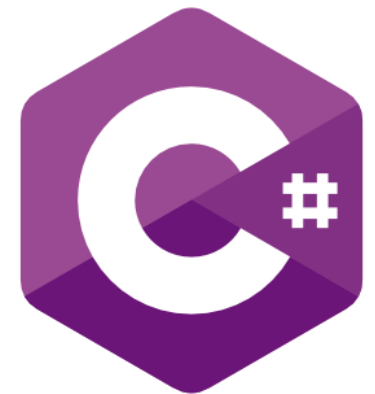


# C# 4 – Cấu trúc

Giảng viên: **ThS. Lê Thiện Nhật Quang**  
Email: [quangln.dotnet.vn@gmail.com](mailto:quangln.dotnet.vn@gmail.com)  
Website: <http://dotnet.edu.vn>  
Điện thoại: **0868.917.786**



# MỤC TIÊU

- Giải thích cấu trúc điều kiện
- Tìm hiểu cấu trúc vòng lặp
- Giải thích các câu lệnh nhảy trong C#

# ĐẶT VẤN ĐỀ

Ví dụ 1: **Nếu** ngày mai trời mưa **thì** ở nhà

Ví dụ 2: **Nếu** ngày mai trời mưa **thì** ở nhà **ngược lại** thì đi chơi

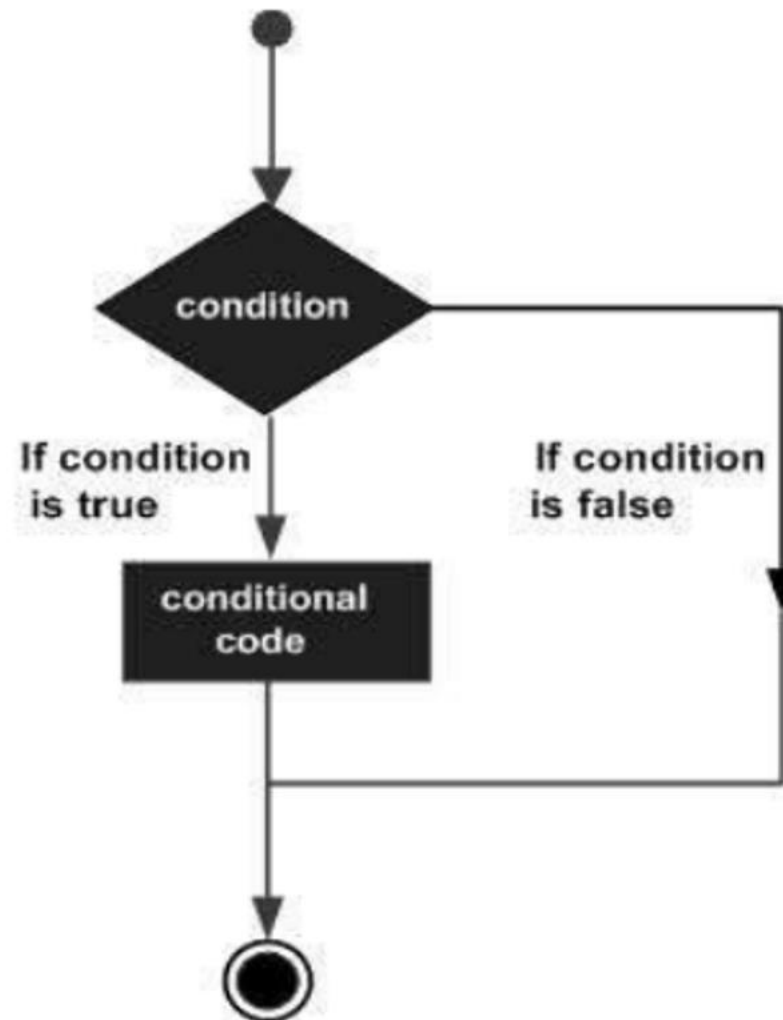
Ví dụ 3:

- **Nếu** điểm tổng kết trung bình lớn hơn hoặc bằng 8.0 **thì** xếp loại học sinh giỏi
- **Nếu** điểm tổng kết trung bình lớn hơn hoặc bằng 6.5 **thì** xếp loại học sinh tiên tiến
- **Nếu** điểm tổng kết trung bình lớn hơn hoặc bằng 5.0 **thì** xếp loại học sinh trung bình
- **Còn lại** bị ở lại lớp

Qua 3 ví dụ trên ta có thể thấy cấu trúc rẽ nhánh là cấu trúc chia bài toán thành nhiều trường hợp, rơi vào trường hợp nào thì thực hiện thao tác tương ứng với trường hợp đó.

## 1.1. LỆNH ĐIỀU KIỆN

- ❑ Được sử dụng để lập điều kiện và sẽ có những xử lý tương ứng được thực hiện tùy vào điều kiện nào được thỏa
- ❑ Câu lệnh if...else
- ❑ Câu lệnh switch...case.



## 1.2. LỆNH IF

### □ Cú pháp

```
if(<<điều kiện>>)  
{  
    << Công việc >>  
}
```

### □ Diễn giải:

❖ Nếu **điều kiện** có giá trị true thì **công việc** được thực hiện

❑ Ví dụ:

```
double diem = 4;  
if (diem >= 5) {  
    System.out.println("Đậu");  
}
```

❑ Diễn giải:

❖ Đoạn mã trên không xuất gì ra màn hình vì biểu thức điều kiện **diem >= 5** có giá trị false

## 1.3. LỆNH IF ... ELSE

### ❑ Cú pháp

```
if (<<điều kiện>>)  
{  
    << công việc 1 >>  
}  
else  
{  
    << công việc 2 >>  
}
```

### ❑ Diễn giải

- ❖ Nếu **điều kiện** có giá trị true thì **công việc 1** được thực hiện, ngược lại **công việc 2** được thực hiện

### ❑ Ví dụ

```
double diem = 4;  
if (diem < 5) {  
    System.out.println("Rót");  
}  
else {  
    System.out.println("Đậu");  
}
```

### ❑ Diễn giải:

❖ Đoạn mã trên xuất chữ “Rót” ra màn hình vì điều kiện **diem < 5** có giá trị là true.

0 references

```
static void Main(string[] args)  
{  
    string name = "Lập trình Back-end";  
    if (name == "Lập trình Back-end")  
    {  
        Console.WriteLine("Lập trình Back-end");  
    }  
    else  
    {  
        Console.WriteLine("Có khả năng là Front-end");  
    }  
}
```



## 1.4. NHIỀU LỆNH IF

### ❑ Cú pháp

```
if (<<điều kiện 1>>){  
    << công việc 1 >>  
}  
else if (<<điều kiện 2>>){  
    << công việc 2 >>  
}  
...  
else {  
    << công việc N+1 >>  
}
```

### ❑ Diễn giải

- ❖ Chương trình sẽ kiểm tra từ **điều kiện 1 đến N** nếu gặp **điều kiện i** đầu tiên có giá trị true thì sẽ thực hiện **công việc i**, ngược lại sẽ thực hiện **công việc N+1**

### ❑ Ví dụ

```
double delta = Math.pow(b, 2) - 4 * a * c;  
if(delta < 0) {  
    Console.WriteLine("Vô nghiệm");  
}  
else if(delta == 0) {  
    Console.WriteLine("Nghiem kép");  
}  
else {  
    Console.WriteLine("2 nghiệm");  
}
```

### ❑ Diễn giải

❖ Đoạn mã trên biện luận và giải phương trình bậc 2

```
static void Main(string[] args)  
{  
    int i = 20;  
    if (i == 10)  
        Console.WriteLine("i is 10");  
    else if (i == 15)  
        Console.WriteLine("i is 15");  
    else if (i == 20)  
        Console.WriteLine("i is 20");  
    else  
        Console.WriteLine("i is not present");  
}
```

# BÀI TẬP

- ❑Viết chương trình tính thuế thu nhập. Giả sử thu nhập gồm lương và thưởng
- ❑Thuế thu nhập được tính như sau
  - ❖Dưới 9 triệu: không đóng thuế
  - ❖Từ 9 đến 15 triệu: thuế 10%
  - ❖Từ 15 đến 30 triệu: 15%
  - ❖Trên 30 triệu: 20%

## 1.5. LỆNH SWITCH

### ❑ Cú pháp

```
switch (<<biểu thức>>)
```

```
{
```

```
    case <<giá trị 1>>:
```

```
        // Công việc 1
```

```
    break;
```

```
    case <<giá trị 2>>:
```

```
        // Công việc 2
```

```
    break;
```

```
    ...
```

```
    default:
```

```
        // Công việc N+1
```

```
    break;
```

```
}
```

### ❑ Diễn giải

- ❖ So sánh giá trị của biểu thức switch với giá trị của các case. Nếu bằng với giá trị của case nào thì sẽ thực hiện công việc của case đó, ngược lại sẽ thực hiện công việc của default.
- ❖ Nếu công việc của case không chứa lệnh break thì case tiếp sau sẽ được thực hiện
- ❖ default là tùy chọn

## 1.5. LỆNH SWITCH (2)

```
double a = 5, b = 7, c = -1;
```

```
char op = '+';
```

```
switch(op){
```

```
    case '+':
```

```
        c = a + b;
```

```
        break;
```

```
    case '-':
```

```
        c = a - b;
```

```
        break;
```

```
    case 'x':
```

```
    case ':':
```

```
        Console.WriteLine("Đang xây dựng");
```

```
        break;
```

```
    default:
```

```
        Console.WriteLine("Vui lòng chọn +, -, x và :");
```

```
        break;
```

```
}
```

Không có **break**

```
using System;
namespace DecisionMaking
{
    class Program
    {
        static void Main(string[] args)
        {
            int a = 100;
            int b = 200;

            switch (a)
            {
                case 100:
                    Console.WriteLine("DAY LA MOT PHAN CUA SWITCH ");
                    switch (b)
                    {
                        case 200:
                            Console.WriteLine("DAY LA MOT PHAN CUA SWITCH ");
                            break;
                    }
                    break;
            }
            Console.WriteLine("GIA TRI CUA A LA : {0}", a);
            Console.WriteLine("GIA TRI CUA B LA : {0}", b);
            Console.ReadLine();
        }
    }
}
```

```
static void Main(string[] args)
{
    Console.WriteLine("Long cac lenh switch trong C#");
    Console.WriteLine("-----");

    int a = 100;
    int b = 200;

    switch (a)
    {
        case 100:
            Console.WriteLine("Dong nay thuoc lenh switch ben ngoai!");
            switch (b)
            {
                case 200:
                    Console.WriteLine("Dong nay thuoc lenh switch ben trong!");
                    break;
            }
            break;
    }

    Console.WriteLine("Gia tri chinh xac cua a la: {0}", a);
    Console.WriteLine("Gia tri chinh xac cua b la: {0}", b);
    Console.ReadLine();

    Console.ReadKey();
}
```



# QUY TẮC NO-FALL-THROUGH

- Luồng thực thi từ câu lệnh case thứ nhất không cho phép tiếp tục câu lệnh case tiếp theo và nó đề cập như một quy tắc 'no-fall-through'
- Danh sách các câu lệnh bên trong khối case thường kết thúc bằng câu lệnh break hoặc goto, sử dụng để điều khiển chương trình thoát khỏi cấu trúc switch...case và chuyển đến câu lệnh tiếp theo.
- Khối case cuối cùng(khối default) cũng cần có câu lệnh break hoặc goto để nắm quyền điều khiển một cách rõ ràng bên ngoài cấu trúc switch...case
- C# giới thiệu quy tắc 'no-fall-through' cho phép trình biên dịch sắp xếp lại thứ tự của các khối case nhằm tối ưu hiệu suất



```
string input;
Console.WriteLine("Enter the month: ");
input = Console.ReadLine().ToUpper();
switch (input)
{
    case "JANUARY":
    case "MARCH":
    case "MAY":
    case "JULY":
    case "AUGUST":
    case "OCTOBER":
    case "DECEMBER":
        Console.WriteLine("This month has 31 days");
        break;
    case "APRIL":
    case "JUNE":
    case "SEPTEMBER":
    case "NOVEMBER":
        Console.WriteLine("This month has 30 days");
        break;
    case "FEBRUARY":
        Console.WriteLine("This month has 28 days in a non-leap year and 29 days in a leap year");
        break;
    default:
        Console.WriteLine("Incorrect choice");
        break;
}
```

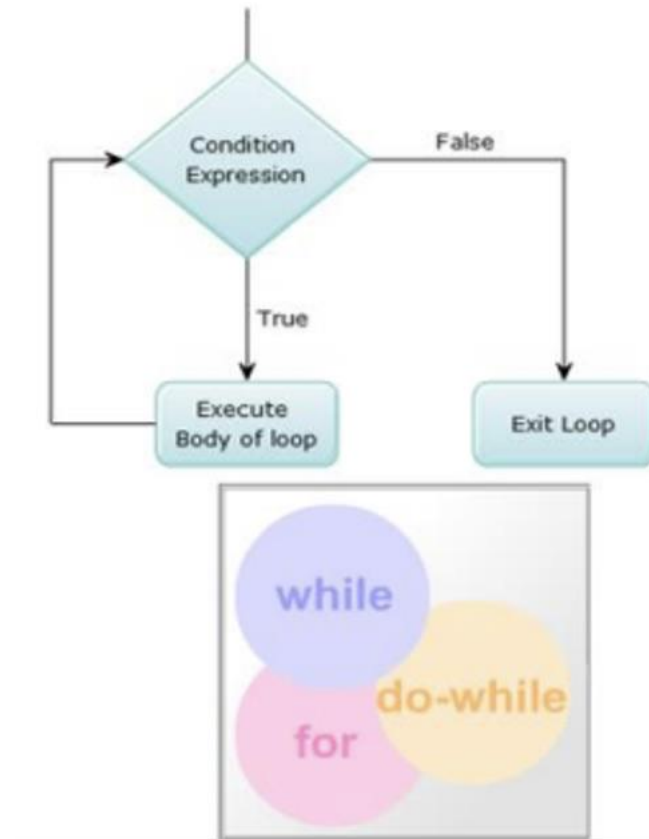
❑ Đoạn code sau cho kết quả nào?

```
string str = "hello";  
if (str)  
    System.Console.WriteLine ("The value  
is True");  
if (str == "hello")  
    System.Console.WriteLine ("The Value  
is True");
```

**Error CS0029 : Cannot implicitly convert type  
'int' to 'bool'**

## 2.1. VÒNG LẶP

- ❑ Cho phép bạn thực thi một hoặc nhiều câu lệnh lặp đi lặp lại cho đến khi thỏa điều kiện được chỉ định.
- ❑ C# hỗ trợ 4 loại vòng lặp là vòng lặp while, vòng lặp do...while, vòng lặp for và vòng lặp foreach.



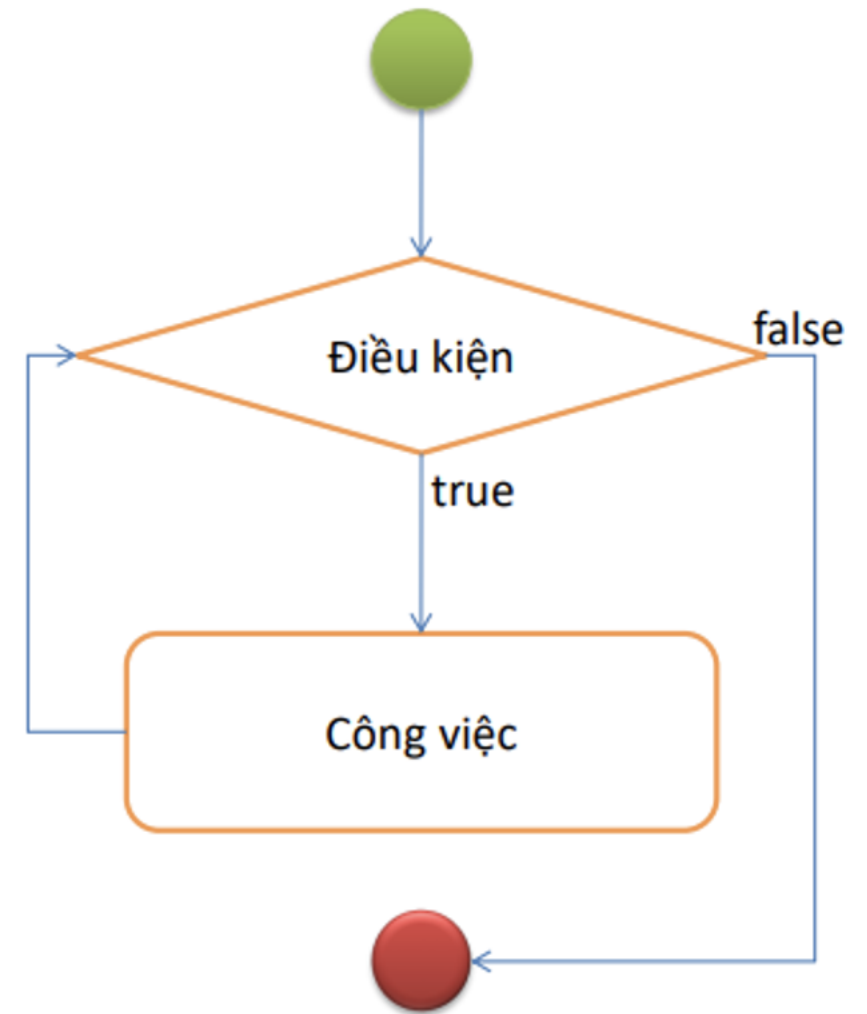
## 2.2. LỆNH LẶP WHILE

❑Cú pháp

```
while (<<điều kiện>>) {  
    // công việc  
}
```

❑Diễn giải:

❖Thực hiện công việc trong khi biểu thức điều kiện có giá trị là true.



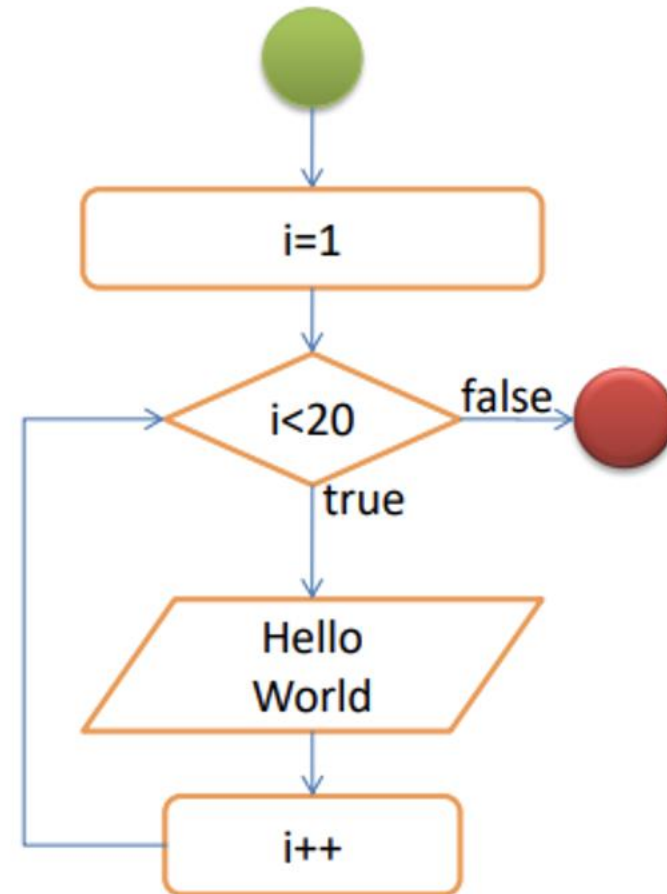
## 2.2. LỆNH LẶP WHILE (2)

### ❑ Ví dụ

```
int i = 1;  
while (i < 20) {  
    Console.WriteLine("Hello World !");  
    i++;  
}
```

### ❑ Diễn giải:

- ❖ Đoạn mã trên xuất 19 dòng Hello World ra màn hình



## 2.3. LỆNH LẶP DO ... WHILE

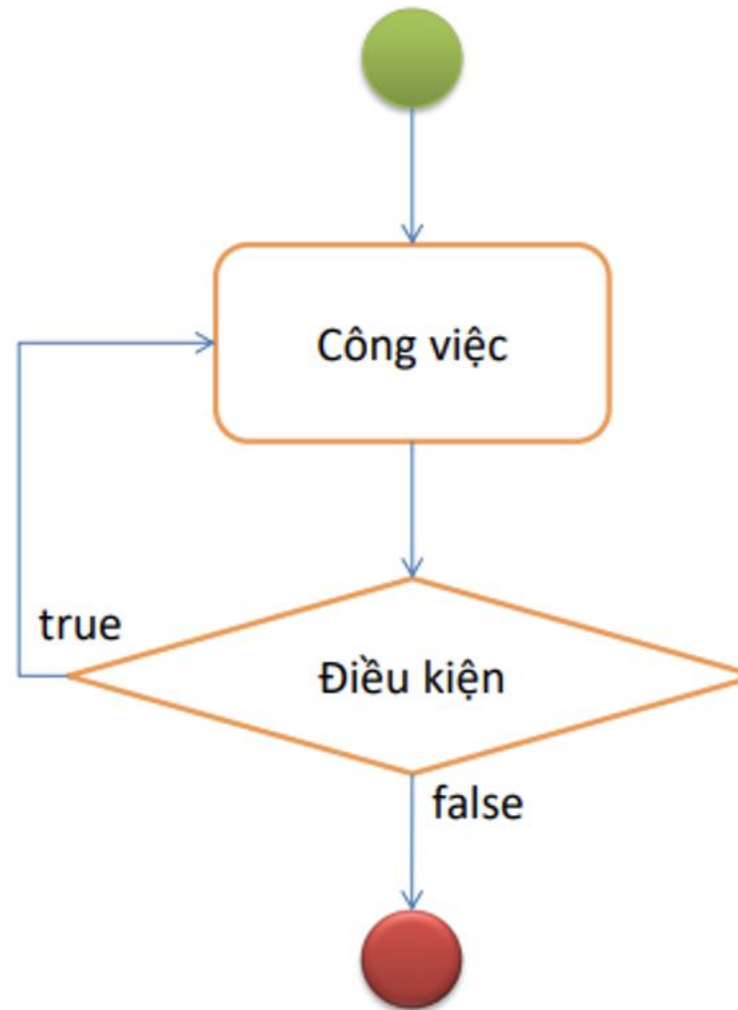
❑ Cú pháp:

```
do {  
    // công việc  
}
```

```
while (<<điều kiện>>);
```

❑ Diễn giải:

- ❖ Tương tự lệnh lặp while chỉ khác ở chỗ điều kiện được kiểm tra sau, nghĩa là công việc được thực hiện ít nhất 1 lần.



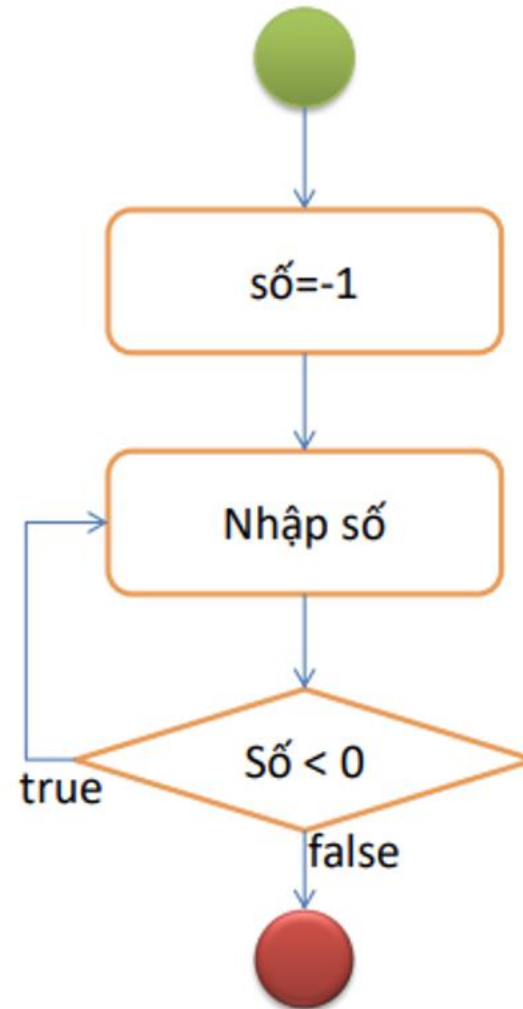
## 2.3. LỆNH LẶP DO ... WHILE (2)

### ❑ Ví dụ

```
int so = -1;  
do {  
    so = Convert.ToInt32(Console.ReadLine());  
}  
while (so < 0);
```

### ❑ Diễn giải:

- ❖ Đoạn mã trên chỉ cho phép nhập số nguyên dương từ bàn phím.



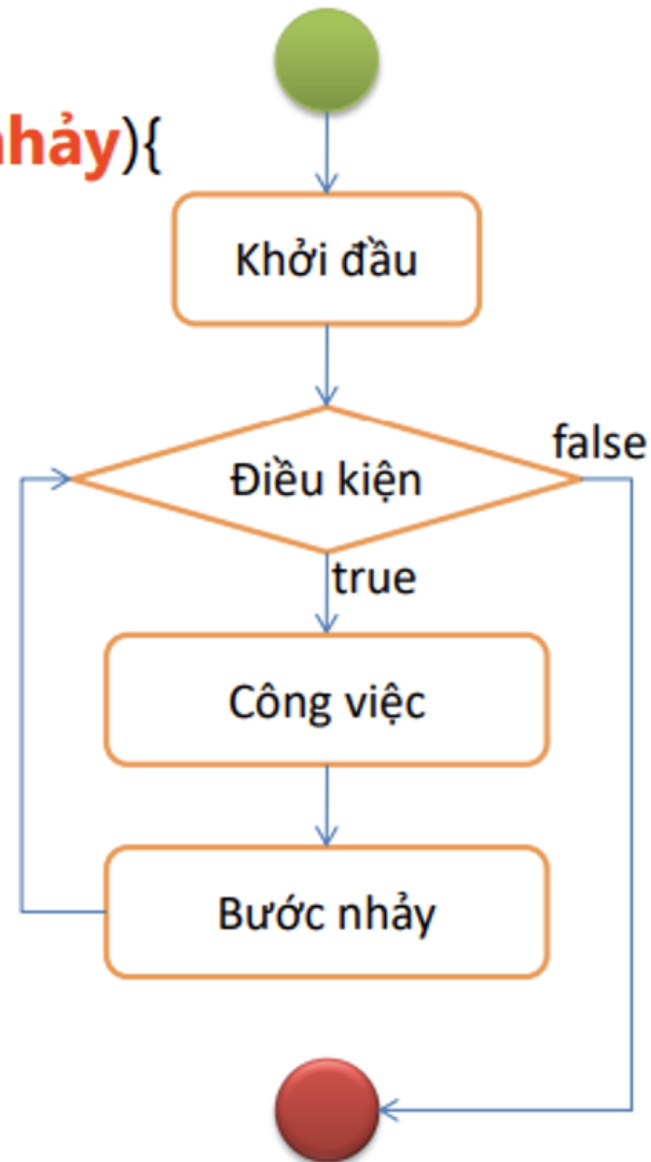
## 2.4. LỆNH LẶP FOR

### ❑ Cú pháp

```
for (khởi đầu ; điều kiện; bước nhảy){  
    // công việc  
}
```

### ❑ Diễn giải

- ❖ B1: Thực hiện <<khởi đầu>>
- ❖ B2: Kiểm tra <<điều kiện>>
  - True: B3
  - False: kết thúc
- ❖ B3: Thực hiện << công việc >>
- ❖ B4: Thực hiện <<bước nhảy>>
- ❖ B5: Trở lại B2





## 2.4. LỆNH LẶP FOR (2)

- ❑ Ví dụ vòng lặp for hiển thị dãy số giảm dần từ 10 đến 1

```
static void Main(string[] args)
{
    Console.WriteLine("For statement example");
    // int idx = 10 là initialization
    // idx > 0 là condition
    // idx-- là decrement
    for (int idx = 10; idx > 0; idx--)
    {
        Console.Write(idx + " ");
    }
}
```

```
For statement example
10 9 8 7 6 5 4 3 2 1
```

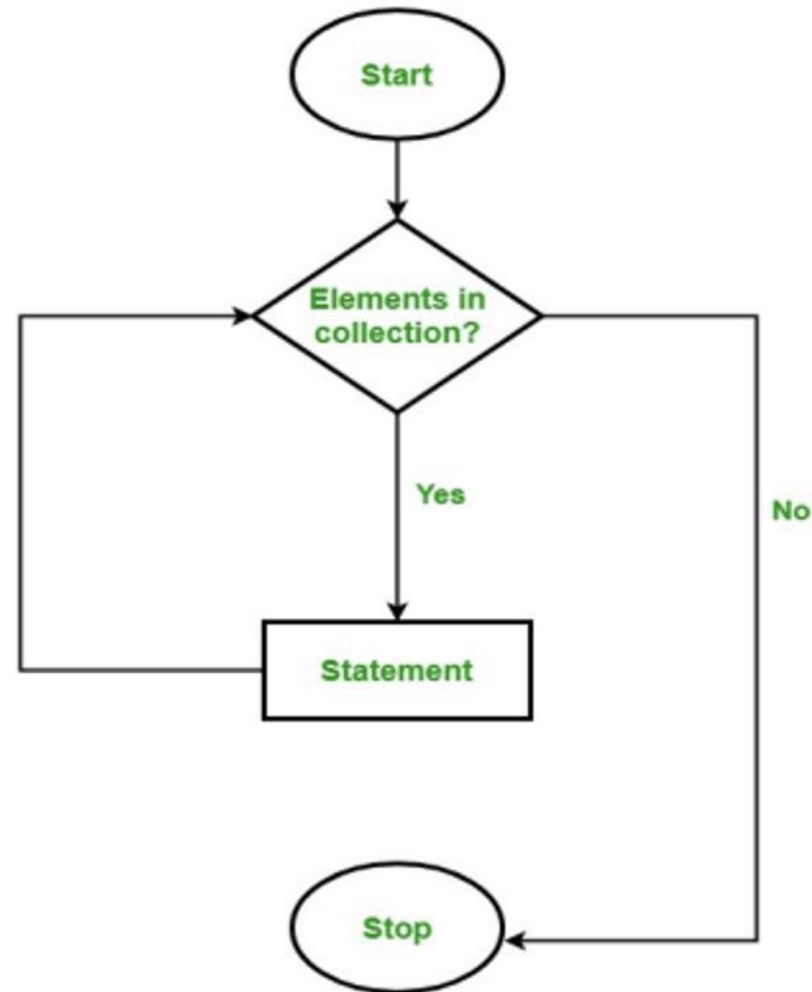
## 2.5. LỆNH LẶP FOREACH

❑ Cú pháp:

```
foreach(val in Array/Collection)  
{  
    // công việc  
}
```

❑ Diễn giải:

- ❖ Vòng lặp foreach thường được sử dụng để xử lý trên mảng hoặc trên collection để truy cập giá trị của các phần tử trong mảng hoặc collection.



## 2.5. LỆNH LẶP FOREACH (2)

- ❑ Ví dụ vòng lặp foreach hiển thị một dãy số trong mảng các số nguyên.

```
static void Main(string[] args)
{
    Console.WriteLine("Foreach statement example");
    int[] intArr = new int[10];
    Random r = new Random();
    // Khởi tạo giá trị cho các phần tử của mảng
    for (int idx = 0; idx < 10; idx++)
    {
        intArr[idx] = r.Next(1, 10);
    }
    // Hiển thị giá trị của các phần tử sử dụng foreach
    Console.Write("Value of element: ");
    foreach (int val in intArr)
    {
        Console.Write(val + " ");
    }
}
```

## 3.1. CÁC CÂU LỆNH NHẢY

Khi đang thực hiện các lệnh trong vòng lặp, có yêu cầu như sau:

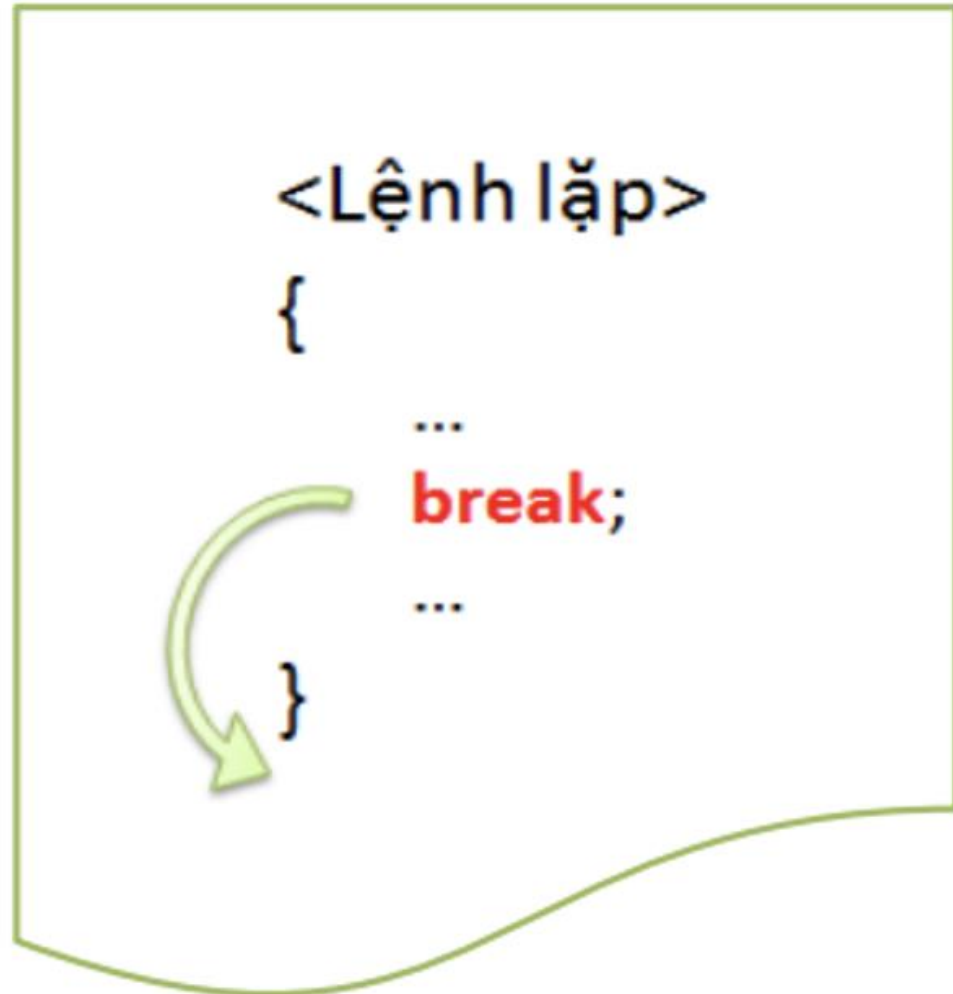
Không thực hiện các lệnh còn lại nữa mà thoát khỏi vòng lặp, hay không thực hiện các công việc còn lại của vòng lặp hiện tại mà nhảy qua vòng lặp tiếp theo.

Các câu lệnh nhảy trong C#:

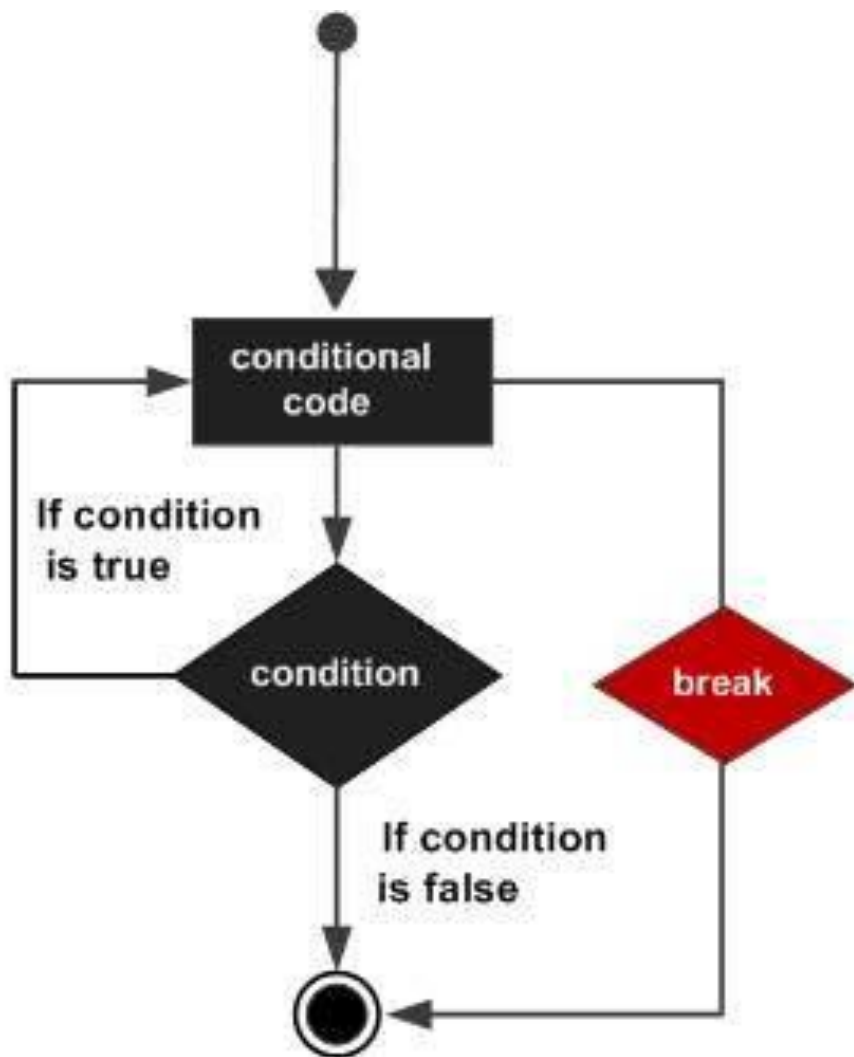
- break
- continue
- goto
- return

## 3.2. CÂU LỆNH BREAK

- ❑ Câu lệnh break được sử dụng để kết thúc vòng lặp.



```
static void Main(string[] args)
{
    int[] intArray = new int[10];
    Console.WriteLine("\nValue of element: ");
    foreach (int e in intArray)
    {
        if (e % 2 == 0)
        {
            break;
        }
        Console.Write(e + " ");
    }
}
```



```
static void Main(string[] args)
{
    // local variable definition
    int a = 10;

    // while loop execution
    while (a < 20)
    {
        Console.WriteLine("value of a: {0}", a);
        a++;

        if (a > 15)
        {
            // terminate the loop using break statement
            break;
        }
    }
    Console.ReadLine();
}
```



### 3.3. LỆNH CONTINUE

- ❑ Câu lệnh continue sẽ bỏ qua những xử lý ở sau câu lệnh continue, vòng lặp tiếp tục thực hiện lần lặp tiếp theo

<Lệnh lặp>

{

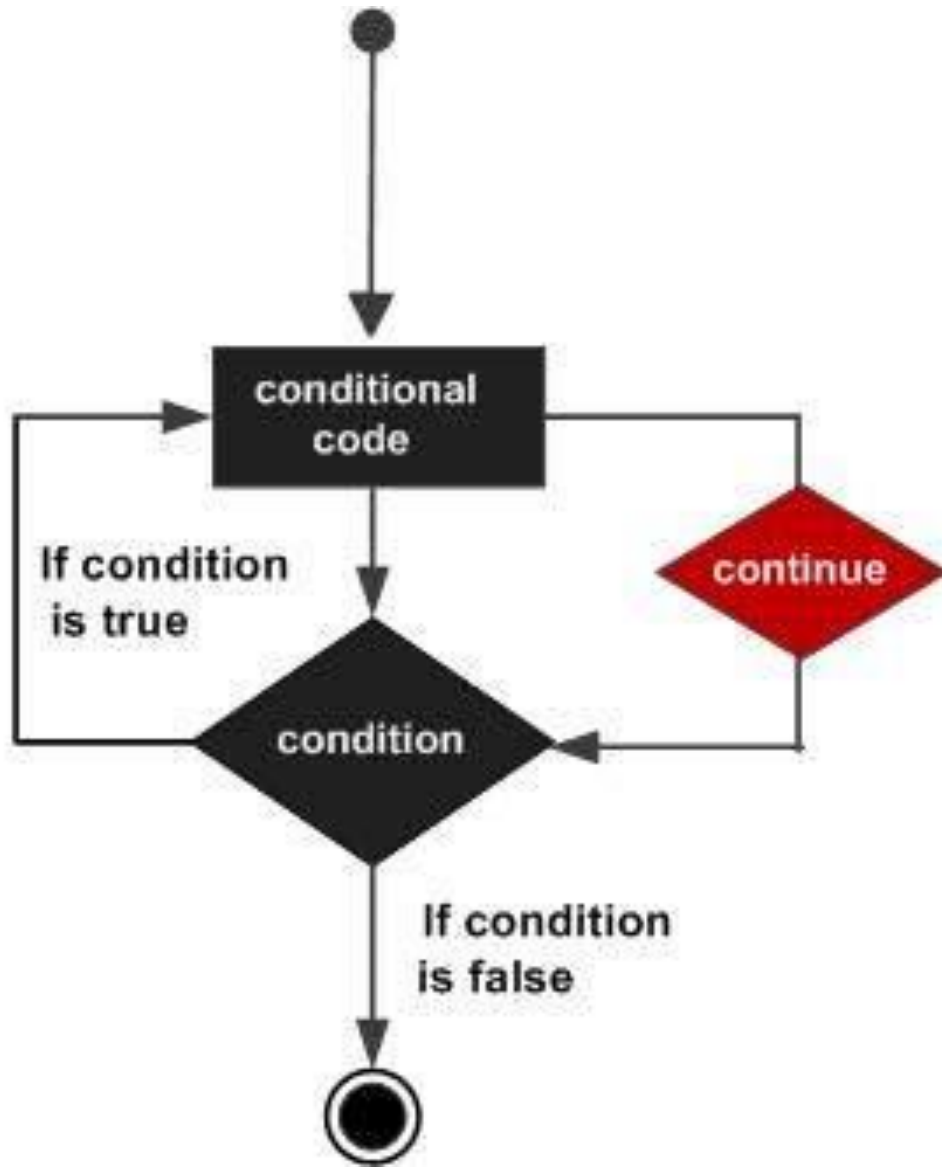
...

**continue;**

...

}

```
static void Main(string[] args)
{
    Console.WriteLine("Continue statement example");
    for (int idx = 1; idx <= 10; idx++)
    {
        if (idx % 2 == 0)
        {
            continue;
        }
        Console.Write(idx + " ");
    }
}
```



```
static void Main(string[] args)
{
    // local variable definition
    int a = 10;

    // do loop execution
    do
    {
        if (a == 15)
        {
            // skip the iteration
            a++;
            continue;
        }
        Console.WriteLine("value of a: {0}", a);
        a++;
    } while (a < 20);
    Console.ReadLine();
}
```



## 3.4. LỆNH GOTO

Cái tên goto có thể hiểu là đi đến đâu đó. Thường sử dụng cấu trúc goto người ta sẽ đi kèm một câu điều kiện (có thể không cần).

### Cú pháp:

```
goto <label>;
```

- Trong đó label là một nhãn đích đến trong code. Nơi mà code sẽ tiếp tục được thực thi từ đó.  
Cấu trúc của một label: <Tên label>:
- goto là từ khóa thông báo cho trình biên dịch biết sẽ đi đến nhãn ngay sau để tiếp tục thực thi code.

```
static void Main(string[] args)
{
    int a = 1;
    switch (a)
    {
        case 1:                // label case 1
            Console.WriteLine("Case 1");
            break;
        case 2:                // label case 2
            Console.WriteLine("Case 2");
            goto case 1;        // dịch chuyển tới label case 1
            break;              // Đoạn code này thừa vì sẽ không bao giờ thực thi
        case 3:                // label case 3
            Console.WriteLine("Case 3");
            break;
    }
    Console.ReadLine();
}
```

## 3.5. LỆNH RETURN

Câu lệnh nhảy return là câu lệnh thoát ngay ra khỏi hàm và sẽ bỏ qua tất cả các lệnh phía sau nó.

Sử dụng khi thao tác với cơ sở dữ liệu, tới file, thiết bị, cần nhờ kiểm tra sự đóng mở kết nối có hợp lý hay không tránh để tình trạng kết nối được mở ra quá nhiều nhưng lại không có sự đóng kết nối đã bị lệnh nhảy return thoát ra khỏi phương thức dẫn tới dùng 1 lúc sẽ bị treo ứng dụng.

### Cú pháp:

return <label>;

0 references

```
static void Main(string[] args)
{
    Console.WriteLine(add(1, 2));
    Console.ReadLine();
}
```

1 reference

```
static int add(int a, int b)
{
    return a + b;
}
```

0 references

```
static void Main(string[] args)
{
    in_ra(1, 10);
    Console.ReadLine();
}
```

1 reference

```
static void in_ra(int a, int b)
{
    for (int i = a; i <= b; ++i)
    {
        Console.WriteLine(i);
        if (i % 5 == 0)
        {
            return;
        }
    }
}
```

# BÀI TẬP

1. Viết chương trình cho phép giải phương trình bậc nhất trong đó các hệ số a và b nhập từ bàn phím.
2. Viết chương trình cho phép giải phương trình bậc hai trong đó các hệ số a,b,c nhập

## HƯỚNG DẪN

- ✓ Nhập a, b và c
- ✓ Xét a
  - Nếu  $a == 0 \Rightarrow$  giải phương trình bậc nhất
  - Ngược lại ( $a \neq 0$ )
    - Tính delta
    - Biện luận theo delta
      - $\Delta < 0$ : vô nghiệm
      - $\Delta = 0$ : nghiệm kép  $x = -b/(2*a)$
      - $\Delta > 0$ : 2 nghiệm phân biệt
        - $X1 = (-b + \sqrt{\Delta})/(2*a)$
        - $X2 = (-b - \sqrt{\Delta})/(2*a)$

3. Nhập vào ngày, tháng của năm hiện tại. Viết chương trình.

- Kiểm tra tính hợp lệ ngày, tháng nhập.
- Cho biết tháng nhập có bao nhiêu ngày.
- Cho biết ngày hôm sau của ngày đã nhập là ngày nào.
- Cho biết ngày hôm trước của ngày đã nhập là ngày nào

4. Viết chương trình nhập một số nguyên từ bàn phím và cho biết số đó có phải là số nguyên tố hay không (số nguyên tố là số chỉ chia hết cho 1 và chính nó)

#### HƯỚNG DẪN

- ✓ Cho một vòng lặp chạy từ 2 cho đến số nhập vào -1. Nếu có một số nhập vào chia hết cho biến chạy thì số đó không phải là số nguyên tố.

```
boolean ok = true;
```

```
for(int i=2; i < N-1; i++){ // N là số nhập từ bàn phím
```

```
    if(N % i == 0){
```

```
        ok = false;
```

```
        break;
```

```
    }
```

```
    i++;
```

```
}
```

5. Viết chương trình xuất ra màn hình bảng cửu chương

- ✓ Kiểm tra biến ok bạn sẽ biết N có phải là số nguyên tố hay không

## PhuongTrinh\_

```
1 reference
public static void giaiPtBac1(float a, float b)
{
    float x;
    if (a == 0)
    {
        if (b == 0)
        {
            Console.WriteLine("Phuong trinh co vo so nghiem");
        }
        else
        {
            Console.WriteLine("Phuong trinh vo nghiem");
        }
    }
    else
    {
        x = -b / a;
        Console.WriteLine("Phuong trinh co 1 nghiem: x = {0}", x);
    }
}
```

```
1 reference
public static void giaiPtBac1()
{
    float a, b;

    Console.Write("Nhap a: ");
    string st = Console.ReadLine();
    a = float.Parse(st);

    Console.Write("Nhap b: ");
    st = Console.ReadLine();
    b = float.Parse(st);

    giaiPtBac1(a, b);
}
```

```
namespace Lab2
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            PhuongTrinh.giaiPtBac1();

            Console.Read();
        }
    }
}
```



```

public static void giaiPtBac2(float a, float b, float c)
{
    if (a == 0)
    {
        giaiPtBac1(b, c);
    }
    else //a != 0
    {
        float delta =(float) Math.Pow(b, 2) - 4 * a * c;

        if (delta < 0)
        {
            Console.WriteLine("Phuong trinh vo nghiem");
        }else if (delta == 0)
        {
            float x = -b / 2 * a;
            Console.WriteLine("Phuong trinh co nghiem kep: x = {0}", x);
        }
        else
        {
            float x1 = (-b + (float) Math.Sqrt(delta)) / 2 * a;
            float x2 = (-b - (float) Math.Sqrt(delta)) / 2 * a;
            Console.WriteLine("Phuong trinh co 2 nghiem phan biet: x1 = {0} va x2 = {1}", x1, x2);
        }
    }
}

```

```

namespace Lab2
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            //PhuongTrinh.giaiPtBac1();

            PhuongTrinh.giaiPtBac2();

            Console.Read();
        }
    }
}

```

```

0 references
public static void giaiPtBac2()
{
    float a, b, c;

    Console.Write("Nhap a: ");
    string st = Console.ReadLine();
    a = float.Parse(st);

    Console.Write("Nhap b: ");
    st = Console.ReadLine();
    b = float.Parse(st);

    Console.Write("Nhap c: ");
    st = Console.ReadLine();
    c = float.Parse(st);

    giaiPtBac2(a, b, c);
}

```

```
10 while (i < num - 1 && isOk == true)
17 {
18     if (num % i == 0)
```

```
public static void kiemTraSoNguyenTo()
```

```
{
    int num;
    Console.WriteLine("Nhap so can kiem tra: ");
    string st = Console.ReadLine();
    num = int.Parse(st);
```

```
    if (isSoNguyenTo(num))
```

```
    {
        Console.WriteLine("So {0} la so nguyen to", num);
    }
```

```
    else
```

```
    {
        Console.WriteLine("So {0} khong phai la so nguyen to", num);
    }
}
```

0 references

```
class SoNguyenTo
```

```
{
```

1 reference

```
public static bool isSoNguyenTo(int num)
```

```
{
```

```
    int i = 2;
```

```
    bool isOk = true;
```

```
    while (i < num - 1 && isOk == true)
```

```
    {
```

```
        if (num % i == 0)
```

```
        {
```

```
            isOk = false;
```

```
        }
```

```
        i++;
```

```
    }
```

```
    return isOk;
```