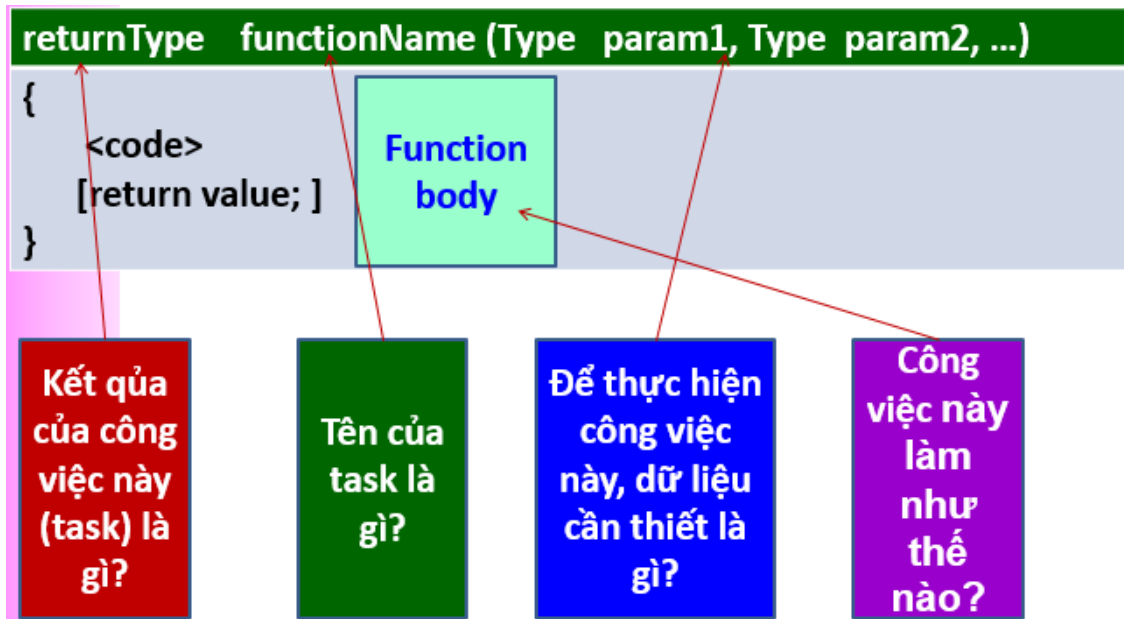


## LAB 8\_9: HÀM - FUNCTIONS



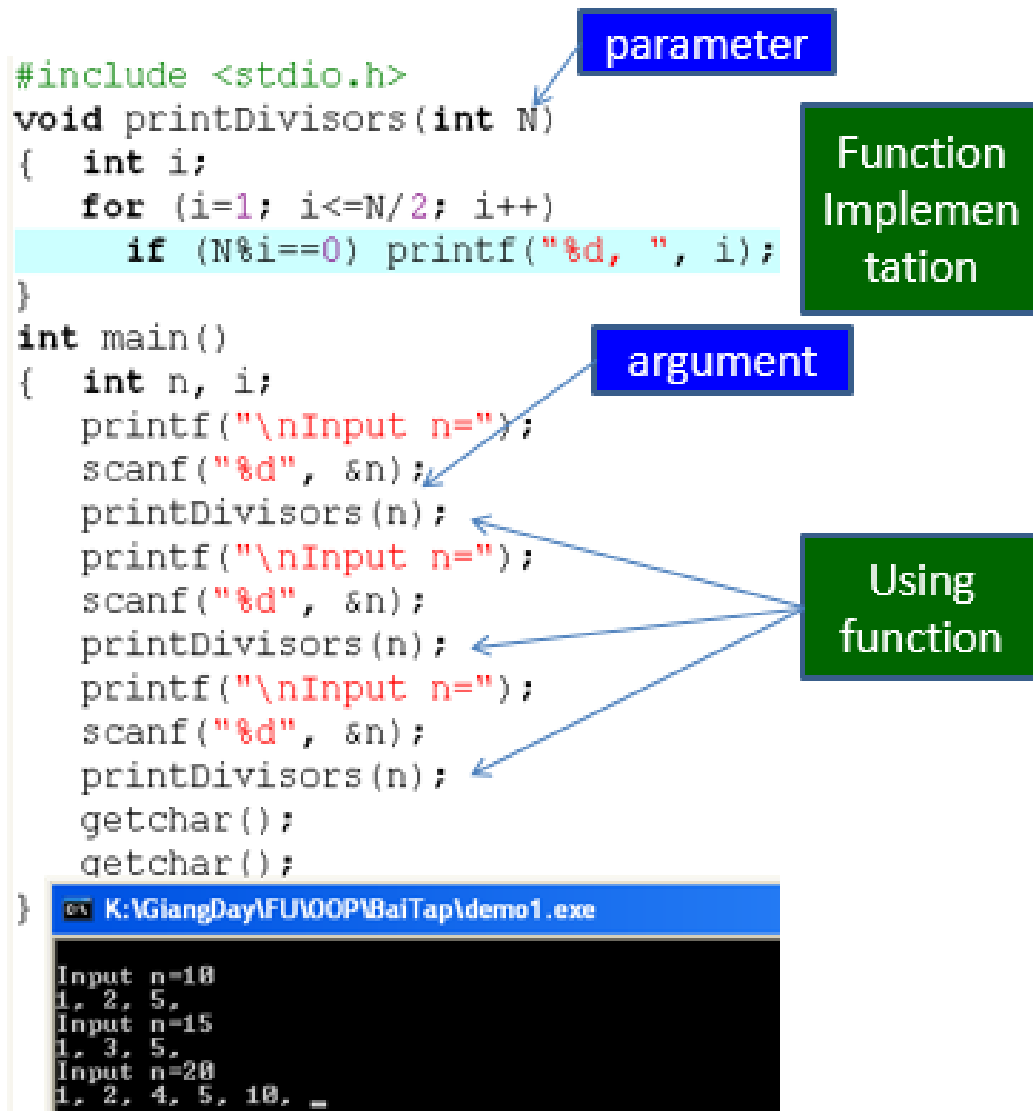
### I. Hàm trả về giá trị

**Ví dụ:** Viết chương trình tính tổng các ước số của số nguyên dương.

```
#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 void

**Ví dụ:** Viết chương trình in ra các ước số của một số nguyên dương được nhập từ bàn phím.



### III. Hàm kiểm tra

**Ví dụ:** Viết chương trình in ra **n** số nguyên tố đầu tiên.

```

1 #include <stdio.h>
2 int isPrime(int n)
3 {   int result=1, i;
4     for (i=2; i*i<=n && result==1; i++)
5         if (n%i==0) result=0;
6     return result;
7 }
8 void print_n_Primes(int n)
9 {   int count = 0; /* count primes printed */
10    int value =2; /* current value is considered */
11    while (count<n)
12    {   if (isPrime(value)==1)
13        {   printf("%d ", value);
14            count++;
15        }
16        value++;
17    }
18 }
19 int main()
20 {   int n;
21     printf("Input number of primes:");
22     scanf("%d",&n);
23     print_n_Primes(n);
24     getchar();
25     getchar();
26     return 0;
27 }

```

#### IV. Bài tập:

##### Bài 1

<b>Problem</b>	Viết chương trình nhập vào số nguyên dương $n$ ( $n \geq 2$ ), in ra các số nguyên tố từ 2 đến $n$ .
<b>Analysis</b> <i>Nouns:</i> <i>positive integer</i> $\rightarrow$ <i>int n</i>	<b>Suggested algorithm (logical order of verbs)</b> Begin Do { Accept $n$ ; } While ( $n < 2$ ); For ( $i = 2$ to $n$ ) If ( $i$ is a prime ) Print out $i$ ; $\rightarrow$ <b>Function int prime (int i)</b> End
<b>Algorithm for checking whether an integer is a prime or not</b>	<pre> <b>int prime( int n ) {</b>     <b>int m = sqrt(n);</b> /* m: square root of n */     <b>int i;</b> /* variable having value from 2 to m */     <b>if (n&lt;2) return 0;</b> /* Condition 1 is not satisfied */     <b>for ( i=2; i&lt;=m; i++)</b> /* checking the second condition */         <b>if (n%i==0) return 0 ;</b> /* n is divided by i <math>\rightarrow</math> n is not a prime */     <b>return 1;</b> /* n is a prime */ <b>}</b> </pre>

## Bài 2

<b>Related knowledge</b>	Năm nhuận (y): $(y \% 400 == 0 \parallel (y \% 4 == 0 \ \&\& \ y \% 100 != 0))$
<b>Problem</b>	Viết chương trình nhập vào ngày/tháng/năm, kiểm tra ngày tháng năm đã nhập có hợp lệ hay không.
<b>Analysis</b> Data of a day → int d, m, y	<b>Suggested algorithm (logical order of verbs)</b> Begin Accept d, m, y If (valid(d,m,y)) print out “valid date” Else print out “invalid date” End
<b>Algorithm for checking whether a date is valid or not</b>	<pre>int validDate ( int d, int m, int y) {     int maxd = 31; /*max day of months 1, 3, 5, 7, 8, 10, 12 */     /* basic checking */     if ( d&lt;1    d&gt;31    m&lt;1    m&gt;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 &amp;&amp; y%100!=0) maxd=29;         else maxd=28;     }     return d&lt;=maxd; }</pre>

## Bài 3

<b>Problem</b>	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.
<b>Analysis</b> Nouns: A point → double x,y A circle → double r Relative position → int result → -1: (x,y) is out of the circle → 0: (x,y) is on the circle → 1: (x,y) is in the circle	<b>Suggested algorithm (logical order of verbs)</b> Begin Accept x, y; Do { Accept r; } While(r<0); result = getRelPos(x,y,r); if (result ==1) Print out “The point is in the circle”; else if (result==0) Print out “The point is on the circle”; else Print out “The point is out of the circle”; End
<b>Algorithm for getting relative position of a point with a circle</b>	<pre>int getRelPos ( double x, double y, double r) {     double d2=x*x + y*y; /* d<sup>2</sup>= x<sup>2</sup>+ y<sup>2</sup> */     double r2= r*r; /* r<sup>2</sup>*/     if (d2&lt;r2) return 1 ; /* d<sup>2</sup>&lt;r<sup>2</sup> →the point is in the circle */     else if (d2==r2) return 0 ; /* d<sup>2</sup>=r<sup>2</sup> →the point is on the circle */     return -1 ; /* d<sup>2</sup>&gt; r<sup>2</sup> →the point is out of the circle */ }</pre>

#### Bài 4

<b>Problem</b>	Viết chương trình nhập số nguyên dương n, tính giai thừa của n. $n! = 1*2*3*...*n$
<b>Analysis</b> A positive integer → int n	<b>Suggested algorithm (logical order of verbs)</b> Begin Do { Accept n; } While (n<0); Print out factorial(n); End.
<b>Algorithm for Computing factorial of an integer</b>	<b>double factorial ( int n) {</b> double p=1; int i; for (i=2; i<=n; i++) p *= i; return p; }

#### Bài 5

<b>Related knowledge</b>	Fibonacci sequence: 1 1 2 3 5 8 13 21 34 ... Two first numbers: 1 Others: Its value is the sum of 2 previous numbers
<b>Problem</b>	Viết chương trình in ra số hạng thứ n của dãy số Fibonacci
<b>Analysis</b> A position → int n	<b>Suggested algorithm (logical order of verbs)</b> Begin Do { Accept n; } While (n<1); Print out fibo(n); End.
<b>Algorithm for Computing the n<sup>th</sup> value of the Fibonacci sequence</b>	<b>double fibo ( int n) {</b> int t1=1, t2=1, f=1, i ; for ( i = 3, i<=n; i++) { f= t1 + t2; t1= t2; t2=f; } return f; }

#### Bài 6

<b>Problem</b>	Viết chương trình tính tổng các số hạng của số nguyên n được nhập.
<b>Analysis</b> Sum → int S=0 Accepted integer → int n	<b>Suggested algorithm (logical order of verbs)</b> Begin Do { Accept n; If (n>=0)

	<pre>         {             S = sumDigits(n);             Print out S;         }     }     While (n&gt;=0); End </pre>
<b>Algorithm for</b> <b>Computing sum of</b> <b>digits of a</b> <b>nonnegative integer</b>	<pre> <b>int sumDigits (int n)</b> {     int sum=0; /* initialize sum of digits */     Do     {         int remainder = n%10 ; /* Get a digit at unit position */         n = n/10;         sum += remainder;     }     while (n&gt;0);     return sum; } </pre>

## Bài 7

<b>Related knowledge</b>	<p>Find out the greatest common divisor (<b>gcd</b>) and least common multiple (<b>lcm</b>) of two positive integers: <i>Find out gcd of a and b</i></p> <table><tr><td>a</td><td>b</td><td>a</td><td>b</td></tr><tr><td>14</td><td>21</td><td>13</td><td>8</td></tr><tr><td>14</td><td>7</td><td>5</td><td>8</td></tr><tr><td>7</td><td>7</td><td>5</td><td>3</td></tr><tr><td></td><td></td><td>2</td><td>3</td></tr><tr><td></td><td></td><td>2</td><td>1</td></tr><tr><td></td><td></td><td>1</td><td>1</td></tr></table> <pre>int gcd( int a, int b) {     while ( a != b )         if a&gt;b then a -=b;         else b -= a;     return a; } int lcm ( int a, int b) {     return a*b/ gcd(a,b); }</pre>	a	b	a	b	14	21	13	8	14	7	5	8	7	7	5	3			2	3			2	1			1	1
a	b	a	b																										
14	21	13	8																										
14	7	5	8																										
7	7	5	3																										
		2	3																										
		2	1																										
		1	1																										
<b>Problem</b>	Viết chương trình nhập 2 số nguyên dương, tìm ước số chung lớn nhất ( <b>gcd</b> ) và bội số chung nhỏ nhất ( <b>lcm</b> ) của chúng.																												
<b>Analysis</b> Two integers → int a, b gcd → int d lcm → int m	<b>Suggested algorithm (logical order of verbs)</b> Begin Do { Accept a, b; } While ( a<=0 OR b <=0); d = <b>gcd</b> (a,b); m = <b>lcm</b> (a,b); Print out d; Print out m; End																												

## Bài 8

<b>Related knowledge</b>	<p><b>Print out the minimum and the maximum digits of a nonnegative integer integer</b></p> <p>Example: <math>n = 10293 \rightarrow</math> Print out 9, 0</p> <pre><b>void printMinMaxDigits( int n)</b> { int digit; /* Variable for extracting 1 digit */   int min, max ; /* Result variables */   digit = n% 10; /* get the first rightmost digit: 3 */   n=n/10; /* 1029, the remainder needs to proceed after*/   min=max=remainder; /* initialize results */   while (n&gt;0)   { digit = n%10; /* Get the next digit */     n=n/10;     if (min &gt; remainder) min=remainder; /* update results */     if (max &lt; remainder) max=remainder;   }   Print out min, max; }</pre>
<b>Problem</b>	<p>Viết chương trình nhập số nguyên không âm, in ra chữ số nhỏ nhất và lớn nhất.</p>
<p><b>Analysis</b> Noun: A integer <math>\rightarrow</math> int n</p>	<p><b>Suggested algorithm (logical order of verbs)</b></p> <pre>Begin   Do     { Accept n;       printMinMaxDigits(n);     }   While (n&lt;0); End</pre>