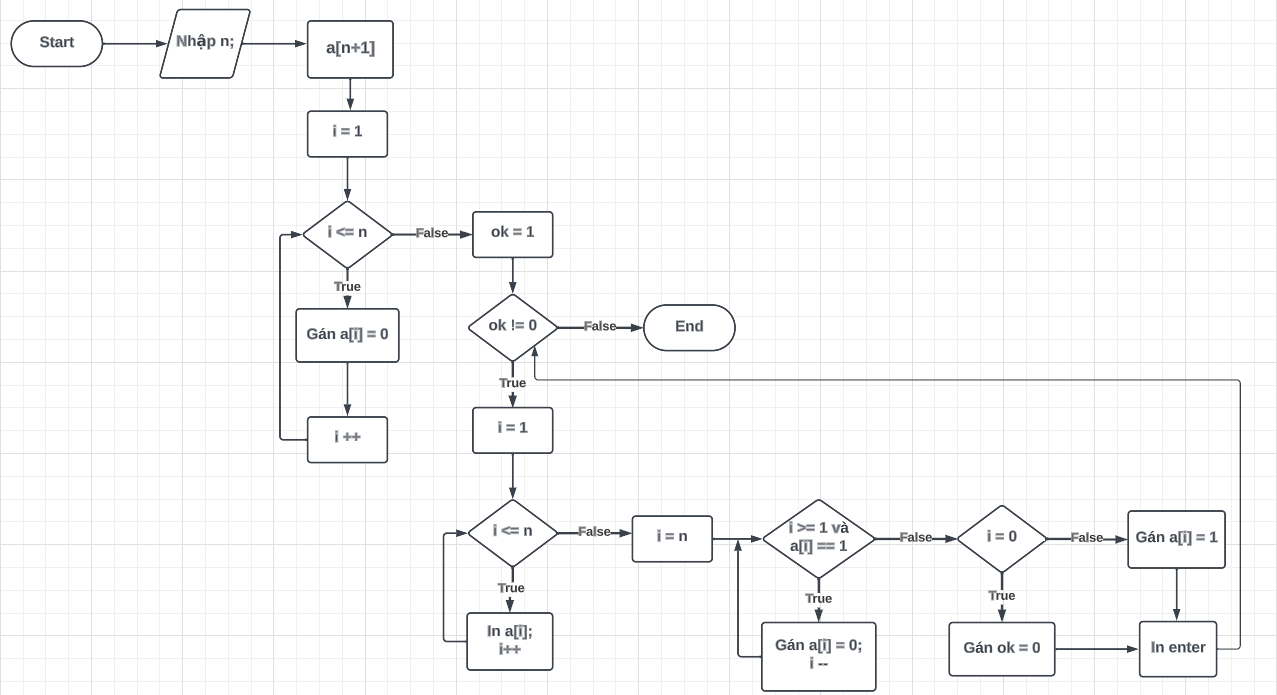
|  |  |
| --- | --- |
| **Họ và tên** | 1. Nguyễn Hồng Giáp 2. Đặng Thị Hồng Vân 3. Lê Thị Ngọc Hương |
| **Nhóm học phần** | Nhóm 05 |
| **Bài lập trình** | Số 01 |
| **Môn** | Toán rời rạc |
| **Giảng viên** | Nguyễn Thị Mai Trang |
| **Ngày** | 9/10/2023 |
| **Điểm** |  |

**Bài 1**: Liệt kê ( duyệt ) các xâu nhị phân có độ dài n

Xâu X = () được gọi là xâu nhị phân có độ dài n.

* 1. **Sơ đồ khối**
  2. **. Chương trình trên C++**

#include <bits/stdc++.h>

using namespace std;

int main(){

int n; cin >> n;

int a[n+1]; // x1x2...xn

for(int i=1; i<=n; i++){

a[i] =0;

}

int ok=1;

while(ok){

for(int i=1; i<=n; i++){

cout << a[i] ;

}

int i=n;

while(i >=1 && a[i]==1){

a[i] = 0;

i--;

}

if(i==0){

ok=0;

}

else {

a[i] = 1;

}

cout << endl;

}

}

* 1. **Kết quả**

4

0000

0001

0010

0011

0100

0101

0110

0111

1000

1001

1010

1011

1100

1101

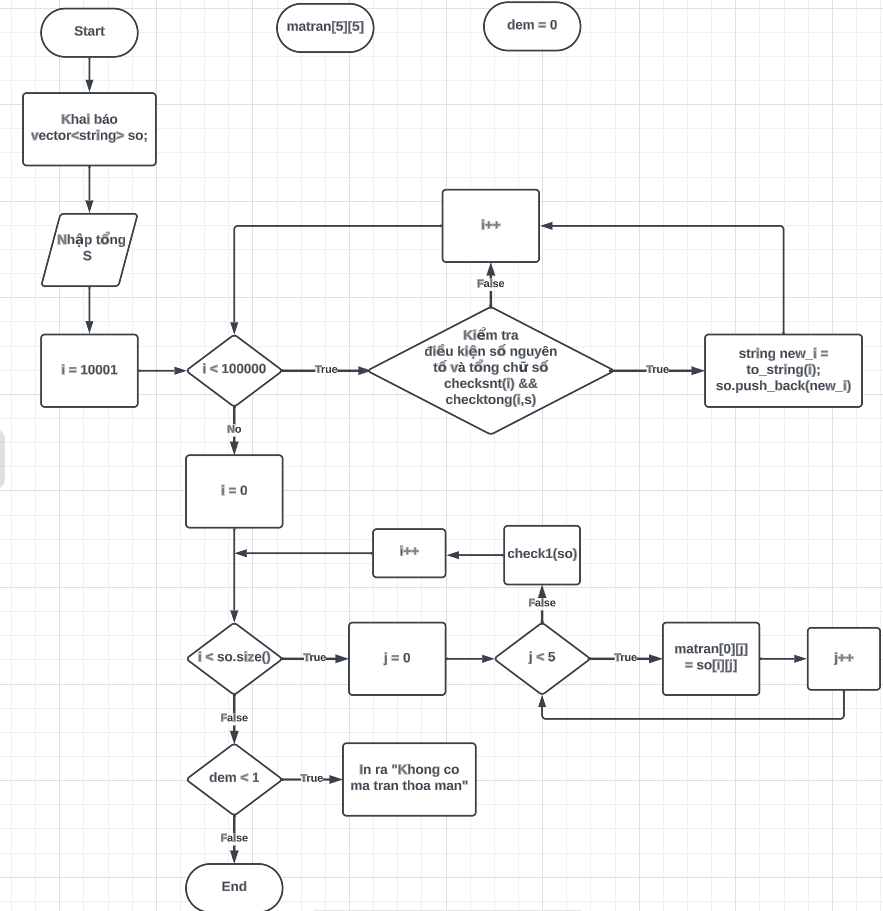
1110

1111

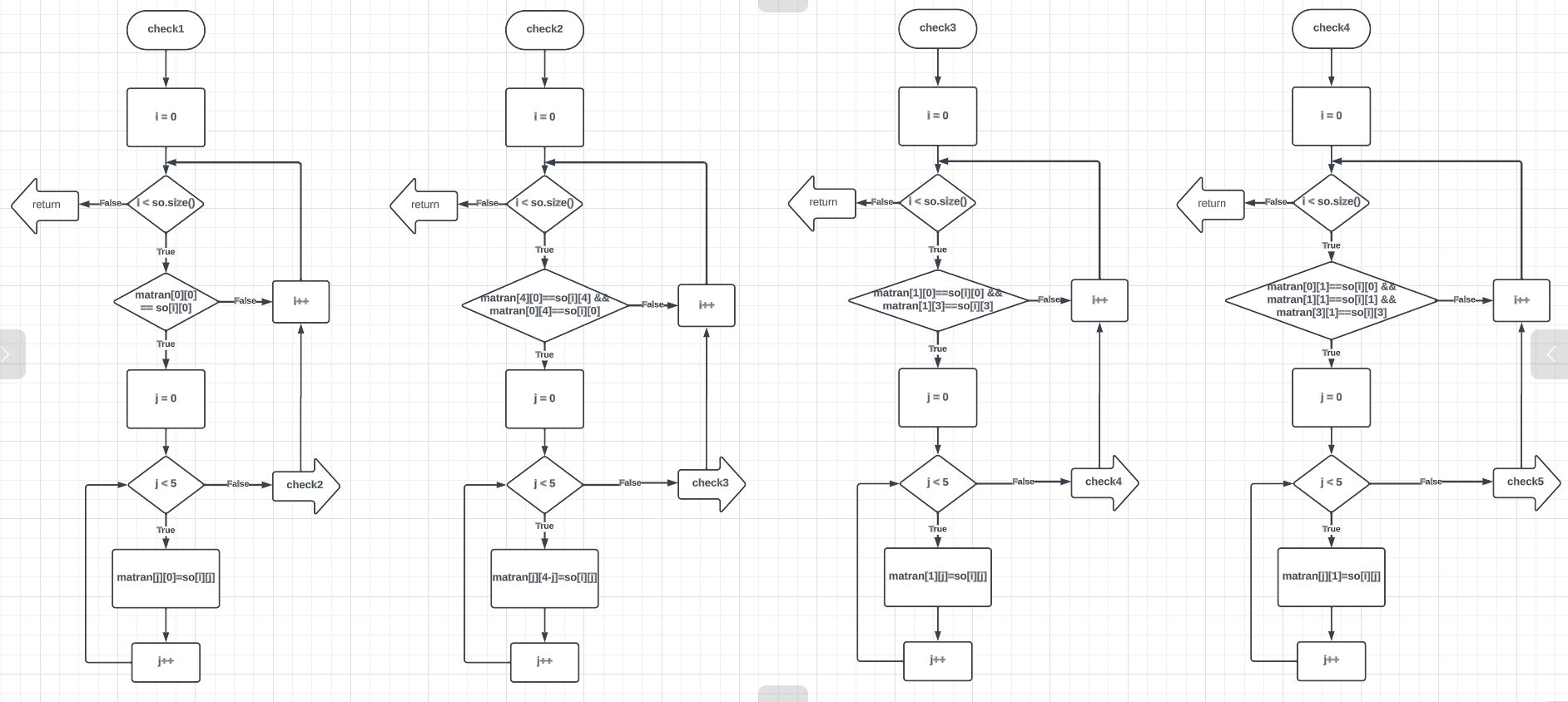
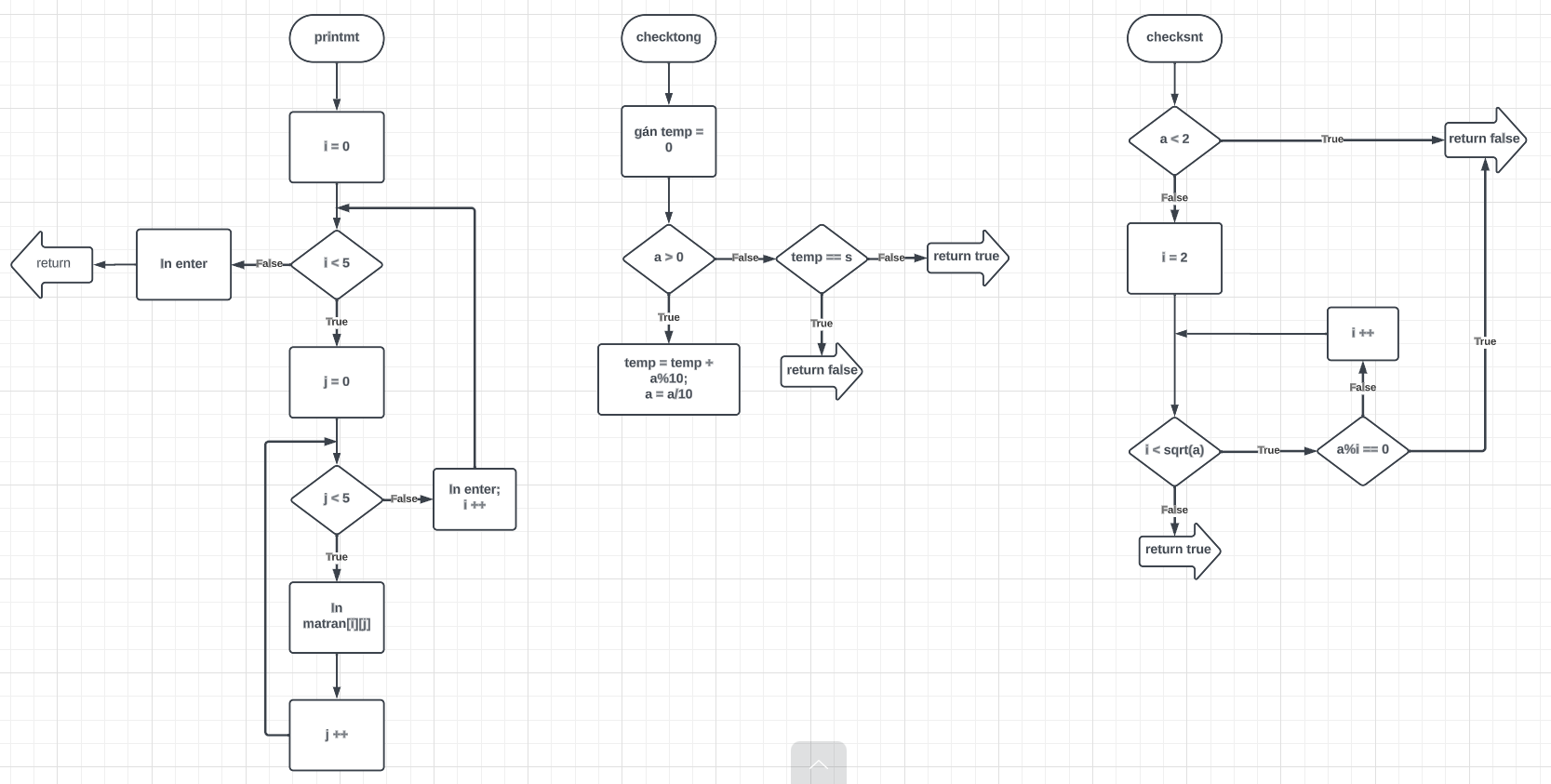
**Bài 2:**

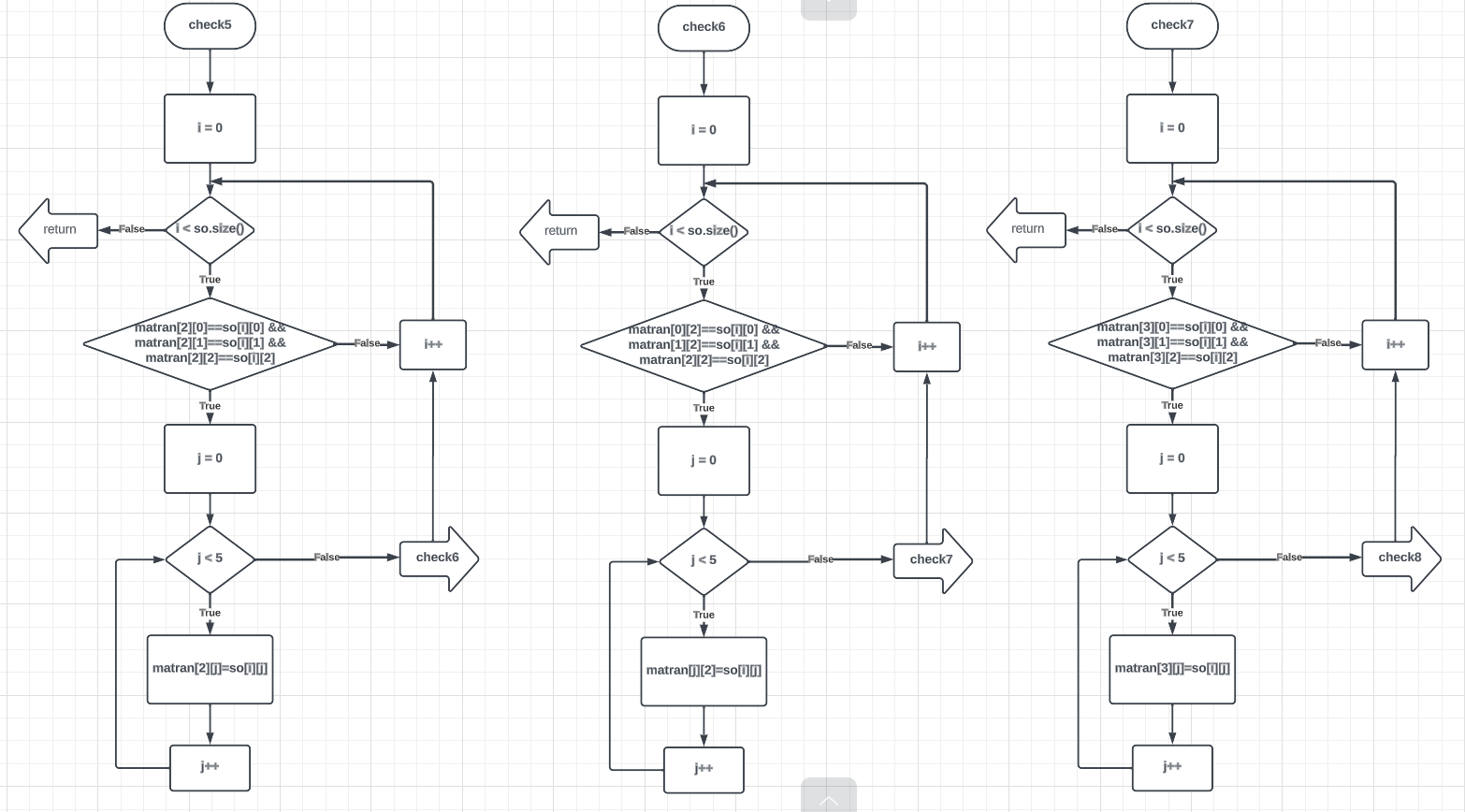
Hình vuông gồm 25 hình vuông đơn vị có thể được điền các số từ 0 đến 9 sao cho những điều kiện sau được thỏa mãn:

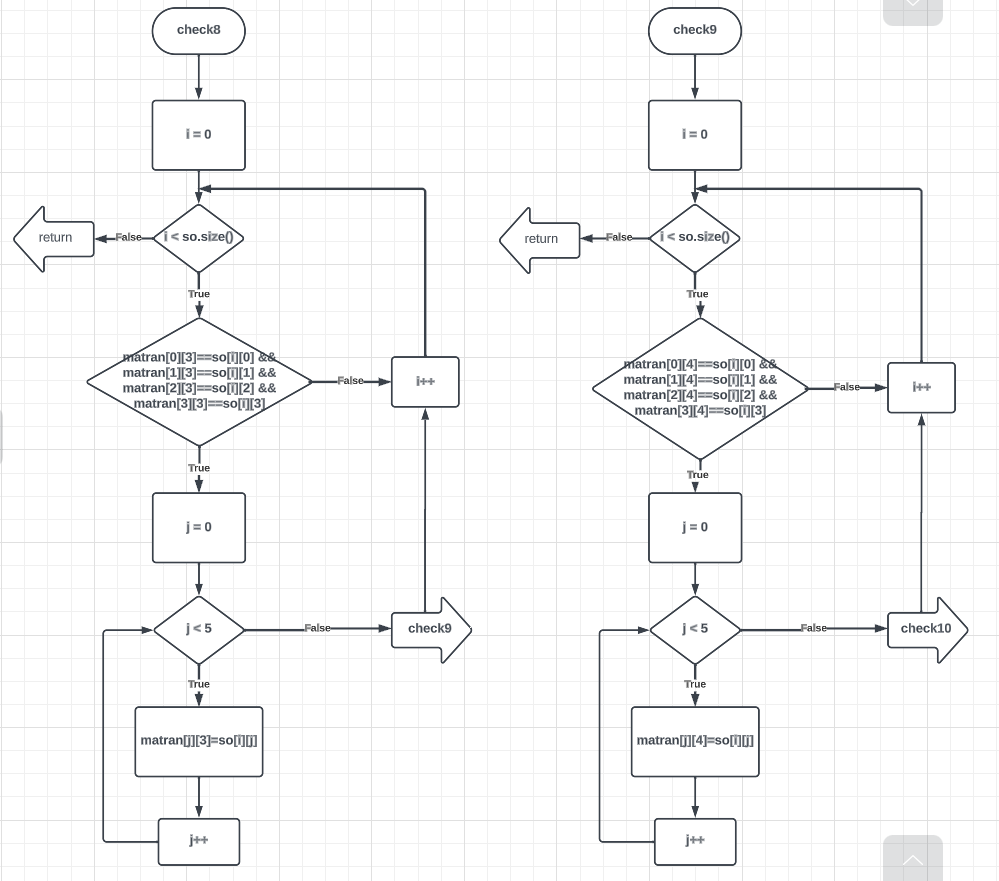
* Đọc từ trái sang phải theo hàng ta nhận được 5 số nguyên tố có 5 chữ số.
* Đọc từ trên xuống dưới theo cột ta nhận được 5 số nguyên tố có 5 chữ số.
* Đọc theo hai đường chéo chính ta nhận được 2 số nguyên tố có 5 chữ số.
* Tổng các chữ số trong mỗi số nguyên tố đều là S cho trước.

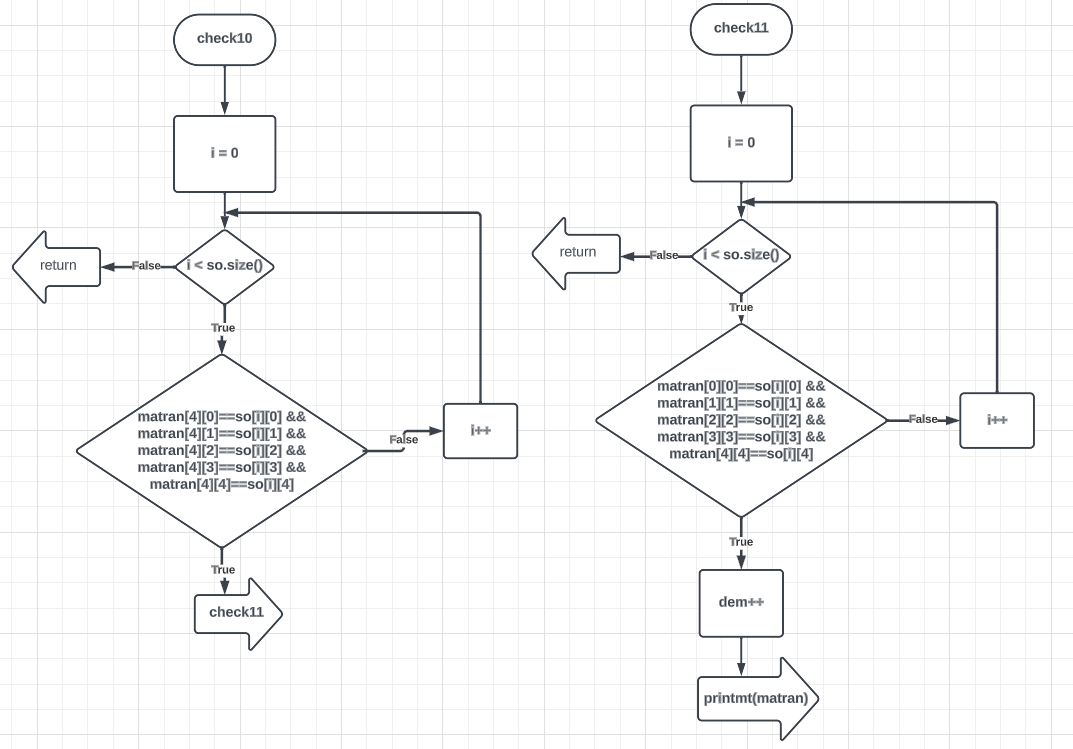
**2.1. Sơ đồ khối**

**Sơ đồ khối hàm chính (hàm main)**

****



****



**2.2. Chương trình C++**

#include<bits/stdc++.h>

using namespace std;

char matran[5][5];

int dem = 0;

bool checksnt(int a){

if (a<2) return false;

for (int i = 2; i <= sqrt(a); i++)

{

if(a%i==0) return false;

}

return true;

}

bool checktong(int a, int s){

int temp = 0;

while (a>0)

{

temp += a%10;

a=a/10;

}

if(temp==s) return true;

return false;

}

void printmt(char matran[5][5]){

cout << "Ma tran " << dem << ":" << endl;

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 5; j++)

{

cout << matran[i][j] << " ";

}

cout << endl;

}

cout << "\n";

}

void check11(const vector<string> so ){

for (int i = 0; i < so.size(); i++)

{

if(matran[0][0]==so[i][0] && matran[1][1]==so[i][1] && matran[2][2]==so[i][2] && matran[3][3]==so[i][3] && matran[4][4]==so[i][4]){

dem ++;

}

}

for (int i = 0; i < so.size(); i++) {

if(matran[0][0]==so[i][0] && matran[1][1]==so[i][1] && matran[2][2]==so[i][2] && matran[3][3]==so[i][3] && matran[4][4]==so[i][4]){

printmt(matran);

}

}

}

void check10(const vector<string> so ){

for (int i = 0; i < so.size(); i++)

{

if(matran[4][0]==so[i][0] && matran[4][1]==so[i][1] && matran[4][2]==so[i][2] && matran[4][3]==so[i][3] && matran[4][4]==so[i][4]){

check11(so);

}

}

}

void check9(const vector<string> so ){

for (int i = 0; i < so.size(); i++)

{

if(matran[0][4]==so[i][0] && matran[1][4]==so[i][1] && matran[2][4]==so[i][2] && matran[3][4]==so[i][3]){

for (int j = 0; j < 5; j++)

{

matran[j][4]=so[i][j];

}

check10(so);

}

}

}

void check8(const vector<string> so ){

for (int i = 0; i < so.size(); i++)

{

if(matran[0][3]==so[i][0] && matran[1][3]==so[i][1] && matran[2][3]==so[i][2] && matran[3][3]==so[i][3]){

for (int j = 0; j < 5; j++)

{

matran[j][3]=so[i][j];

}

check9(so);

}

}

}

void check7(const vector<string> so ){

for (int i = 0; i < so.size(); i++)

{

if(matran[3][0]==so[i][0] && matran[3][1]==so[i][1] && matran[3][2]==so[i][2]){

for (int j = 0; j < 5; j++)

{

matran[3][j]=so[i][j];

}

check8(so);

}

}

}

void check6(const vector<string> so ){

for (int i = 0; i < so.size(); i++)

{

if(matran[0][2]==so[i][0] && matran[1][2]==so[i][1] && matran[2][2]==so[i][2]){

for (int j = 0; j < 5; j++)

{

matran[j][2]=so[i][j];

}

check7(so);

}

}

}

void check5(const vector<string> so ){

for (int i = 0; i < so.size(); i++)

{

if(matran[2][0]==so[i][0] && matran[2][1]==so[i][1] && matran[2][2]==so[i][2]){

for (int j = 0; j < 5; j++)

{

matran[2][j]=so[i][j];

}

check6(so);

}

}

}

void check4(const vector<string> so ){

for (int i = 0; i < so.size(); i++)

{

if(matran[0][1]==so[i][0] && matran[1][1]==so[i][1] && matran[3][1]==so[i][3]){

for (int j = 0; j < 5; j++)

{

matran[j][1]=so[i][j];

}

check5(so);

}

}

}

void check3(const vector<string> so ){

for (int i = 0; i < so.size(); i++)

{

if(matran[1][0]==so[i][0] && matran[1][3]==so[i][3]){

for (int j = 0; j < 5; j++)

{

matran[1][j]=so[i][j];

}

check4(so);

}

}

}

void check2(const vector<string> so ){

for (int i = 0; i < so.size(); i++)

{

if(matran[4][0]==so[i][4] && matran[0][4]==so[i][0]){

for (int j = 0; j < 5; j++)

{

matran[j][4-j]=so[i][j];

}

check3(so);

}

}

}

void check1(const vector<string> so ){

for (int i = 0; i < so.size(); i++)

{

if(matran[0][0] == so[i][0]){

for (int j = 0; j < 5; j++)

{

matran[j][0]=so[i][j];

}

check2(so);

}

}

return;

}

int main(){

vector<string> so;

int s;

cout << "Nhap tong S: ";

cin >> s;

for (int i = 10001; i < 100000; i++)

{

if(checksnt(i) && checktong(i,s))

{

string new\_i = to\_string(i);

so.push\_back(new\_i);

}

}

for (int i = 0; i < so.size(); i++)

{

for (int j = 0; j < 5; j++)

{

matran[0][j]=so[i][j];

}

check1(so);

}

if(dem<1){

cout << "Khong co ma tran thoa man\n";

}

else cout << "So truong hop thoa man la: " << dem << endl;

}

**2.3. Kết quả**

Nhap tong S: 11

Ma tran 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

Ma tran 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

Ma tran 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

Ma tran 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

Ma tran 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

Ma tran 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

So truong hop thoa man la: 6