



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

## TIN HỌC ĐẠI CƯƠNG

### Bài 10. Các cấu trúc lập trình trong C

#### Nội dung

- 1. Cấu trúc lệnh khối
- 2. Cấu trúc rẽ nhánh
  - 2.1. Cấu trúc if, if ... else
  - 2.2. Cấu trúc lựa chọn switch
- 3. Cấu trúc lặp
  - 3.1. Vòng lặp for
  - 3.2. Vòng lặp while
- 4. Các lệnh thay đổi cấu trúc lập trình
  - 4.1. continue
  - 4.2. break

## 1. Cấu trúc lệnh khối

- Thể hiện cấu trúc tuần tự
- Lệnh khối là dãy các câu lệnh được đặt trong cặp dấu ngoặc nhọn { }

```
{  
    lenh_1;  
    lenh_2;  
    ...  
    lenh_n;  
}
```

- C cho phép khai báo biến trong lệnh khối, nhưng phần khai báo phải nằm trước câu lệnh.

3

## 1. Cấu trúc lệnh khối (2)

- Lệnh khối lồng nhau:
  - Trong một lệnh khối có thể chứa lệnh khối khác
  - Sự lồng nhau là không hạn chế

```
{  
    lenh;  
    {  
        lenh;  
        ...  
    }  
    ...  
}
```

4

## 1. Cấu trúc lệnh khối (2)

- Ví dụ:

```
#include <conio.h>
#include <stdio.h>
int main()
//noi dung cua ham main() cung la mot khoi lenh
{
    // khai bao bien
    int c = 10;
    printf("Gia tri cua c = %d day la c ngoai",c);
    // bat dau mot khoi lenh khac
    {
        int c = 20;
        printf("\n Gia tri cua c = %d day la c trong",c);
        printf("\n Tang gia tri cua c them 10 don vi");
        c = c + 10;
        printf("\n Gia tri cua c = %d day la c trong",c);
    }
    printf("\n Gia tri cua c = %d day la c ngoai",c);
    getch();
    return 0;
} // ket thuc khoi lenh cua ham main()
```

5

## 1. Cấu trúc lệnh khối (2)

- Kết quả:

Gia tri cua c = 10 day la c ngoai  
Gia tri cua c = 20 day la c trong  
Tang gia tri cua c them 10 don vi  
Gia tri cua c = 30 day la c trong  
Gia tri cua c = 10 day la c ngoai

6

## Nội dung

### 1. Cấu trúc lệnh khởi

### → 2. Cấu trúc rẽ nhánh

2.1. Cấu trúc if, if ... else

2.2. Cấu trúc lựa chọn switch

### 3. Cấu trúc lặp

3.1. Vòng lặp for

3.2. Vòng lặp while

### 4. Các lệnh thay đổi cấu trúc lập trình

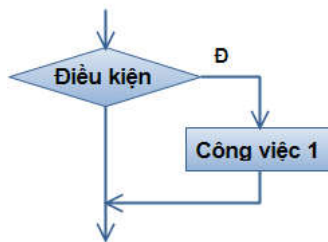
4.1. continue

4.2. break

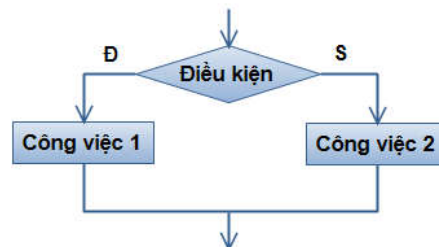
7

## Cấu trúc rẽ nhánh

- nếu biểu thức điều kiện đúng (giá trị chân lý là True) thực hiện công việc 1.



- nếu biểu thức điều kiện đúng (giá trị chân lý là True) thực hiện công việc 1.
- nếu biểu thức điều kiện sai (giá trị chân lý là False) thực hiện công việc 2.



8

## 2.1. Cấu trúc if, if ... else

- Cú pháp cấu trúc **if**

```
if (bieu_thuc_dieu_kien)
    lenh; //bieu_thuc_dieu_kien đúng
```

- Cú pháp cấu trúc **if ... else**

```
if (bieu_thuc_dieu_kien)
    lenh_1; //bieu_thuc_dieu_kien đúng
else
    lenh_2; // bieu_thuc_dieu_kien sai
```

9

## Biểu thức điều kiện

- **bieu\_thuc\_dieu\_kien**

– Biểu thức logic : !, &&, ||

– Biểu thức quan hệ : ==, !=, >, >=, <, <=

- Biểu thức số học

`if (bieu_thuc_so_hoc)` tương đương với  
`if (bieu_thuc_so_hoc != 0)`

– Phức hợp

- Kiểm tra điều kiện bộ 3 số (a,b,c) là số đo 3 cạnh của một tam giác

```
(a < b + c) && (b < c + a) && (c < a + b)
```

- Kiểm tra với 3 cạnh (a,b,c), tam giác có cân không ?

```
(a == b) || (b == c) || (c == a)
```

10

## Ví dụ

- Tìm số lớn hơn trong 2 số thực  $a$  và  $b$ :

```
#include <conio.h>
#include <stdio.h>
int main()
{
    // khai bao bien
    float a, b, max;
    printf(" Nhap gia tri a va b: ");
    scanf("%f %f",&a,&b);
    if(a < b) max = b;
    else max = a;
    printf("\n So lon hon trong 2 so %f va %f la %f",
           a, b, max);

    getch();
    return 0;
} //ket thuc ham main()
```

- Kết quả:

Nhap vao 2 gia tri a va b: 23 247

So lon nhat trong hai so 23 va 247 la 247

11

## Kết hợp lệnh khối

```
if (bieu_thuc_dieu_kien)
{
    Khoi_lenh_1;
}
else
{
    Khoi_lenh_2;
}
```

12

## Ví dụ

```
//Khai báo tập tiêu đề
#include <stdio.h>
#include <conio.h>
//nội dung chương trình chính
int main (){
    float x,y; //khai báo 2 biến đầu vào
    float thuong; //Khai báo thuong 2 số
    if (y==0)
        printf("Loi chia cho 0");
    else {
        thuong = x/y;
        printf("Ket qua la : %f",thuong);
    }
    getch();
    return 0;
}
```

13

## Cấu trúc if...else if

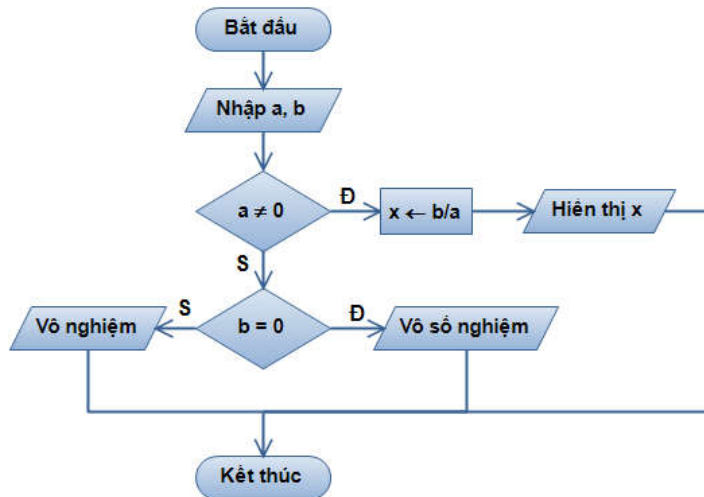
- “Rẽ” nhiều nhánh

```
if (bieu_thuc_dieu_kien_1)
{
    Khoi_lenh_1;
}
else if (bieu_thuc_dieu_kien_2)
{
    Khoi_lenh_2;
}
...
else if (bieu_thuc_dieu_kien_n)
{
    Khoi_lenh_n;
}
else
{
    Khoi_lenh_cuoi_cung;
}
```

14

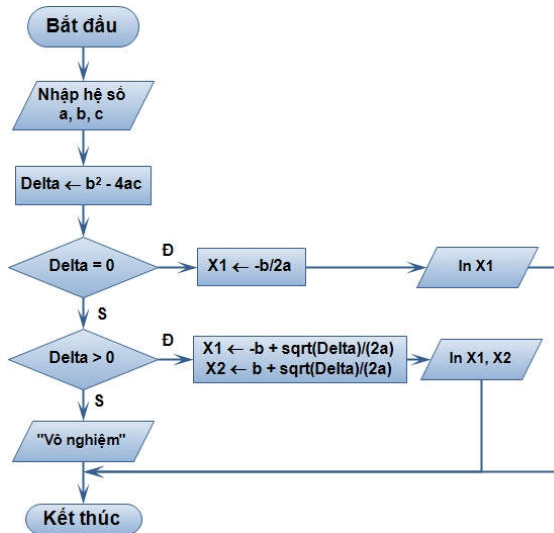
## Bài tập – Giải phương trình bậc nhất

- Giải phương trình  $ax + b = 0$



15

## Bài tập - Giải phương trình bậc 2



1. Giải PT bậc 2 trên tập số thực  $ax^2 + bx + c = 0$  ( $a \neq 0$ )
2. Cải tiến 1 : giải trên tập số phức
3. Cải tiến 2 : kiểm tra điều kiện của a

16



## Chú ý

### • Tương đương ?

```
if (dieu_kien)
    Cong_viec_1;
else Cong_viec_2;
```



```
if (dieu_kien)
    Cong_viec_1;
if (! dieu_kien)
    Cong_viec_2;
```

```
if (dieu_kien_1)
    if (dieu_kien_2)
        Cong_viec_1;
else
    Cong_viec_2;
```



?

```
if (dieu_kien_1){
    if (dieu_kien_2)
        Cong_viec_1;
    else
        Cong_viec_2;}
}
```



```
if (dieu_kien_1){
    if (dieu_kien_2)
        Cong_viec_1;}
else
    Cong_viec_2;
```

17

## Bài tập

Viết chương trình có những chức năng sau:

### 1. Hiện thị một bảng chọn sau

```
-----CHƯƠNG TRÌNH TÍNH TOÁN VỚI SỐ NGUYÊN DƯƠNG-----
1. Tổng 2 số (a+b)
2. Hiệu 2 số (a-b)
3. Tích 2 số (a*b)
4. Thương 2 số (a/b)
5. UCLN của 2 số
6. BCNN của 2 số
```

2. Cho phép người dùng nhập 2 số nguyên dương
3. Cho phép người dùng đưa ra lựa chọn 1 chức năng. Tùy theo lựa chọn của người dùng, hãy thực hiện công việc tương ứng. nếu người dùng nhấn 1 phím khác hãy thông báo và dừng chương trình.

18

## 2.2. Cấu trúc lựa chọn switch

- Cú pháp cấu trúc **switch**

```
switch (bieu_thuc)
{
    case gia_tri_1: lenh_1; break;
    case gia_tri_2: lenh_2; break;
    ...
    case gia_tri_n: lenh_n; break;
    default: lenh_n+1; break;
}
```

- Giá trị của biểu thức kiểm tra (bieu\_thuc) phải là số nguyên:
  - Phải có kiểu dữ liệu là **char**, **int**, **long**.
- Tương ứng các giá trị sau **case** (gia\_tri\_1, gia\_tri\_2,...) cũng phải là số nguyên.
- **Case** chỉ đóng vai trò là “nhãn”, lệnh **break** được sử dụng để thoát ra khỏi cấu trúc **switch**

19

## 2.2. Cấu trúc lựa chọn switch

- Ví dụ:

- Trong một năm các tháng có 30 ngày là 4, 6, 9, 11 còn các tháng có 31 ngày là 1, 3, 5, 7, 8, 10, 12. Riêng tháng hai có thể có 28 hoặc 29 ngày.
- Hãy viết chương trình nhập vào 1 tháng, sau đó đưa ra kết luận tháng đó có bao nhiêu ngày.

20

## 2.2. Cấu trúc lựa chọn switch (6)

```
#include <conio.h>
#include <stdio.h>
int main ()
{
    int thang; clrscr();
    printf("\n nhap vao thang trong nam ");
    scanf("%d",&thang);
    switch(thang)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            printf("\n Thang %d co 31 ngay ",thang);
            break;
```

21

## 2.2. Cấu trúc lựa chọn switch (7)

```
        case 4:
        case 6:
        case 9:
        case 11:
            printf("\n Thang %d co 30 ngay ",thang);
            break;
        case 2:
            printf ("\ Thang 2 co 28 hoac 29 ngay");
            break;
        default :
            printf("\n Khong co thang %d", thang);
            break;
    }
    getch();
    return 0;
}
```

22

## Bài tập

- **Bài 1:** Viết chương trình kiểm tra bộ 3 số dương  $a, b, c$  có phải là số đo 3 cạnh tam giác cân không?
- **Bài 2:** Viết chương trình giải hệ phương trình bậc nhất dạng:

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$

- **Bài 3:** Giải phương trình bậc 2 không suy biến trong tập số phức.

23

## Bài tập (tiếp)

**Bài 4:** Để đánh giá mức độ gầy hay béo của một người, các nhà khoa học sử dụng chỉ số BMI được tính theo công thức  $BMI = \text{Cân nặng}/(\text{Chiều cao})^2$

Trong đó Chiều cao tính bằng đơn vị mét. Ví dụ một người cao 1.70 m, nặng 65kg có chỉ số  $BMI = 65/1.70^2 = 22.5$ .

Kết quả đánh giá như sau:

Chỉ số	Đánh giá
$BMI < 18$	người gầy
$18 \leq BMI < 25$	Cân nặng bình thường
$25 \leq BMI < 30$	Thừa cân
$BMI \geq 30$	Béo phì

Viết chương trình kiểm tra chỉ số BMI của một người.

24

## Bài tập (tiếp)

**Bài 5:** Đơn giá điện sinh hoạt được tính theo bảng sau

Mức tiêu thụ (kWh)	Giá tiền
0 đến 50	1000 đồng
51 đến 100	1200 đồng
101 đến 150	1500 đồng
Trên 150	2000 đồng

Viết chương trình tính số tiền điện phải trả của một gia đình trong tháng.

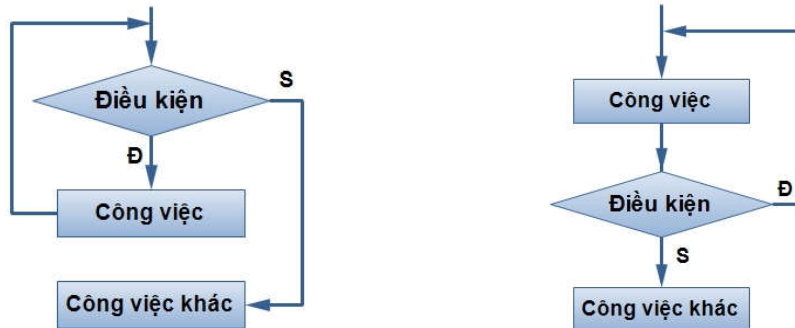
25

## Nội dung

1. Cấu trúc lệnh khối
2. Cấu trúc rẽ nhánh
  - 2.1. Cấu trúc if, if ... else
  - 2.2. Cấu trúc lựa chọn switch
- 3. Cấu trúc lặp
  - 3.1. Vòng lặp for
  - 3.2. Vòng lặp while
4. Các lệnh thay đổi cấu trúc lặp trình
  - 4.1. continue
  - 4.2. break

26

## Cấu trúc lặp



- nếu biểu thức điều kiện còn đúng, thực hiện công việc trong vòng lặp
- Thực hiện công việc trong vòng lặp 1 lần  
→ Khi biểu thức điều kiện còn đúng, thực hiện công việc trong vòng lặp

27

## 3.1. Vòng lặp while và do...while

- Mục đích:
  - Dùng để thực hiện lặp đi lặp lại một công việc trong khi còn thỏa mãn điều kiện nào đó.
- 2 cấu trúc:

```
while (biểu_thức_điều_kiện)
{
    Khôi_lệnh;
}
```

hoặc

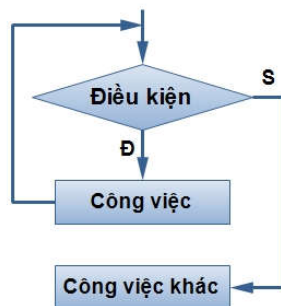
```
do
{
    Khôi_lệnh;
} while (biểu_thức_điều_kiện);
```

28

## 3.2. Vòng lặp while và do ... while (2)

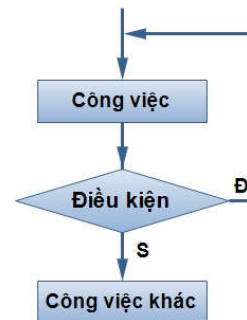
- **while:**

- Kiểm tra giá trị của **biểu\_thức\_điều\_kiện** trước rồi mới thực hiện **Khoi\_lệnh**
- Các câu lệnh có thể không được thực hiện lần nào.



- **do{...} while:**

- Thực hiện **Khoi\_lệnh** trước rồi mới kiểm tra **biểu\_thức\_điều\_kiện** của vòng lặp.
- Các câu lệnh được thực hiện **ít nhất 1 lần**.



29

## 3.2. Vòng lặp while và do...while (3)

- **biểu\_thức\_điều\_kiện**

- Biểu thức logic : `!`, `&&`, `||`
- Biểu thức quan hệ : `==`, `!=`, `>`, `>=`, `<`, `<=`
  - Biểu thức số học  
`while (biểu_thức_số_học)` tương đương với  
`while (biểu_thức_số_học != 0)`
- Phức hợp
- Sau mỗi bước lặp, kiểm tra lại giá trị của **biểu\_thức\_điều\_kiện**

30

## 3.2. Vòng lặp while (4)

- Ví dụ: nhập vào điểm của một sinh viên, nếu điểm đó không  $\in [0, 10]$  thì thông báo cho người dùng nhập lại.
- nhận xét:
  - nếu dùng lệnh `if`  $\rightarrow$  Chỉ kiểm tra được 1 lần
  - Chưa biết trước số lần lặp, người dùng được nhập ít nhất 1 lần

31

## 3.1. Vòng lặp while (5)

- Sử dụng cấu trúc `do {...} while`

```
do {
    printf("nhap diem (0<=diem<=10):");
    scanf("%f",&diem);
    if ((diem < 0) || (diem > 10))
        printf("\nBan nhap khong dung! Hay nhap lai\n");
} while ((diem < 0) || (diem > 10));
```

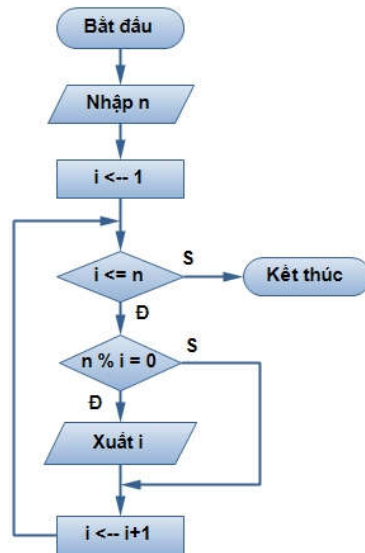
- Sử dụng cấu trúc `while {...}`

```
//nhập lần đầu tiên
printf("nhap diem (0<=diem<=10):");
scanf("%f",&diem);
//kiểm tra
while ((diem < 0) || (diem > 10)){
    printf("\nBan nhap khong dung! Hay nhap lai\n");
    printf("nhap diem(0<=diem<=10):");
    scanf("%f",&diem);
}
```

32



## Ví dụ



- Cài đặt thuật toán
- Bổ sung 1 : kiểm tra giá trị n
- Bổ sung 2 : cho người dùng thực hiện lại cho đến khi nhấn phím 'X'
- Bổ sung 3 : Kiểm tra tính hoàn hảo của n

33

## Thảo luận

- Vòng lặp vô hạn : Khi nào xảy ra? Tác hại?
  - Trong một số trường hợp cần dùng `while (1) {...}` thì trong vòng lặp phải có các câu lệnh thay đổi cấu trúc để thoát ra ngoài.
- Vòng lặp hữu hạn : làm thế nào để điều kiện dừng xảy ra ?

- nhận xét

```
while(dieu_kien1 || dieu_kien2) { }
```

tương đương với

```
while (1) {  
if (! (dieu_kien1 || dieu_kien2) exit;  
}
```

```
while(dieu_kien1 && dieu_kien2) {}
```

Tương đương với ?

34

## 3.2. Vòng lặp for

- Mục đích
  - Dùng để thực hiện lặp đi lặp lại một công việc nào đó (thông thường với số lần lặp xác định)
- Cú pháp:

```
for(bieu_thuc_1;bieu_thuc_2;bieu_thuc_3)
{
    Khoi_lenh;
}
```
- Trong đó:
  - **bieu\_thuc\_1**: Khởi tạo giá trị ban đầu cho vòng lặp
  - **bieu\_thuc\_2**: Điều kiện tiếp tục vòng lặp
  - **bieu\_thuc\_3**: Thay đổi giá trị biến điều khiển vòng lặp
  - Chú ý các biểu thức 1, 2, 3 có thể có hoặc không

35

## 3.2. Vòng lặp for (2)

- Ví dụ: Đưa ra màn hình các số nguyên lẻ nhỏ hơn 100

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int i;
    for(i = 1;i < 100;i++)
    {
        if(i%2 == 1) printf("%5d",i);
        if((i+1)%20 == 0) printf("\n");
    }
    getch();
    return 0;
}
```

36

## 3.2. Vòng lặp for (3)

- Ví dụ: Đưa ra màn hình các số nguyên lẻ nhỏ hơn 100

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int i;
    for(i = 1; i < 100; i+=2)
    {
        printf("%5d", i);
        if((i+1)%20 == 0) printf("\n");
    }
    getch();
    return 0;
}
```

37

## 3.2. Vòng lặp for (4)

- Kết quả thực hiện

```
1   3   5   7   9  11  13  15  17  19
21  23  25  27  29  31  33  35  37  39
41  43  45  47  49  51  53  55  57  59
61  63  65  67  69  71  73  75  77  79
81  83  85  87  89  91  93  95  97  99
```

38

## Nội dung

1. Cấu trúc lệnh khối
2. Cấu trúc rẽ nhánh
  - 2.1. Cấu trúc if, if ... else
  - 2.2. Cấu trúc lựa chọn switch
3. Cấu trúc lặp
  - 3.1. Vòng lặp for
  - 3.2. Vòng lặp while
- 4. Các lệnh thay đổi cấu trúc lặp trình
  - 4.1. continue
  - 4.2. break

39

## 4. Các lệnh thay đổi cấu trúc lặp trình

- Đối với các lệnh lặp:
  - **while**, **do{...} while**, hoặc **for**
- Thay đổi việc thực hiện lệnh trong vòng lặp → C cung cấp 2 lệnh:
  - **continue**;
  - **break**;

40

## 4. Các lệnh thay đổi cấu trúc lặp trình

- **continue**

- Bỏ qua việc thực hiện các câu lệnh nằm sau lệnh **continue** trong thân vòng lặp.
- Chuyển sang thực hiện một vòng lặp mới

- **break**

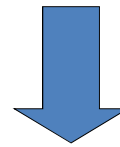
- Thoát khỏi vòng lặp ngay cả khi biểu thức điều kiện của vòng lặp vẫn còn được thỏa mãn.

41

## 4. Các lệnh thay đổi cấu trúc lặp trình

- Ví dụ:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int i;
    for(i = 1;i<=10;i++)
    {
        if(i == 5) continue;
        printf("%5d",i);
        if(i==7) break;
    }
    getch();
    return 0;
}
```



```
C:\ D:\FIT-HUT\Lectures\NHAPMO-...
1 2 3 4 6 7_
```

42

## 4. Các lệnh thay đổi cấu trúc lập trình

- Ví dụ: nhập vào 1 số nguyên. Kết luận số đó là số nguyên tố hay là hợp số?

43

```
#include <stdio.h>
#include <math.h>
int main()
{
    int n, i, kiemTra = 1;
    printf("\nnhap gia tri n : "); scanf("%d", &n);
    if (n<2) printf("\nSo %d khong la so nguyen to va
                    khong la hop so.", n);
    else
    {
        for (i=2; i<n; i++)
            if (n%i == 0)
            {
                kiemTra = 0;
                break;
            }
        if (kiemTra)
            printf("\nSo %d la so nguyen to.", n);
        else
            printf("\nSo %d la hop so.", n);
    }
    getch();
    return 0;
}
```

44

```
#include <stdio.h>
#include <math.h>
int main()
{
    int n, i, kiemTra = 1;
    printf("\nnhap gia tri n : "); scanf("%d", &n);
    if (n<2) printf("\nSo %d khong la so nguyen to va
                    khong la hop so.", n);
    else
    {
        for (i=2; i<=(int)sqrt(n); i++)
            if (n%i == 0)
            {
                kiemTra = 0;
                break;
            }
        if (kiemTra)
            printf("\nSo %d la so nguyen to.", n);
        else
            printf("\nSo %d la hop so.", n);
    }
    getch();
    return 0;
}
```

45

# Câu hỏi ?

46