HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BỘ MÔN TOÁN RỜI RẠC 1

BÀI LẬP TRÌNH SỐ 1

Nhóm : 14 – Tổ 3
Giảng viên : Nguyễn Thị Mai Trang
Ngày: 27/10/2023
Điểm:

Thành viên nhóm :

1. Lê Ngọc Đức B22DCAT092
2. Phạm Hồng Dương B22DCAT068
3. Đỗ Quốc Trung B22DCAT306
4. Nguyễn Thị Thùy Linh B22DCAT176
5. Nguyễn Minh Lương B22DCPT155

BÀI TOÁN MA TRẬN SỐ NGUYÊN TỐ

ĐỀ BÀI :

Cho hình vuông gồm 25 hình vuông đơn vị. Hãy điền các số từ 0 đến 9 vào các hình vuông đơn vị sao cho những điều kiện sau được thỏa mãn:

a, Đọc từ trái sang phải theo hang ta nhận được 5 số nguyên tố có 5 chữ số;

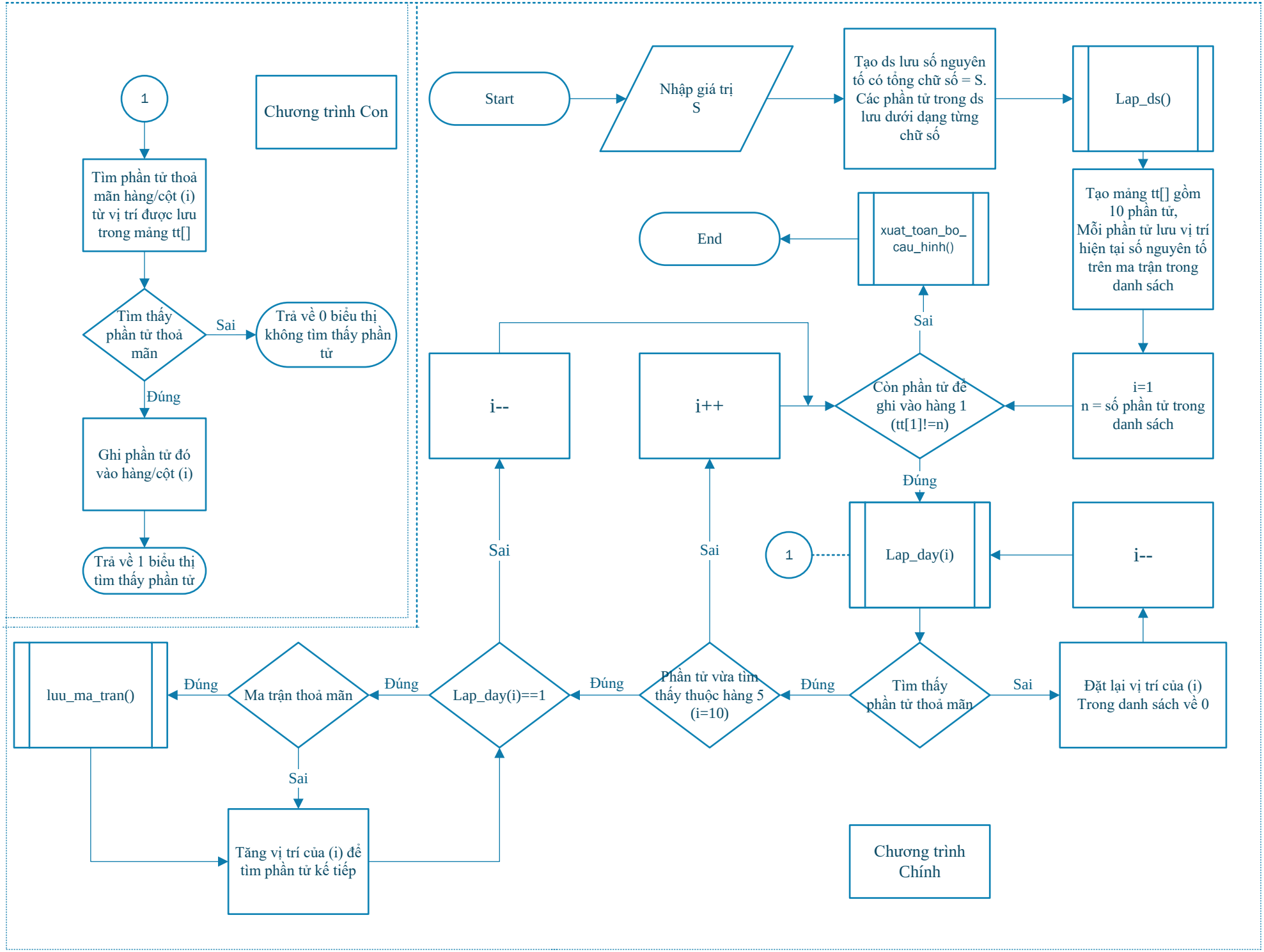
b, Đọc từ trên xuống dưới theo cột ta nhận được 5 số nguyên tố có 5 chữ số;

c, Đọc theo 2 đường chéo chính ta nhận được 2 số nguyên tố có 5 chữ số;

d, Tổng các chữ số trong mỗi số nguyên tố đều là S cho trước.

Ví dụ hình vuông dưới đây có S = 11.

3	5	1	1	1
5	0	0	3	3
1	0	3	4	3
1	3	4	2	1
1	3	3	1	3


Chương trình Con
1
Tìm phần tử thoả mãn hàng/cột (i) từ vị trí được lưu trong mảng tt[]
Tìm thấy phần tử thoả mãn
Sai
Trả về 0 biểu thị không tìm thấy phần tử
Đúng
Ghi phần tử đó vào hàng/cột (i)
Trả về 1 biểu thị tìm thấy phần tử
Start
Nhập giá trị S
Tạo ds lưu số nguyên tố có tổng chữ số = S. Các phần tử trong ds lưu dưới dạng từng chữ số
Lap_ds()
Tạo mảng tt[] gồm 10 phần tử, Mỗi phần tử lưu vị trí hiện tại số nguyên tố trên ma trận trong danh sách
i=1
n = số phần tử trong danh sách
Còn phần tử để ghi vào hàng 1 (tt[1]!=n)
Sai
xuat_toan_bo_cau_hinh()
End
Đúng
Lap_day(i)
1
i--
Tìm thấy phần tử thoả mãn
Sai
Đặt lại vị trí của (i) Trong danh sách về 0
Đúng
Phần tử vừa tìm thấy thuộc hàng 5 (i=10)
Đúng
i++
Sai
Lap_day(i)==1
Đúng
i--
Sai
Ma trận thoả mãn
Đúng
luu_ma_tran()
Sai
Tăng vị trí của (i) để tìm phần tử kế tiếp
Chương trình Chính

CHƯƠNG TRÌNH TRÊN C++

```
#include "iostream"
#include "vector"
#include "cmath"
#include "fstream"
using namespace std;
fstream fio ("SAVE.txt",fstream::out);  //Tạo file "save.txt" để lưu cấu hình
struct cau_truc_so {    //lưu số thoả mãn thành các phần tử riêng lẻ để dễ tham chiếu điều kiện
   int a[5];
};
vector<cau_truc_so> ds (0); //Tạo danh sách lưu các số dưới dạng cấu trúc như trên
int MT[5][5],dem=0,tt[11];
//Chương trình con kiểm số nguyên tố kiểu int
bool ktra_nto(int x) {
   if (x%2==0) return 0;   //Nếu chia hết cho 2 thì x không phải số nguyên tố
   //Nếu không thì kiểm tra tính chia hết cho các số lẻ từ 3 đến căn 2 của số đó
   for (int i=3;i<=sqrt(x);i+=2)
      if (x%i==0) return 0;
   return 1;
}
//Chương trình lập danh sách các số nguyên tố có 5 chữ số, tổng các chữ số bằng S
//Các phần tử được lưu dưới dạng cau_truc_so
```

```
void lap_ds(int S) {
   for (int i=10000;i<=100000;i++)
      if (ktra_nto(i)==1) {
         int t=i;
         cau_truc_so X;
         for (int i=4;i>=0;i--) {    //chuyển số nguyên tố i kiểu int thành
kiểu cau_truc_so
            X.a[i]=t%10;
            t/=10;
         }
         if (X.a[0]+X.a[1]+X.a[2]+X.a[3]+X.a[4]==S)  //Kiểm tra tổng
các chữ số có bằng S không
            ds.push_back(X);    //Đúng thì cho vào cuối danh sách
      }
}
//Chương trình con kiểm tra ma trận có thoả mãn hay không
bool ktra() {
   int test=0;
   for (int i=0;i<ds.size();i++) {
      //Kiểm tra đường chéo có nằm trong danh sách không
      if (ds[i].a[0]==MT[0][0] && ds[i].a[1]==MT[1][1] &&
ds[i].a[2]==MT[2][2] && ds[i].a[3]==MT[3][3] && ds[i].a[4]==MT[4][4])
test++;
      //Kiểm tra cột 5 có nằm trong danh sách không
      if (ds[i].a[0]==MT[0][4] && ds[i].a[1]==MT[1][4] &&
ds[i].a[2]==MT[2][4] && ds[i].a[3]==MT[3][4] && ds[i].a[4]==MT[4][4])
test++;
   }
   if (test==2)    //test=2 <=> Cột ả 2 điều kiện trên đúng, trả về 1
      return 1;
```

```
        return 0;   //Nếu không thì ma trận không thoả mãn trả về 0
    }

    //Chương trình con tìm phần tử ở cột/hàng/đường thứ x
    //các cột/hàng/đường được đánh số theo thứ tự ưu tiên tìm phần tử thoả mãn
    //Nếu tìm thấy phần tử thoả mãn thì trả về 1, nếu không thì trả về 0
    int lap_day (int x) {
        if (x==1) { //x=1 tương đương hàng 1
            for (tt[x];tt[x]<ds.size();tt[x]++)
                if (ds[tt[x]].a[1]!=0 && ds[tt[x]].a[2]!=0 && ds[tt[x]].a[3]!=0) {
                    MT[0][0]=ds[tt[x]].a[0];
                    MT[0][1]=ds[tt[x]].a[1];
                    MT[0][2]=ds[tt[x]].a[2];
                    MT[0][3]=ds[tt[x]].a[3];
                    MT[0][4]=ds[tt[x]].a[4];
                    return 1;
                }
            return 0;
        }
        if (x==2) { //x=2 tương đương cột 1
            for (tt[x];tt[x]<ds.size();tt[x]++)
                if (ds[tt[x]].a[0]==MT[0][0] && ds[tt[x]].a[1]!=0 && ds[tt[x]].a[2]!=0 && ds[tt[x]].a[3]!=0) {
                    MT[1][0]=ds[tt[x]].a[1];
                    MT[2][0]=ds[tt[x]].a[2];
                    MT[3][0]=ds[tt[x]].a[3];
                    MT[4][0]=ds[tt[x]].a[4];
```

```
                return 1;
            }
        return 0;
    }
    if (x==3) { //x=3 tương đương đường chéo chính 2
        for (tt[x];tt[x]<ds.size();tt[x]++)
            if (ds[tt[x]].a[0]==MT[0][4] && ds[tt[x]].a[4]==MT[4][0]) {
                MT[1][3]=ds[tt[x]].a[1];
                MT[2][2]=ds[tt[x]].a[2];
                MT[3][1]=ds[tt[x]].a[3];
                return 1;
            }
        return 0;
    }
    if (x==4) { //x=4 tương đương hàng 2
        for (tt[x];tt[x]<ds.size();tt[x]++)
            if (ds[tt[x]].a[0]==MT[1][0] && ds[tt[x]].a[3]==MT[1][3]) {
                MT[1][1]=ds[tt[x]].a[1];
                MT[1][2]=ds[tt[x]].a[2];
                MT[1][4]=ds[tt[x]].a[4];
                return 1;
            }
        return 0;
    }
    if (x==5) { //x=5 tương đương cột 2
        for (tt[x];tt[x]<ds.size();tt[x]++)
            if (ds[tt[x]].a[0]==MT[0][1] && ds[tt[x]].a[1]==MT[1][1] &&
ds[tt[x]].a[3]==MT[3][1]) {
                MT[2][1]=ds[tt[x]].a[2];
                MT[4][1]=ds[tt[x]].a[4];
```

```
                return 1;
            }
        return 0;
    }
    if (x==6) { //x=6 tương đương hàng 3
        for (tt[x];tt[x]<ds.size();tt[x]++)
            if (ds[tt[x]].a[0]==MT[2][0] && ds[tt[x]].a[1]==MT[2][1] &&
ds[tt[x]].a[2]==MT[2][2]) {
                MT[2][3]=ds[tt[x]].a[3];
                MT[2][4]=ds[tt[x]].a[4];
                return 1;
            }
        return 0;
    }
    if (x==7) { //x=7 tương đương cột 3
        for (tt[x];tt[x]<ds.size();tt[x]++)
            if (ds[tt[x]].a[0]==MT[0][2] && ds[tt[x]].a[1]==MT[1][2] &&
ds[tt[x]].a[2]==MT[2][2]) {
                MT[3][2]=ds[tt[x]].a[3];
                MT[4][2]=ds[tt[x]].a[4];
                return 1;
            }
        return 0;
    }
    if (x==8) { //x=8 tương đương hàng 4
        for (tt[x];tt[x]<ds.size();tt[x]++)
            if (ds[tt[x]].a[0]==MT[3][0] && ds[tt[x]].a[1]==MT[3][1] &&
ds[tt[x]].a[2]==MT[3][2]) {
                MT[3][3]=ds[tt[x]].a[3];
                MT[3][4]=ds[tt[x]].a[4];
```

```
                return 1;
            }
        return 0;
    }
    if (x==9) { //x=9 tương đương cột 4
        for (tt[x];tt[x]<ds.size();tt[x]++)
            if (ds[tt[x]].a[0]==MT[0][3] && ds[tt[x]].a[1]==MT[1][3] &&
ds[tt[x]].a[2]==MT[2][3] && ds[tt[x]].a[3]==MT[3][3]) {
                MT[4][3]=ds[tt[x]].a[4];
                return 1;
            }
        return 0;
    }
    if (x==10) { //x=10 tương đương hàng 5
        for (tt[x];tt[x]<ds.size();tt[x]++)
            if (ds[tt[x]].a[0]==MT[4][0] && ds[tt[x]].a[1]==MT[4][1] &&
ds[tt[x]].a[2]==MT[4][2] && ds[tt[x]].a[3]==MT[4][3]) {
                MT[4][4]=ds[tt[x]].a[4];
                return 1;
            }
        return 0;
    }
    return 0;
}
//Chương trình con in cấu hình được lưu trong file
void xuat_toan_bo_cau_hinh() {
    std::cout << "Tong cau hinh: " << dem << '\n';
    for (int t=0;t<dem;t++) {
        cout << "Cau hinh: " << t+1 << '\n';
        for (int i=0,x;i<5;i++) {
```

```
            for (int j=0;j<5;j++) {
               fio >> x;
               cout << x << ' ';
            }
            cout << '\n';
         }
      }
      fio.close();
   }
   //Chương trình con lưu cấu hình hiện tại
   void luu_ma_tran() {
      for (int i=0;i<5;i++) {
         for (int j=0;j<5;j++)
            fio << MT[i][j] << ' ';
         fio << '\n';
      }
      dem++;
   }
   int main() {
      cout << "nhap gia tri S: ";
      int S;
      cin >> S;
      lap_ds(S);

      //tt[] là mảng lưu vị trí hiện tại trong danh sách các số nguyên tố trên ma trận, ứng với 5 hàng, 4 cột, 1 đường chéo
      //tt[1] <=> Hàng 1 | tt[2] <=> Cột 1 | tt[3] <=> Chéo 2 | tt[4] <=> Hàng 2 | tt[5] <=> Cột 2
      //tt[6] <=> Hàng 3 | tt[7] <=> Cột 3 | tt[8] <=> Hàng 4 | tt[9] <=> Cột 4 | tt[10] <=> Hàng 5
```

```
        for (int i=0;i<11;i++)
           tt[i]=0;    //khởi tạo tất cả về 0

        int i=1,n=ds.size();
        //Lặp đến khi hết phần tử để điền vào hàng 1
        while (tt[0]==0) {
           int k=lap_day(i);   //lấp đầy cột/hàng i
           if (k==0) { //k=0 -> không tìm được số thoả mãn cột/hàng i
              tt[i]=0;    //vị trí cột/hàng i về 0, để tìm lại từ đầu trong danh
sách
              tt[--i]++;  //vị trí cột/hàng i-1 tăng thêm 1, để tìm lại từ phần
từ kế tiếp trong danh sách
           }
           else if (i==10) {    //i=10 tương đương hàng cuối cùng cần tìm
              while (lap_day(i)==1) { //nếu vẫn còn số nguyên tố trong
danh sách thoả mãn hàng cuối
                 if (ktra()==1)
                    luu_ma_tran(); //nếu ma trận thoả mãn thì lưu ma trận
vào file
                 tt[i]++; //tăng vị trí để tìm phần tử kế tiếp
              }
              i--;    //nếu hết phần tử thì quay lại bước tìm phần tử
hàng/cột thứ 9
           }
           else i++;   //Tìm phần tử kế tiếp
        }
        fio.close();
        fio.open("SAVE.txt",fstream::in);   //mở file vừa lưu ma trận và
đọc
        xuat_toan_bo_cau_hinh(); //xuất ma trận đã lưu ra màn hình}
```

CHẠY KẾT QUẢ

```
nhap gia tri S: 11
Tong cau hinh: 6
Cau hinh: 1
1 1 3 5 1
1 4 0 3 3
3 0 3 2 3
5 3 2 0 1
1 3 3 1 3
Cau hinh: 2
1 1 3 5 1
3 3 2 0 3
3 0 3 2 3
1 4 0 3 3
3 3 3 1 1
Cau hinh: 3
1 3 3 1 3
1 3 0 4 3
3 2 3 0 3
5 0 2 3 1
1 3 3 3 1
Cau hinh: 4
2 1 5 2 1
5 0 4 1 1
1 2 1 6 1
2 7 0 1 1
1 1 1 1 7
Cau hinh: 5
3 5 1 1 1
5 0 0 3 3
1 0 3 4 3
1 3 4 2 1
1 3 3 1 3
Cau hinh: 6
5 1 1 3 1
1 0 4 3 3
1 4 3 0 3
3 3 0 2 3
1 3 3 3 1
```

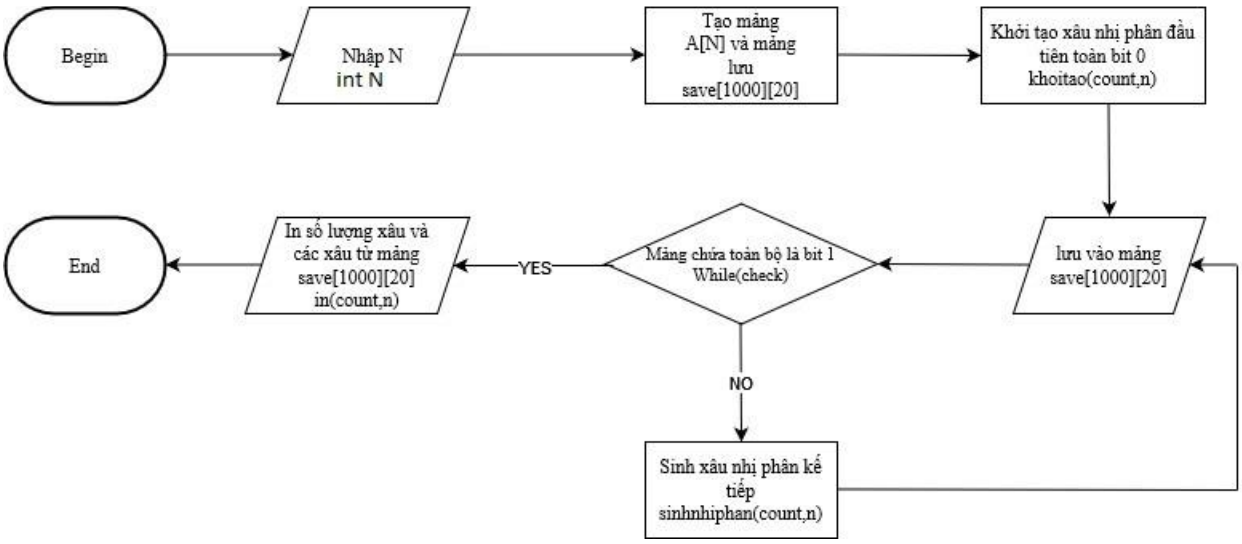
BÀI TOÁN SINH XÂU NHỊ PHÂN

ĐỀ BÀI

Liệt kê (duyệt) các xâu có độ dài n . Xâu X = (x1x2…xn), xi = 0,1 ; i= 1,2,…n được gọi là xâu nhị phân có độ dài n . Ví dụ, n=4 ta có 16 xâu nhị phân dưới đây:

STT	X = (x1x2…xn)	F(x)	STT	X =(x1x2…xn)	F(x)
1	0000	0	9	1000	8
2	0001	1	10	1001	9
3	0010	2	11	1010	10
4	0011	3	12	1011	11
5	0100	4	13	1100	12
6	0101	5	14	1101	13
7	0110	6	15	1110	14
8	0111	7	16	1111	15

LƯU ĐỒ :



CHƯƠNG TRÌNH C++:

```
#include<bits/stdc++.h>
```

```
using namespace std;
int a[23]={0};
int n;  //n = 4
int save[1000][20];
bool check;
void khoitao(int &count, int &n)
{
	for(int i=0;i<n;i++)
	{
		a[i] = 0;
		save[count][i] = a[i];
	}
}
void sinhnhiphan(int &count, int &n)
{
		int i = n-1;
		while(i>=0 && a[i])
		{
			a[i] = 0;
			i--;
		}
		if(i>=0)
```

```
            {
                a[i] = 1;
                for(int index = 0; index < n; index++)
                {
                    save[count][index] = a[index];
                }
                count++;
            }
            else check = false;
}
void in(int &count, int &n)
{
    int tt=1;
    for(int i=0;i<count;i++)
    {
        cout<<"xau thu "<<tt<<": ";
        for(int j=0;j<n;j++)
        {
            cout<<save[i][j];
        }
        cout<<endl;
        tt++;
```

```
        }
}
int main()
{
        ios_base::sync_with_stdio(0);
        cin.tie(0); cout.tie(0);
        cout<<"Nhap so n la:"<<endl;
        cin>>n;
        int count = 0;
        khoitao(count, n);
        count++;
        check = true;
        while(check){
                sinhnhiphan(count,n);
        }
        cout<<"Tong so xau n la: "<<count<<endl;
        in(count,n);
}
```

CHẠY CHƯƠNG TRÌNH C++

```
Nhap so n la:
4
Tong so xau n la: 16
xau thu 1: 0000
xau thu 2: 0001
xau thu 3: 0010
xau thu 4: 0011
xau thu 5: 0100
xau thu 6: 0101
xau thu 7: 0110
xau thu 8: 0111
xau thu 9: 1000
xau thu 10: 1001
xau thu 11: 1010
xau thu 12: 1011
xau thu 13: 1100
xau thu 14: 1101
xau thu 15: 1110
xau thu 16: 1111

--------------------------------
Process exited after 9.001 seconds with return value 0
Press any key to continue . . .
```