

## BÀI THỰC HÀNH SỐ 8 :: HÀM (FUNCTION)

### MỤC TIÊU

1. Hiểu bản chất đơn vị chương trình hàm. Khai thác các hàm có sẵn trong các thư viện chuẩn; thư viện toán học và thư viện chuỗi.
2. Viết được nguyên mẫu hàm, định nghĩa hàm và sử dụng (gọi) hàm
3. Phân tích ngữ cảnh bài toán phù hợp và cần thiết xây dựng hàm

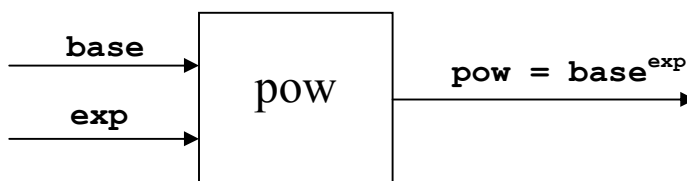
### NỘI DUNG THỰC HÀNH

#### Vấn đề 1: Thư viện `<math.h>`: Hàm mũ (tính lũy thừa)

- 1.1. Hàm số mũ **double pow(double base, double exp)**; tính và trả về giá trị của biểu thức  $\text{base}^{\text{exp}}$ .

BIẾN	MÔ TẢ	LOẠI	GIÁ TRỊ HỢP LỆ	KIỂU
base	Cơ số hàm mũ	input	Số thực	double
exp	Lũy thừa	input	Số thực	double
pow	Tên hàm chứa giá trị trả về	output	Số thực	double

**Yêu cầu:** Viết chương trình nhập vào 2 số a và b, hiển thị lên màn hình giá trị của  $a^b$ .

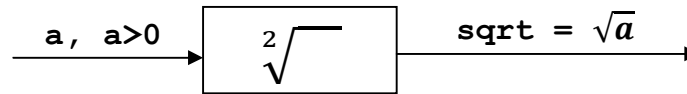


- 1.2. Kiểu nguyên 4 bytes lưu trữ các số nguyên trong giới hạn  $[-2^{31} .. 2^{31}-1]$ . Và kiểu nguyên không dấu 4 bytes lưu trữ các số nguyên trong giới hạn  $[0 .. 2^{32}-1]$ . Sử dụng hàm pow() để hiển thị lên màn hình giới hạn của kiểu **int** và **unsigned int**. Lưu vào thư mục **LAB08** với tên file **Range\_L8.CPP**.
- 1.3. Phát triển chương trình **Range\_L8.CPP** để điền dữ liệu cho bảng tính sau:

Tên kiểu	Kích thước	Giới hạn
char		
unsigned char		
int		
unsigned int		

#### Vấn đề 2: Thư viện `<math.h>`: Hàm lấy căn bậc 2

- 2.1. Hàm khai căn: **double sqrt(double num)**; nhận vào 1 số **num** không âm có kiểu **double** và trả về căn bậc 2 của **num** có kiểu **double**. Viết chương trình nhập vào số không âm a và hiển thị lên màn hình giá trị của  $\sqrt{a}$ .



2.2. Viết chương trình nhập tọa độ của 2 điểm  $A(x_A, y_A)$  và  $B(x_B, y_B)$ . Tính và hiển thị lên màn hình độ dài của đoạn thẳng AB. Lưu vào thư mục **LAB08** với tên file **AB.CPP**

Ghi chú: Độ dài véc tơ  $\overrightarrow{AB}$  được tính:  $|\overrightarrow{AB}| = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$

**Vấn đề 3: Tính tổng  $S = 0! + 1! + 2! + \dots + n!$  ( $n \geq 0$ )**

Định nghĩa:  $n! = \begin{cases} 1 & \text{nếu } n = 0 \\ 1 \times 2 \times \dots \times n & \text{nếu } n \geq 1 \end{cases}$

Yêu cầu:

1. Lập bảng dữ liệu vào/ra của bài toán

BIẾN	MÔ TẢ	LOẠI	GIÁ TRỊ HỢP LỆ	Kiểu

2. Viết các nguyên mẫu hàm cho 2 chức năng sau:

- Nhận vào từ bàn phím 1 số nguyên không âm
- Tính k!

3. Dựa vào mã giải dưới đây, viết định nghĩa các hàm input() và factorial()

<b>input</b>	Nhập số nguyên không âm k
<b>Analysis</b> A positive integer $\rightarrow$ int k	<b>Suggested algorithm</b> Begin do { Accept k; } while (n < 0); return k; End.
<b>factorial</b>	<b>Tính n!</b>
<b>Algorithm for</b> <b>Computing factorial</b> <b>of an integer</b>	<pre>double factorial( int k) { double p=1; int i; for (i=1; i&lt;=n; i++) p *= i; return p; }</pre>

4. Xây dựng test case và viết chương trình tính S

**Vấn đề 4:** Nhận vào từ bàn phím tọa độ của 3 điểm trong mặt phẳng Oxy. Tính và trả về diện tích của tam giác được tạo thành (trường hợp 3 điểm thẳng hàng, trả về giá trị 0)

Yêu cầu:

1. Viết các nguyên mẫu hàm sau:
  - Nhận vào tọa độ của 2 điểm A, B và trả về độ dài đoạn thẳng AB
  - Nhận vào độ dài 3 cạnh a, b, c của tam giác và trả về diện tích của nó.
2. Lập test case và viết chương trình thực hiện yêu cầu đặt ra

**Vấn đề 5:** Số tự nhiên  $N$  có  $k$  chữ số thì cấu tạo số của nó là  $a_{k-1}a_{k-2}...a_2a_1a_0$ , trong đó  $a_0$  là chữ số hàng đơn vị và mỗi chữ số  $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$  với  $i=0, \dots, k-1$

**Yêu cầu:**

- Viết nguyên mẫu hàm <kiểu trả về> sumDigits(<tham số>) nhận vào số nguyên không âm  $N$  và trả về tổng các chữ số của  $N$ .
- Định nghĩa hàm sumDigits() và lập bảng test\_case
- Dựa vào mã giả dưới đây, viết chương trình nhập số nguyên không âm  $N$  từ bàn phím và hiển thị tổng các chữ số của  $N$

<b>Problem</b>	<b>Viết chương trình tính tổng các số hạng của số nguyên <math>n</math> được nhập.</b>
<b>Analysis</b> Sum $\rightarrow$ int $S=0$ Accepted integer $\rightarrow$ int $n$	<b>Suggested algorithm ()</b> Begin Do { Accept $n$ ; If ( $n \geq 0$ ) { $S = \text{sumDigits}(n)$ ; Print out $S$ ; } } While ( $n \geq 0$ ); End
<b>Algorithm for Computing sum of digits of a 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 TẬP

### Bài 0047

Dựa vào mã giả dưới đây, hãy 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> Nouns: positive integer $\rightarrow$ int $n$	<b>Suggested algorithm (logical order of verbs)</b> Begin Do { Accept $n$ ; } While ( $n < 2$ ); For ( $i=2$ to $n$ ) If ( <b><math>i</math> is a prime</b> ) Print out $i$ ; End
---	--

Algorithm for checking whether an integer is a prime or not	<pre> <b>int</b>prime( <b>int</b> n ) {     <b>int</b> m = sqrt(n); /* m: square root of n */     <b>int</b> i; /* variable having value from 2 to m */     <b>if</b> (n&lt;2) <b>return</b> 0; /* Condition 1 is not satisfied */     <b>for</b> ( i=2; i&lt;=m; i++)/* checking the second condition */         <b>if</b> (n%i==0) <b>return</b> 0 ;         /* n is divided by i → n is not a prime */     <b>return</b> 1; /* n is a prime */ } </pre>
---	--

**Bài 0048****Yêu cầu:**

1. Đọc đoạn code dưới đây và chuyển thành chương trình
2. Lập bảng test\_case

<b>Related knowledge</b>	Năm nhuận (y): (y%400==0    ( y%4==0 && y%100!=0))
<b>Problem</b>	<i>Viết chương trình nhập vào ngày/tháng/năm, kiểm tra ngày tháng năm đã nhập có hợp lệ hay không.</i>
<b>Analysis</b> Data of a day <b>int</b> d, m, y	<b>Suggested algorithm (logical order of verbs)</b> Begin Accept d, m, y <b>if</b> (valid(d,m,y)) print out "valid date" <b>else</b> print out "invalid date" End
Algorithm for checking whether a date is valid or not	<pre> <b>Int</b> validDate ( <b>int</b> d, <b>int</b> m, <b>int</b> y) {     <b>int</b> maxd = 31;     /*max day of months 1, 3, 5, 7, 8, 10, 12 */     /* basic checking */     <b>if</b> ( d&lt;1    d&gt;31    m&lt;1    m&gt;12) <b>return</b> 0;     /* update maxd of a month */     <b>if</b> ( m==4    m==6    m==9    m==11) maxd=30;     <b>else if</b> (m==2)     { /* leap year? */         <b>if</b> ( y%400==0    ( y%4==0 &amp;&amp; y%100!=0)             maxd=29;         <b>else</b> maxd=28;     }     <b>return</b> d&lt;=maxd; } </pre>

**Bài 0049****Yêu cầu:**

3. Đọc đoạn code dưới đây và chuyển thành chương trình
4. Lập bảng test\_case

<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 0050****Yêu cầu:**

1. Đọc đoạn code dưới đây và chuyển thành chương trình
2. Lập bảng test\_case

<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> <b>int t1=1, t2=1, f=1, i ;</b> <b>for ( i= 3, i&lt;=n; i++)</b> <b>{</b> <b>f= t1 + t2;</b> <b>t1= t2;</b> <b>t2=f;</b> <b>}</b> <b>return f;</b> <b>}</b>

**Bài 0051****Yêu cầu:**

1. Đọc đoạn code dưới đây và chuyển thành chương trình
2. Lập bảng test\_case

<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:</p> <p><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; }</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																										

	<pre>int lcm ( int a, int b) {     return a*b/ gcd(a,b); }</pre>
<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 0052****Yêu cầu:**

1. Đọc đoạn code dưới đây và chuyển thành chương trình
2. Lập bảng test\_case

<b>Related knowledge</b>	<p><b>Print out the minimum and the maximum digits of a nonnegative integer integer</b></p> <p>Example: n= 10293 → Print out 9, 0</p> <pre>void printMinMaxDigits( int n) {     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>	Viết chương trình nhập số nguyên không âm, in ra chữ số nhỏ nhất và lớn nhất.
<b>Analysis</b> Noun: A integer → int n	<b>Suggested algorithm (logical order of verbs)</b> Begin Do {     Accept n; printMinMaxDigits (n) ; } While (n<0); End

**Bài 0053**

**Tạo menu công việc. Tên chương trình allLab8.CPP lưu trong LAB08**

Anh/chị hãy lập chương trình đưa ra bảng chọn sau:

**Chương trình**

1. Bai tap 47 :: Primer
2. Bai tap 48 :: validate
3. Bai tap 49 :: getRelPos
4. Bai tap 50 :: Fibonacci
5. Bai tap 51 :: gcd & lcm
6. Bai tap 52 :: MinMax
7. Thoat

Bạn hãy bấm 1 số(1, ..6) để chọn chức năng tương ứng hoặc bấm số 7 để kết thúc:

Sau khi người dùng chọn một số, chương trình gọi hàm thực hiện bài tập tương ứng

**English**

There are two important operations on a stack: PUSH and POP. PUSH adds the new data to the top of the stack leaving previous data below, and POP removes and returns the current top data of the stack. When the operations shown below are sequentially executed, which of the following is the correct combination of the values x and y?

Here, the size of the stack is big enough to hold the entire data. “PUSH(a)” inserts the data a into the stack, and “POP(b)” removes the data b from the stack.

**[Operations]**

PUSH (5);  
PUSH (3);  
PUSH (6);  
PUSH (1);  
x= POP ();  
PUSH (7);  
y= POP ();

	x	y
a)	1	6
b)	1	7
c)	5	3
d)	5	7