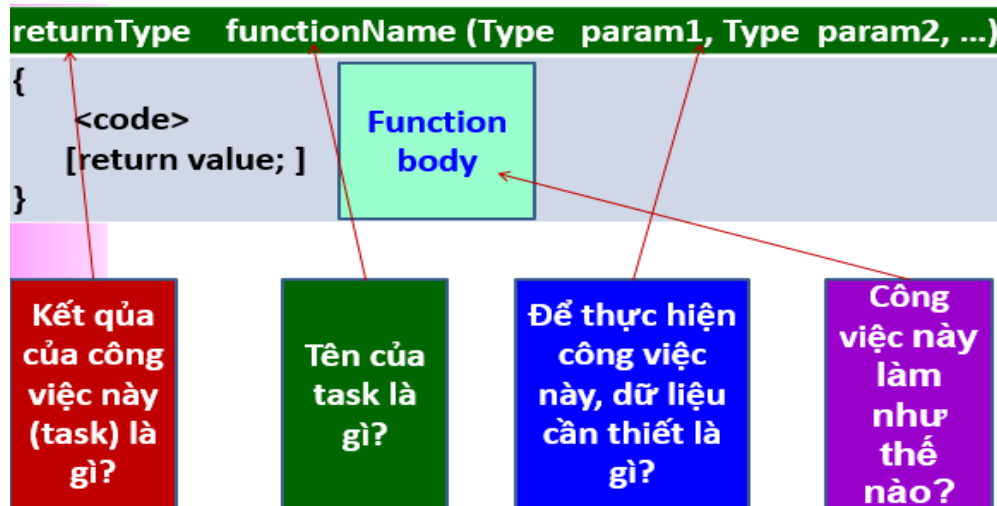


BÀI THỰC HÀNH SỐ 4: HÀM - USER-DEFINED FUNCTIONS

Syntax of function prototype



I. Hàm trả về giá trị (dùng để: tìm kiếm, tính, đếm)

Example 1: Hàm tính tổng 2 số nguyên – addNumbers

```

1. #include <stdio.h>
2. int addNumbers(int a, int b);           // function prototype
3.
4. int main()
5. {
6.     int n1,n2,sum;
7.
8.     printf("Enters two numbers: ");
9.     scanf("%d %d",&n1,&n2);
10.
11.     sum = addNumbers(n1, n2);           // function call
12.     printf("sum = %d",sum);
13.
14.     return 0;
15. }
16.
17. int addNumbers(int a, int b)           // function definition
18. {
19.     int result;
20.     result = a+b;
21.     return result;                     // return statement
22. }

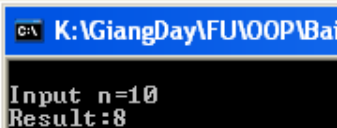
```

Example 2: Hàm tính tổng các ước số của số nguyên dương N: **sumDivisors**

```

#include <stdio.h>
int sumDivisors(int N)
{
    int i, S=0;
    for (i=1; i<=N/2; i++)
        if (N%i==0) S+=i;
    return S;
}
int main()
{
    int n, sum;
    printf("\nInput n=");
    scanf("%d", &n);
    sum = sumDivisors(n);
    printf("Result:%d\n", sum);
    getchar();
    getchar();
}

```



II. Hàm dùng để kiểm tra đúng sai (trả về true/false)

Example: Hàm kiểm tra số nguyên dương **n** có phải là số nguyên tố không: **isPrime**

```

#include <stdio.h>
int isPrime(int n)
{
    for(int i=2; i<=n/2; i++)
        if(n%i==0)
            return 0;
    return 1;
}
int main()
{
    int n;
    do
    {
        printf("Nhap so nguyen duong:");
        scanf("%d", &n);
    }
    while(n<=0);

    if( isPrime(n) )
        printf("%d la so nguyen to\n", n);
    else
        printf("%d khong phai so nguyen to\n", n);
    return 0;
}

```

III. Hàm không trả về giá trị: (dùng trong các trường hợp khác)

Example: Hàm xuất các ước số của số nguyên N.

```
#include <stdio.h>
void printDivisors(int N)
{
    int i;
    for (i=1; i<=N/2; i++)
        if (N%i==0) printf("%d, ", i);
}
int main()
{
    int n, i;
    printf("\nInput n=");
    scanf("%d", &n);
    printDivisors(n);
    printf("\nInput n=");
    scanf("%d", &n);
    printDivisors(n);
    printf("\nInput n=");
    scanf("%d", &n);
    printDivisors(n);
    getchar();
    getchar();
}
```

parameter (points to `int N`)

argument (points to `n` in `scanf` and `printDivisors`)

Function Implementation (points to the body of `printDivisors`)

Using function (points to the calls to `printDivisors`)

Output of the program:

```
Input n=10
1, 2, 5,
Input n=15
1, 3, 5,
Input n=20
1, 2, 4, 5, 10, _
```

Bài tập thực hành

Bài 1

Viết chương trình nhập số nguyên dương n, tính giai thừa của n.

$$n! = 1*2*3*...*n$$

Main

Begin

do

{

nhập n;

}

while (n<0);

P= factorial(n);

Xuất P;

End.

double factorial (int n)

```
{
    double p=1;
    int i;
    for (i=2; i<=n; i++)
        p *= i;
    return p;
}
```

Bài 2

Dãy số Fibonacci: 1 1 2 3 5 8 13 21 34 ...n

Two first numbers: 1

Others: Its value is the sum of 2 previous numbers

Viết chương trình tìm số hạng thứ n của dãy số Fibonacci

Main

```
Begin
do
{
    nhập n;
}
while (n<1);
Xuất fibo(n);
End.
```

double fibo (int n)

```
{
    int t1=1, t2=1, f=1, i ;
    for ( i = 3, i<=n; i++)
    {
        f= t1 + t2;
        t1= t2;
        t2=f;
    }
    return f;
}
```

Bài 3

Viết chương trình tính tổng các chữ số của số nguyên n. Chương trình sẽ thực hiện lặp lại cho đến khi nhập n<0 thì dừng.

Main

Begin

do

{

nhập n;

if (n>=0)

{

s = sumDigits(n);

xuất s;

}

}

while (n>=0);

End

int sumDigits (int n)

{

int sum=0; /* khởi tạo tổng */

Do

{

int soDu = n%10 ; /* Get a digit at unit position */

n = n/10;

sum += soDu;

}

while (n>0);

return sum;

}

Bài 4

Viết chương trình nhập 2 số nguyên dương, tìm ước số chung lớn nhất (**gcd - greatest common divisor**) và bội số chung nhỏ nhất (**lcm - least common multiple**) của chúng.

<u>a</u>	<u>b</u>	<u>a</u>	<u>b</u>
14	21	13	8
14	7	5	8
7	7	5	3
		2	3
		2	1
		1	1

int gcd(int a, int b)

```
{
    while ( a != b )
        if (a>b) a -=b;
        else b -= a;
    return a;
}
```

int lcm (int a, int b)

```
{
    return a*b/ gcd(a,b);
}
```

Main

Begin

```
do
{
    nhập a, b;
}
while ( a<=0 or b <=0);
d = gcd(a,b);
m = lcm (a.b);
xuất d;
xuất m;
```

End

Bài 5

Viết chương trình nhập vào số nguyên dương n ($n \geq 2$), in ra các số nguyên tố từ 2 đến n.

Main

```

Begin
do
{
    nhập n;
}
while ( n<2);
For ( i=2 to n )
    If (i là số nguyên tố )
        Xuất i;
End

```

int isPrime(int n)

```

{
    int m = sqrt(n);
    int i;
    if (n<2) return 0;
    for ( i=2; i<=m; i++)
        if (n%i==0)
            return 0 ; /* n không phải là số nguyên tố */
    return 1; /* n là số nguyên tố */
}

```

Bài 6

Viết chương trình nhập vào ngày/tháng/năm, kiểm tra ngày tháng năm đã nhập có hợp lệ hay không.

Lưu ý: Năm nhuận (y): (y%400==0 || (y%4==0 && y%100!=0))

Main

```

Begin
    Nhập d, m, y;
    If (validDate(d,m,y))
        Xuất "valid date"
    Else
        xuất "invalid date"
End

```

int validDate (int d, int m, int y)

```

{
    int maxd = 31; /*max day of months 1, 3, 5, 7, 8, 10, 12 */
    if ( d<1 || d>31 || m<1 || m>12) return 0;
    /* update maxd of a month */
    if ( m==4 || m==6 || m==9 || m==11) maxd=30;
    else if (m==2)

```

```

{
    /* leap year? */
    if ( y%400==0 || ( y%4==0 && y%100!=0) maxd=29;
    else maxd=28;
}
return d<=maxd;
}

```

Bài 7

Viết chương trình nhập tọa độ của một điểm và bán kính của đường tròn có tâm (0,0), xét vị trí tương đối của điểm so với đường tròn.

Main

Begin

```

    Nhập x, y;
do
{
    nhập r;
}
while(r<0);
result = getRelPos(x,y,r);
if (result ==1)
    xuất "The point is in the circle";
else if (result==0)
    xuất "The point is on the circle";
else
    xuất "The point is out of the circle";

```

End

int getRelPos (double x, double y, double r)

```

{
    double d2=x*x + y*y; /* d2= x2+ y2 */
    double r2= r*r;      /* r2 */
    if (d2<r2)
        return 1 ; /* d2<r2 → the point is in the circle */
}

```



```
else if (d2==r2)
```

```
    return 0 ; /*  $d^2=r^2 \rightarrow$  the point is on the circle */
```

```
    return -1 ; /*  $d^2 > r^2 \rightarrow$  the point is out of the circle */
```

```
}
```

Bài 8

Viết chương trình nhập số nguyên không âm n, tìm chữ số nhỏ nhất và lớn nhất của n.

Example: n= 10293 \rightarrow Print out 9, 0

```
void printMinMaxDigits( int n)
```

```
{
    int digit;
    int min, max ;
    remainder = n% 10;
    n=n/10; /* 1029 */
    min=max=remainder;*/
    while (n>0)
    {
        remainder = n%10; /* Get the next digit */
        n=n/10;
        if (min > remainder)
            min=remainder; /* update results */
        if (max < remainder)
            max=remainder;
    }
    Xuất min, max;
}
```

Main

```
Begin
```

```
Do
```

```
{
```

```
    Nhập n;
```

```
    printMinMaxDigits(n);
```

```
}
```

```
While (n<0);
```

```
End
```

Bài 9

Viết chương trình nhập vào tọa độ của 3 điểm trong mặt phẳng Oxy. Nếu 3 điểm lập thành tam giác, hãy tính diện tích của tam giác đó.

Hướng dẫn:

- Viết hàm tính khoảng cách của 2 điểm: A(X_A, Y_A), B(X_B, Y_B)
- Viết hàm kiểm tra độ dài 3 cạnh có lập thành tam giác hay không?

- Viết hàm tính diện tích khi biết độ dài của 3 cạnh.

Bài 10

Viết chương trình nhập số nguyên $N > 1$ và số thực x . Tính:

$$e^x \approx S = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Hướng dẫn:

- Viết hàm tính $n!$ (**xem lại bài cũ**)
- Viết hàm tính e mũ x:

```
double tinhEmuX( double x, int n)
{
    double s = 0;
    lặp i từ 0 đến n
        s += xi / giaithua(i);
    return s;
}
```

Bài 11

Viết chương trình dùng menu sau:

- 1- So nguyên to
 - 2- In chu so min, max
 - 3- Thoat
- Chon chuc nang:

- 1- Khi chọn 1, chương trình cho phép nhập một số nguyên dương, kiểm tra số đã nhập có phải là số nguyên tố hay không.
- 2- Khi chọn 2, chương trình cho phép nhập vào 1 số nguyên dương, thông báo chữ số nhỏ nhất và lớn nhất của số đã nhập.
- 3- Chương trình sẽ thoát khi chọn 3.

Functions:

int prime(int n) → xem lại bài cũ

void printMinMaxDigits(int n) → xem lại bài cũ

```
Begin
Do
{
    Print out "1- So nguyen to \n";
    Print out "2- In chu so min, max.\n";
    Print out "3- Thoat\n";
    Print out "Chon chuc nang:";
    scanf(choice);

    switch(choice)
    {
        case 1:
            do
            {
                Nhập n;
```

```
        }
        while(n<0);
        If ( prime(n)==1)
            Xuất “n la so nguyen to\n”;
        Else
            Xuất “ n khong phai la so nguyen to\n”;
        break;
    case 2:
        do
        {
            Nhập n;
        }
        while(n<0);
        printMinMaxDigits( int n) ;
        break;
    }
}
while ( choice >0 & choice<3);
End
```

Bài 12

Viết chương trình dùng menu

Cho người dùng chọn thực hiện 1 trong các công việc sau:

- 1- Kiểm tra ký tự có phải là nguyên âm hay không.
- 2- Tính tổng các ước số của n.
- 3- Kiểm tra một số nguyên có phải là số chính phương hay không.