

CHƯƠNG TRÌNH CON

1. Viết hàm để xác định số nhỏ hơn trong 2 số, sau đó sử dụng hàm này để xác định số nhỏ hơn trong 3 số.

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

```
//prototype của hàm, nên khai báo để code được rõ ràng
//Prototype gồm: kiểu_trả_về_của_hàm tên_hàm các_đối_số
//Lưu ý cách khai báo các đối số
int isBigger(int, int);
```

```
int main (int argc, char* argv[])
{
    int a, b, c;
    cout << "nhap 3 so a, b, c" << endl;
    cout << "a = ";
    cin >> a;
    cout << "b = ";
    cin >> b;
    cout << "c = ";
    cin >> c;

    int max = isBigger(a, b);

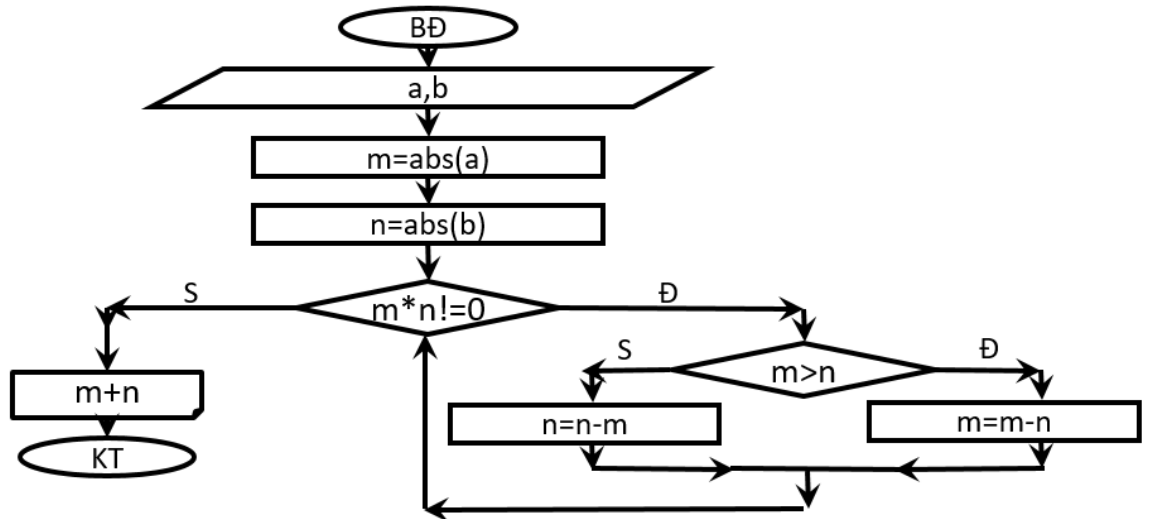
    if (max < c)
        max = c;

    cout << "So lon nhat trong 3 so la " << max << endl;

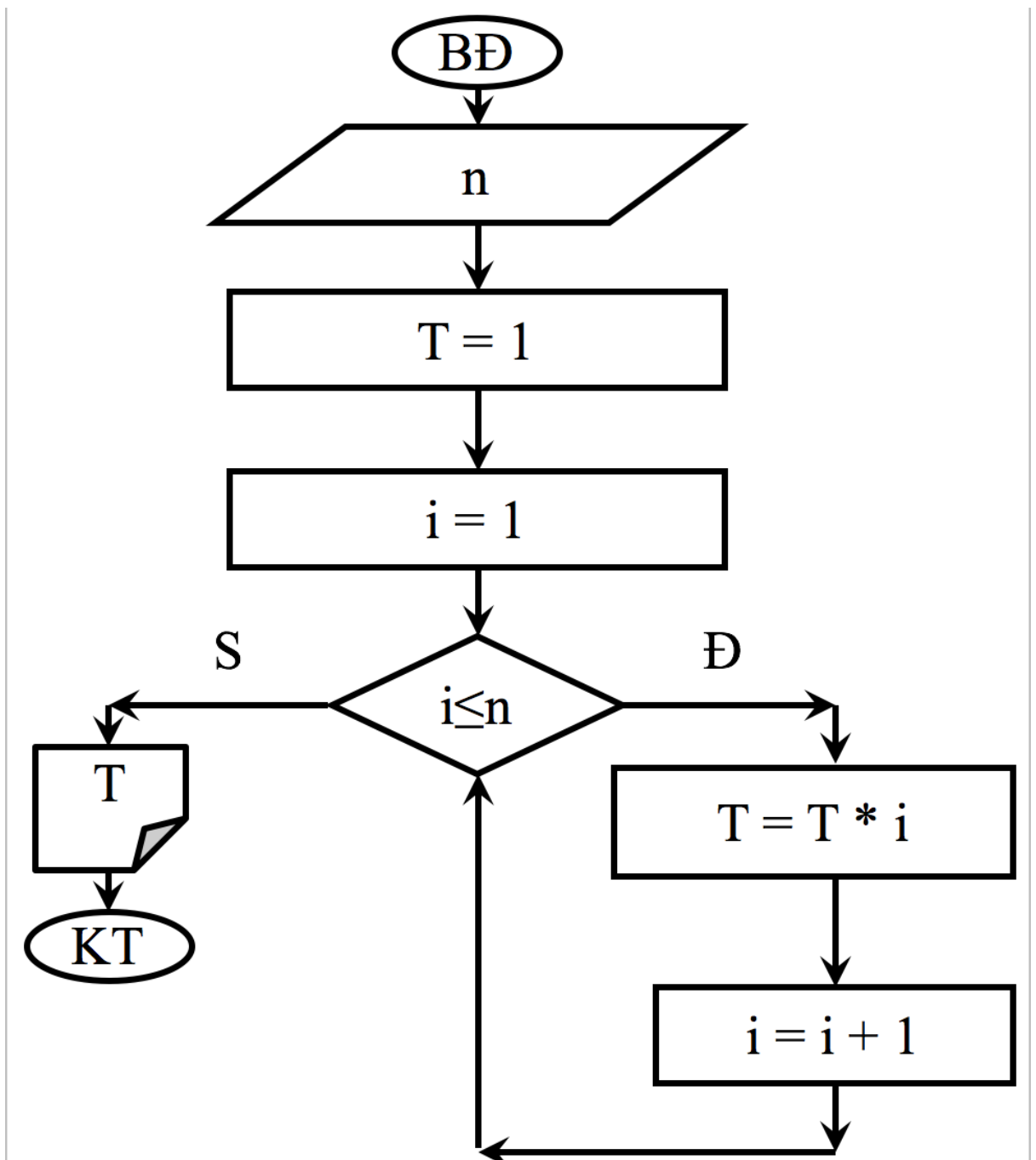
    system("pause");
    return 0;
}

int isBigger(int m, int n){
    if (m > n)
        return m;
    else
        return n;
}
```

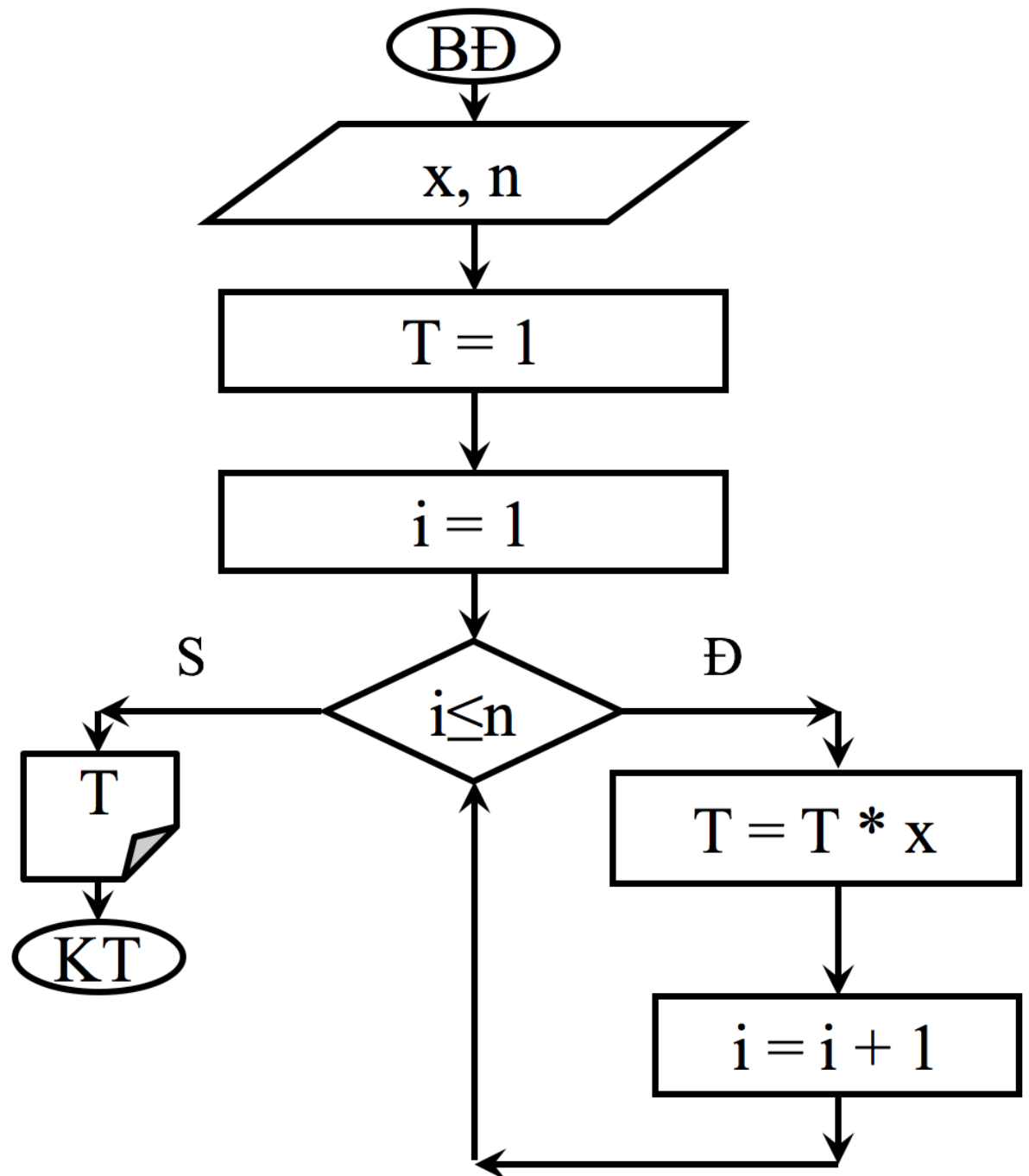
2. Viết hàm tính ước số chung lớn nhất và bội số chung nhỏ nhất của hai số nguyên dương a, b.



3. Viết hàm tính giá trị $n!$, với n là số nguyên dương và $n > 1$.



4. Viết hàm tính X^n



5. Viết hàm tính tổ hợp, trong đó cần cài đặt hàm tính giai thừa $n!$.

$$C(n, k) = \frac{n!}{k!(n - k)!}$$

6. Viết hàm tính chu vi và diện tích hình chữ nhật khi biết độ dài 2 cạnh.
Sau đó vẽ hình chữ nhật ra màn hình bằng các dấu *. Hàm tính chu vi, diện tích và hàm vẽ hình chữ nhật phải độc lập nhau.

```

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

```

7. Viết chương trình con xuất ra tam giác Pascal như sau:
Tam giác Pascal chứa các số là khai triển của nhị thức Newton

$$(x + y)^n = \sum_{k=0}^n \binom{n}{k} x^k y^{n-k}$$

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

1: (x+y)⁰

1 1: (x + y)¹

1 2 1: (x + y)²

1 3 3 1: (x + y)³

1 4 6 4 1: (x + y)⁴

8. Viết hàm nhập vào tháng bằng số rồi in ra tên tháng bằng chữ ra màn hình.
9. Viết hàm kiểm tra một ngày nào đó có hợp lệ hay không, kiểm tra năm nhuận.

10. Viết hàm đổi ngày tháng năm thành thứ trong tuần.
11. Viết hàm để nhận biết một số nguyên dương có phải là số nguyên tố hay không.
12. Viết chương trình in ra tất cả các số nguyên tố nhỏ hơn số nguyên dương M cho trước (sử dụng hàm kiểm tra số nguyên tố đã cài đặt ở trên).
13. Viết hàm kiểm tra một số nguyên dương có phải là số chính phương hay không. Xuất tất cả các số chính phương trong khoảng A, B .