Giải thuật:

Bước 1: Mở file văn bản ở chế độ đọc.

Bước 2: Đọc từng ký tự từ đầu đến cuối và sử dụng 4 biến tạm (bd, kr, cd, max) để xác định từ dài nhất có bao nhiêu ký tự (*trong đó: biến bd xác định vị trí ký tự đầu của 1 từ, biến kt xác định vị trí ký tự cuối của 1 từ, biến cd xác định chiều dài của 1 từ ($cd = kt - bd + 1$) và biến max xác định cd của từ dài nhất tạm thời*); khi xét hết file thì max chính là chiều dài của từ dài nhất thật sự có trong văn bản.

Bước 3: Đưa con trỏ tập tin về đầu.

Bước 4: Thực hiện tương tự bước 2 và từ nào có $cd = max$ thì in từ đó ra.

Bước 5: Đóng tập tin.

Bước 6: Kết thúc.

4. Viết chương trình ghép một file văn bản thứ hai vào file văn bản thứ nhất, trong đó tất cả chữ cái của file văn bản thứ nhất phải đổi thành chữ in hoa.

Sinh viên tự làm

5. Viết chương trình in nội dung file ra màn hình và cho biết tổng số chữ cái, tổng số chữ số đã xuất hiện trong file.

Chương trình:

```
#include <iostream>
#include <fstream>
#include <conio.h>
using namespace std;
main()
{
    char fname[20], ch;
    cout<<"Nhập tên file cần mở: ";
    cin.getline(fname, 20);
    ifstream f(fname, ios::binary);
    if (!f.good())
```

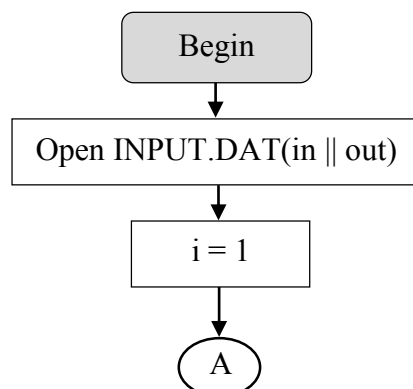
```
{  
    cout<<"File "<<fname<<" chưa tồn tại!";  
    getch();  
    exit(1);  
}  
int dc=0, ds=0;  
while (!f.eof())  
{  
    f.get(ch);  
    cout<<ch;  
    if(isalpha(ch)) dc++;  
    else if(isdigit(ch)) ds++;  
}  
f.close();  
cout<<"\nSố chữ cái có trong file "<<fname<<" là: "<<dc;  
cout<<"\nSố chữ số có trong file "<<fname<<" là: "<<ds;  
}
```

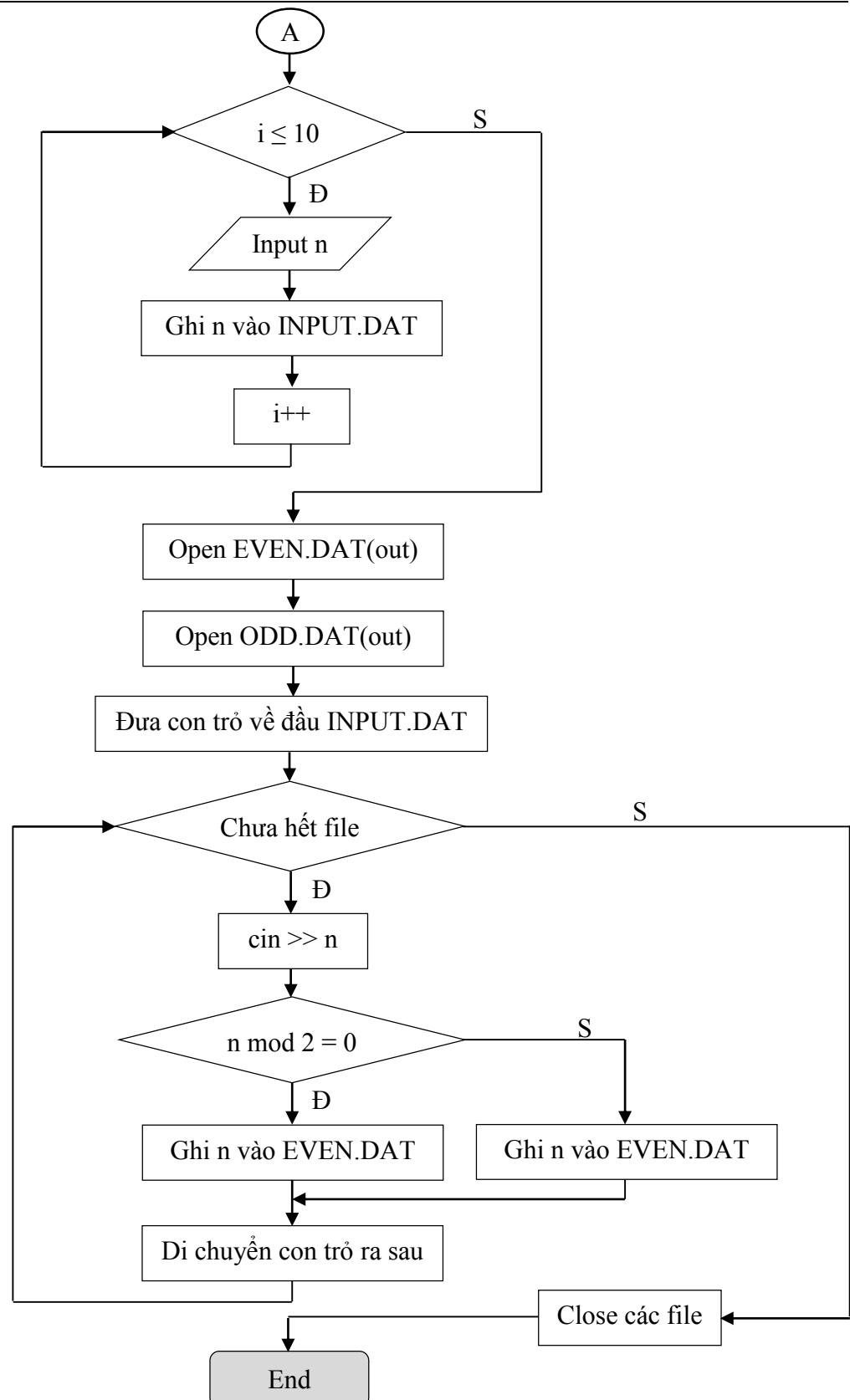
6. Viết chương trình nhập 10 số thực từ bàn phím vào file INPUT.DAT. Sau đó đọc các số thực từ file trên và in tổng bình phương của chúng ra màn hình.

Sinh viên tự làm

7. Viết chương trình nhập 10 số nguyên từ bàn phím vào file văn bản tên INPUT.DAT. Sau đó đọc các số nguyên từ file trên và ghi những số chẵn vào file EVEN.DAT còn các số lẻ vào file ODD.DAT.

Giải thuật:





8. Thông tin về một nhân viên trong cơ quan bao gồm: họ tên, chức vụ, số điện thoại, địa chỉ nhà riêng. Viết chương trình nhập từ bàn phím thông tin của N nhân viên và ghi vào file INPUT.DAT. Sau đó tìm và in ra thông tin nhân viên theo số điện thoại được nhập từ bàn phím.

Sinh viên tự làm

PHỤ LỤC 1: NGUYÊN TẮC CHUNG KHI SỬA LỖI



Khi biên dịch chương trình thông thường ta nhận được 2 thông báo là error (lỗi) và warning (cảnh báo). Nguyên tắc chung là khi gặp lỗi thì người lập trình bắt buộc phải sửa lại cho đúng, còn khi gặp cảnh báo thì có thể bỏ qua. Tuy nhiên một số cảnh báo nếu không được sửa sẽ làm cho chương trình chạy không đúng.

Khi gặp thông báo lỗi thì ta nên tiến hành theo các bước sau để sửa lỗi:

1. Nhấp đúp chuột vào thông báo lỗi để nhảy đến vị trí có lỗi trong chương trình.
2. Đọc dòng chứa con trỏ để sửa lỗi.
3. Nếu ngay dòng chứa con trỏ không phát hiện lỗi thì đọc dòng ngay trên hoặc ngay dưới để tìm lỗi và sửa.
4. Nếu vẫn không phát hiện lỗi thì phải dò lỗi từ đầu chương trình đến dòng chứa con trỏ (vì có thể là do lỗi xuất hiện ở các dòng phía trên nữa).

✧ **Lưu ý:** Các lỗi mà ta có thể phát hiện và sửa theo các bước trên được gọi là lỗi cú pháp (syntax error). Tuy nhiên trong lập trình còn một loại lỗi mà trình biên dịch không phát hiện được đó là lỗi ngữ nghĩa hay lỗi giải thuật (chương trình vẫn thực thi nhưng cho kết quả sai). Để xử lý lỗi này thì người lập trình phải xem xét lại toàn bộ giải thuật để tìm ra nơi phát sinh lỗi (làm cho chương trình trả về kết quả sai) và sửa lại cho đúng.

Một số từ tiếng Anh thông dụng trong lập trình:

undeclared: chưa khai báo

unable: không thể

undefined: không xác định

incorrectly: không chính xác/không đúng

symbol: biểu tượng

statement: câu lệnh

parameter/argument: tham số/đối số

declaration: khai báo

terminated: chấm dứt

syntax: cú pháp

constant: hằng số

expression: biểu thức

PHỤ LỤC 2: CÁC THÔNG BÁO LỖI THƯỜNG GẶP



STT	Thông báo lỗi gốc	Ý nghĩa
1	(expected	Thiếu dấu ... Các lỗi này thường xảy ra khi ta sơ sót, dẫn đến thiếu các dấu mở hoặc đóng ngoặc, ...
2) expected	
3	; expected	
4	{ expected	
5	} expected	
6	286/287 instructions not enabled	Tập lệnh của bộ Vi xử lý 80286 và bộ xử lý toán học chưa được kích hoạt. Vào Options/Compiler/Advanced Code generation... để điều chỉnh lại
7	Ambiguity between 'function1' and 'function2'	2 hàm function1 và function2 giống nhau, không thể phân biệt được.
8	Array bounds missing]	Thiếu dấu đóng ngoặc] khi truy xuất đến các phần tử của mảng
9	Array must have at least one element	Khi khai báo mảng phải có ít nhất 1 phần tử. Xảy ra khi khai báo mảng mà SPT tối đa là âm hoặc bằng 0
10	Array size too large	Kích thước của mảng quá lớn, vượt quá dung lượng vùng nhớ quy ước là 64K
11	Bit field cannot be static	Kiểu dữ liệu bit field không thể có kiểu static
12	Bit field too large	Kích thước của bit field quá lớn
13	Bit fields must be signed or unsigned int	Kiểu dữ liệu của bit field phải là số nguyên
14	Bit fields must contain at least one bit	Kích thước của mỗi bit field phải ≥ 1 bit
15	Body already defined for this function	Hàm đã được định nghĩa rồi. Lỗi xảy ra khi ta viết phần thân của một hàm nào đó ≥ 2 lần
16	Call of nonfunction	Câu lệnh gọi hàm của ta là sai. Tên hàm mà ta gọi có thể là một tên kiểu/hằng/biến,...
17	Cannot call 'main' from within the program	Không thể gọi thực hiện hàm main() trong chương trình, vì đây là một hàm đặc biệt, tự động thực hiện 1 lần trong mỗi lần chạy CT
18	Cannot cast from 'type1' to 'type2'	Không thể ép kiểu dữ liệu từ kiểu 1 sang kiểu
19	Cannot convert 'type1' to 'type2'	Không thể chuyển đổi kiểu dữ liệu từ kiểu 1 sang kiểu 2
20	Cannot initialize 'type1' with 'type2'	Không thể khởi gán dữ liệu thuộc kiểu 2 cho biến thuộc kiểu 1

21	Cannot modify a const object	Không thể thay đổi giá trị của một hằng số. Xảy ra khi ta thực hiện phép gán giá trị mới cho 1 hằng
22	Case outside of switch	Lệnh CASE nằm bên ngoài SWITCH
23	Case statement missing :	Lệnh CASE thiếu dấu 2 chấm (:))
24	Character constant must be one or two characters long	Kích thước của hằng kí tự không đúng. Xảy ra khi ta ghi một chuỗi dài các kí tự vào giữa cặp dấu nháy đơn ‘
25	Compound statement missing }	Thiếu dấu } kết thúc khối lệnh
26	Constant expression required	Vị trí này lẽ ra phải là một biểu thức hằng, có giá trị không đổi
27	Could not find a match for argument(s)	Không tìm thấy đối số thích hợp
28	Could not find file 'filename'	Không tìm thấy tập tin
29	Declaration is not allowed here	Vị trí khai báo sai. Không được khai báo tại
30	Declaration missing ;	Khai báo thiếu dấu chấm phẩy (;)
31	Declaration syntax error	Khai báo không đúng cú pháp
32	Declaration terminated incorrectly	Khai báo sai (gần giống lỗi trên)
33	Declaration was expected	Thiếu khai báo
34	Default outside of switch	Lệnh mặc định DEFAULT nằm bên ngoài khối lệnh SWITCH
35	Default value missing	Thiếu giá trị mặc định
36	Division by zero	Chia cho 0, lỗi này xảy ra khi mẫu số của một phân số có giá trị bằng 0
37	do statement must have while	Lệnh do phải đi với while . Xảy ra khi thiếu while trong câu lệnh do...
38	do-while statement missing (Thiếu ... trong câu lệnh do...while
39	do-while statement missing)	
40	do-while statement missing ;	
41	Duplicate case	Lệnh CASE bị trùng, xảy ra khi ta viết 2 dòng case khác nhau nhưng cùng một giá trị như nhau
42	Expression expected	Vị trí này phải là một biểu thức
43	Expression syntax	Sai cú pháp khi xây dựng biểu thức
44	Extra parameter in call to function	Gọi thực hiện hàm nhưng lại truyền dư tham
45	File name too long	Tên tập tin quá dài
46	For statement missing (Thiếu ... trong câu lệnh for
47	For statement missing)	
48	For statement missing ;	

49	'function' cannot return a value	Hàm có tên 'function' không thể trả về một giá trị, thông thường vì ta khai báo nó là hàm kiểu void
50	'function' must be declared with no parameters	Hàm có tên 'function' phải được khai báo không có tham số, xảy ra khi phần khai báo (prototype) và phần thân hàm không giống nhau về số tham số
51	'function' must be declared with one parameter	Tương tự như lỗi trên
52	'function' must be declared with two parameters	
53	Function 'function' should have a prototype	Hàm có tên 'function' cần phải được khai báo. Lỗi thường gặp khi trình biên dịch C không hiểu 1 tên hàm nào đó mà ta sử dụng, có thể do thiếu #include tập tin tiêu đề tương ứng, hoặc gõ sai tên.
54	Function call missing)	Gọi thực hiện hàm thiếu)
55	Function calls not supported	Không thể gọi hàm dạng này/kiểu này
56	Function should return a value	Hàm cần phải trả về 1 giá trị, xảy ra khi ta khai báo hàm có kiểu trả về nhưng lại thiếu câu lệnh return...
57	Goto statement missing label	Dùng lệnh goto mà không có nhãn
58	'identifier' is not a member of struct	Tên ... không phải là thành phần của cấu trúc, xảy ra khi ta viết tên thành phần sai
59	'identifier' is not a parameter	Tên ... không phải là một tham số
60	Identifier expected	Thiếu tên biến
61	If statement missing (Câu lệnh if thiếu mở hay đóng ngoặc
62	If statement missing)	
63	Illegal character 'character' (0x'value')	Kí tự không hợp lệ, thường xảy ra khi ta biểu diễn các hằng số hệ hexa, nhưng lại sử dụng các chữ cái khác A..F hay a..f
64	Illegal octal digit	Không phải là một số hệ 8 hợp lệ
65	Illegal pointer subtraction	Thực hiện phép trừ không hợp lệ trên con trỏ
66	Illegal use of floating point	Dùng dấu chấm động không đúng, ví dụ sử dụng phép toán % trên số thực chẳng hạn
67	Illegal use of pointer	Dùng con trỏ không hợp lệ
68	Implicit conversion of 'type1' to 'type2' not allowed	Không cho phép ngầm chuyển từ kiểu 1 sang kiểu 2
69	Improper use of typedef 'identifier'	Kiểu dữ liệu ... được sử dụng không đúng

70	Incompatible type conversion	Không thể chuyển đổi kiểu dữ liệu
71	Incorrect number format	Không phải là dữ liệu dạng số, thường xảy ra khi ta gõ các kí tự khác 0..9 trong một dữ liệu kiểu số
72	Incorrect use of default	Dùng DEFAULT không đúng
73	Invalid use of dot	Dùng dấu chấm (.) không đúng vị trí
74	Lvalue required	Vế trái của phép gán phải là một tên biến, lỗi xảy ra khi ta gán giá trị cho một hằng
75	Main must have a return type of int	Hàm main phải trả về 1 giá trị kiểu int
76	Misplaced break	Dùng break ngoài vòng lặp hoặc ngoài SWITCH
77	Misplaced continue	Dùng continue ngoài vòng lặp
78	Misplaced decimal point	Dấu chấm thập phân sai vị trí
79	Misplaced else	Dùng else sai vị trí (thiếu if, ...)
80	'new' and 'delete' not supported	Không được phép dùng new và delete trong cấp phát vùng nhớ động
81	No : following the ?	Toán tử điều kiện thiếu dấu 2 chấm (:)
82	No file name ending	Không có phần kết thúc tên tập tin
83	No file names given	Không có tên tập tin
84	No type information	Không tìm thấy thông tin gì về kiểu dữ liệu
85	Not an allowed type	Kiểu dữ liệu này không cho phép dùng ở đây
86	Numeric constant too large	Hằng số có giá trị quá lớn
87	Pointer to structure required on left side of -> or ->*	Xảy ra khi dùng con trỏ cấu trúc không đúng cách để truy xuất các thành phần của cấu trúc
88	sizeof may not be applied to a bit field	Toán tử sizeof() không dùng cho kiểu bit field
89	sizeof may not be applied to a function	Toán tử sizeof() không dùng cho hàm
90	Size of 'identifier' is unknown or zero	Kích thước của ... bằng 0 hoặc không xác định
91	Size of the type is unknown or zero	Kích thước của kiểu dữ liệu bằng 0 hoặc không xác định
92	Statement missing ;	Thiếu dấu chấm phẩy (;), thông thường do thiếu dấu ; tại dòng trên của dòng báo lỗi
93	Structure required on left side of . or .*	Xảy ra khi truy xuất các thành phần của cấu trúc không đúng cách.

94	Structure size too large	Kích thước của cấu trúc quá lớn, vượt quá giới hạn 64K chẳng hạn.
95	Switch statement missing (Câu lệnh switch thiếu ngoặc. Phần giá trị của lệnh switch phải được đặt trong cặp dấu ngoặc
96	Switch statement missing)	
97	The value for 'identifier' is not within the range of an int	Giá trị của biến ... không nằm trong phạm vi của một biến kiểu nguyên (int)
98	Too few parameters in call to function	Gọi thực hiện hàm nhưng lại truyền không đủ số
99	Too many decimal points	Biểu diễn số thực nhưng dùng nhiều hơn 1 dấu chấm thập phân
100	Too many default cases	Trong câu lệnh switch có nhiều hơn 1 lệnh default
101	Too many errors or warning messages	Có quá nhiều lỗi hoặc cảnh báo trong chương trình. Xảy ra khi chương trình có nhiều hơn 25 lỗi
102	Too many types in declaration	Khai báo quá nhiều kiểu dữ liệu mới (ít gặp)
103	Too much global data defined in file	Có quá nhiều biến toàn cục trong chương trình, gây tràn vùng nhớ dành riêng cho các biến này
104	Type mismatch in default argument value	Giá trị mặc định của tham số truyền cho CT con bị sai kiểu
105	Type mismatch in default value for parameter 'parameter'	Giá trị mặc định của tham số ... bị sai kiểu.
106	Type mismatch in parameter 'number' in call to 'function'	Truyền tham số cho hàm 'function' bị sai kiểu ở tham số 'number'
107	Type mismatch in parameter 'parameter'	Tham số ... bị sai kiểu
108	Type mismatch in parameter 'parameter' in call to 'function'	Gần giống lỗi 106
109	Type name expected	Thiếu tên kiểu tại vị trí báo lỗi
110	Type 'typename' may not be defined here	Kiểu dữ liệu ... không thể định nghĩa ở vị trí này
111	Unable to create turboc.\$ln	Không thể tạo được tập tin turboc.1\$ Thường xảy ra khi ta chạy TurboC trên đĩa mềm hay đĩa CD.
112	Unable to execute command 'command'	Không thể thực hiện lệnh ...

113	Unable to open include file 'filename'	Không thể mở được tập tin tiêu đề ... thường xảy ra do ta viết tên tập tin tiêu đề sai, hoặc tập tin này không tồn tại trên đĩa
114	Undefined label 'identifier'	Nhãn ... chưa được khai báo
115	Undefined structure 'structure'	Cấu trúc ... chưa được khai báo
116	Undefined symbol 'identifier'	Ký hiệu ... chưa được khai báo, thường xảy ra trong trường hợp ta sử dụng biến mà chưa khai báo
117	Unexpected }	Dư dấu đóng ngoặc }
118	Unexpected end of file in comment started on 'line number'	Thường xảy ra trong trường hợp thiếu dấu đóng ngoặc } của hàm main()
119	Unexpected end of file in conditional started on 'line number'	
120	Unknown language, must be C or C++	Một cú pháp lạ, không phải là cú pháp của C hay C++
121	User break	Chương trình bị ngắt do người sử dụng
122	Value of type void is not allowed	Không được phép gán dữ liệu cho biến kiểu void
123	Variable 'identifier' is initialized more than once	Biến ... được khởi tạo nhiều lần.
124	void & is not a valid type	Không chấp nhận tham chiếu đến biến kiểu void
125	While statement missing (Câu lệnh while thiếu ngoặc. Phần điều kiện của lệnh while phải được đặt trong dấu ngoặc
126	While statement missing)	

TÀI LIỆU THAM KHẢO



- [1] Nguyễn Văn Hiếu, ***Giáo trình Lập trình căn bản***, Khoa CNTT – Trường ĐHSPKT Vĩnh Long, 2015.
- [2] Phạm Văn Ất, Nguyễn Hiếu Cường, Đỗ Văn Tuấn, Lê Trường Thông, ***Giáo trình Kỹ thuật lập trình C***, NXB Hồng Đức, 2009.
- [3] Trần Đình Quế, Nguyễn Mạnh Hùng, ***Ngôn ngữ lập trình C++***, Học viện công nghệ bưu chính viễn thông, 2006.
- [4] James P. Cohoon and Jack W. Davidson, ***C++ Program Design – An Introduction to Programming and Object-Oriented Design***, 2nd edition, WCB McGraw-Hill, 1999.
- [5] Robert Sedgewick, ***Cẩm nang Thuật Toán Vol.1***, Nhà xuất bản Khoa học Kỹ thuật.
- [6] Robert Sedgewick, ***Cẩm nang Thuật Toán Vol.2***, Nhà xuất bản Khoa học Kỹ thuật.

MỤC LỤC



CHƯƠNG 1: TỔNG QUAN VỀ NGÔN NGỮ C++	Trang 1
CHƯƠNG 2: CÁC CẤU TRÚC ĐIỀU KHIỂN	Trang 5
CHƯƠNG 3: DỮ LIỆU KIỂU MẢNG.....	Trang 13
CHƯƠNG 4: DỮ LIỆU KIỂU CHUỖI	Trang 23
CHƯƠNG 5: CON TRỎ VÀ HÀM	Trang 29
CHƯƠNG 6: DỮ LIỆU KIỂU CẤU TRÚC	Trang 42
CHƯƠNG 7: DỮ LIỆU KIỂU TẬP TIN	Trang 48
PHỤ LỤC 1: NGUYÊN TẮC CHUNG KHI SỬA LỖI.....	Trang 52
PHỤ LỤC 2: CÁC THÔNG BÁO LỖI THƯỜNG GẶP	Trang 53
TÀI LIỆU THAM KHẢO	Trang 59