



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

C PROGRAMMING INTRODUCTION

TUẦN 9: HÀM

Hàm

- một nhóm các khai báo và câu lệnh được gán tên
 - thường có một giá trị
- một chương trình con
 - chương trình luôn có một hàm **main**
 - trong hàm **main** có thể **gọi** các hàm khác
 - các hàm này có thể gọi các hàm khác...

VD: Bình phương

```
double square(double a)
{
    return a * a;
}
```

Hàm được định nghĩa
bên ngoài hàm main

```
int main(void)
{
```

```
    double num = 0.0, sqr = 0.0;
```

```
    printf("enter a number\n");
    scanf("%lf", &num);
```

```
    sqr = square(num);
```



Lời gọi hàm

```
    printf("square of %g is %g\n", num, sqr);
```

```
    return 0;
```

Tại sao sử dụng hàm?

- Chia nhỏ vấn đề ra các tác vụ con
 - dễ dàng xử lý các vấn đề phức tạp
- Tổng quát hóa một tập các câu lệnh lặp đi lặp lại
 - không phải gõ lại cùng một đoạn code nhiều lần
 - Ví dụ: hàm **printf** và **scanf**
- Chương trình dễ đọc và dễ bảo trì

Tính chất của hàm

```
return-type name (argument-list)
{
    local-declarations
    statements
    return return-value;
}
```

- Khi gọi hàm, có thể kèm theo **tham số của hàm**
- Khai báo tham số trong nguyên mẫu hàm

Ví dụ 1

```
#include <stdio.h>
/* Ham in loi chao mung */

void sayHello ( void )
{
    printf("Hello World!\n");
}

/*
 * Goi ham in loi chao mung
 */

int main(void)
{
    sayHello();
    return 0;
}
```

Ví dụ 2a và 2b

```
#include <stdio.h>

int giaithua (int a) {
    int i, gt=1;
    for(i=1; i<=a; i++)
        gt = gt * i;
    return gt;
}

int main( void ) {
    int num;

    printf("Nhap so nguyen:");
    scanf ("%d", &num);

    printf ("%d!=%d\n",
            num, giaithua(num));
}
```

```
#include <stdio.h>

int giaithua (int);

int main( void ) {
    int num;

    printf("Nhap so nguyen:");
    scanf ("%d", &num);

    printf ("%d!=%d\n",
            num, giaithua(num));
}

int giaithua (int a) {
    int i, gt=1;
    for(i=1; i<=a; i++)
        gt = gt * i;
    return gt;
}
```

Ví dụ 3

```
#include <stdio.h>

int addOne ( int i )
{
    i = i + 1;
    return i;
}

int main(void)
{
    int i = 3;

    printf("%d\n", addOne(i) );
    printf("%d\n", i);

    return 0;
}
```


Ví dụ 4

```
void badSwap ( int a, int b )
{ int temp;
  temp = a;
  a = b;
  b = temp;
  printf("Called environment: %d %d\n",a,b) ;
}

int main(void)
{ int a = 3, b = 5;
  printf("Calling environment: %d %d\n",a,b) ;
  badSwap ( a, b ) ;
  printf("Calling environment: %d %d\n",a,b) ;
  return 0;
}
```

Ví dụ 5

```
int i;  
void f() {  
    int i;  
    i++; // chỉ làm thay đổi giá trị biến i cục bộ  
}  
void g() {  
    i++; // làm thay đổi giá trị của biến i tổng thể  
}  
int main() {  
    i = 10;  
    f();  
    printf("i = %d", i);  
    g();  
    print("i = %d", i);  
}
```

Bài tập 9.1

- Viết hàm tính động năng của một vật
 $ke = mv^2/2$, với m là khối lượng (kg) và v là tốc độ (m/s)
- Sử dụng hàm trong chương trình

Lời giải

```
#include <stdio.h>

double kineticEnergy(double m, double v) {
    return m*v*v/2;
}

void main() {
    double m, v;
    do {
        printf("Enter mass:"); scanf("%f",&m);
        printf("Enter speed:"); scanf("%f",&v);
    } while (m>0 && v >=0);
    printf("Kinetic Energy of element is:%f",
        kineticEnergy(m,v));
}
```

Bài tập 9.2

- Viết hàm `is_prime` nhận vào một số nguyên dương và trả về 1 nếu là số nguyên tố, 0 nếu ngược lại.

Nguyên mẫu:

```
int is_prime(int n);
```

- Viết chương trình cho người dùng nhập vào một số nguyên dương N và in ra màn hình tất cả các số nguyên tố từ 2 tới N sử dụng hàm trên

Lời giải

```
int is_prime(int n)
{
    int i = 0;

    /* Check if any of the numbers 2, ...,
       n-1 divide it. */
    for (i = 2; i < sqrt(n); ++i)
    {
        if (n % i == 0)
        {
            return 0;
        }
    }
    return 1;
    /* If we got here
       n is necessarily prime */
}
```

Lời giải

```
int main(void)
{
    int num = 0, i = 0;
    /* Get input from user */
    printf("enter a positive integer\n");
    scanf("%d", &num);

    printf("prime numbers up to %d:\n", num);
    for (i = 2; i <= num; ++i)
    {
        if (is_prime(i))
        {
            printf("%d\n", i);
        }
    }
    return 0;
}
```

Truyền tham số

- Tham số được truyền cho hàm bằng cách sao chép giá trị thay vì cung cấp truy cập đến biến
- Tham số thay đổi bên trong thân hàm sẽ không làm thay đổi giá trị của biến

Bài tập 9.3

- Viết chương trình có các hàm sau và sử dụng chúng trong hàm main
 - Hàm tính tổng lập phương của các số từ 1 tới n
 - Hàm liệt kê các ước số của một số nguyên dương N
 - Hàm liệt kê n bình phương của n số đầu tiên.

Lời giải

Hàm tính tổng lập phương của các số từ 1 tới n

```
long sumcube(int n)
{
    int i = 0;
    long s=0;
    for(i=1; i<=n; i++)
        s += i*i*i;
    return s;
}
```

Lời giải

Hàm liệt kê các ước số của một số nguyên dương N

```
void printsubmultiples(int n)
{
    int i;
    for(i=2; i<n; i++)
        if (n%i ==0) printf("%d ",i) ;
    printf("\n") ;
}
```

Lời giải

```
void printsquares(int n)
{
    int i;
    for(i=1; i<=n; i++)
        printf("%d ", i*i);
    printf("\n");
}
```

Bài tập 9.4

- Viết chương trình tính lương công nhân trong tuần.
- Lương trung bình là 15000 VND một giờ.
- Công nhân phải làm 40 giờ một tuần.
- Nếu làm thêm giờ, công nhân được trả gấp 1.5 số tiền trong giờ.
- Công nhân không được làm dưới 10 giờ hoặc hơn 65 giờ một tuần

Lời giải

```
#include <stdio.h>
long salary(int hours)
{
    if (hours > 40)
        return 15000*40 + 15000(hours-40)*3/2;
    else
        return hours*40;
}
```

Lời giải

```
int main()
{
    int n;
    do {
        printf("Enter number of working hours:");

        scanf("%d", &n);
    } while (m<10 || n>=65);

    printf("The salary you get:%ld\n",
           salary(n));
    return 0;
}
```

Bài tập 9.5

- Viết hàm

`void printnchars(int ch, int n)` để hiển thị một kí tự `n` lần.

Sử dụng hàm này để in ra tam giác * như sau

*

**

Lời giải

```
void printnchars(int ch, int n)
{
    int i;
    for(i = 0; i < n; i++)
        printf("%c", ch);
}
```

Bài tập 9.6

- Cho công thức chuyển từ độ F sang độ C:
$$C = 5/9 (F-32)$$
- In ra bảng liệt kê độ F từ 1 tới 20 và độ C tương ứng

```
// function to convert fahrenheit to celsius  
double celsius(double);
```

Lời giải

```
int main() {  
    double fahr = 0;  
  
    printf("Fahrenheit\tCelsius\n");  
    while (fahr < 21) {  
        printf("%6.1f\t%6.1f\n", fahr, celsius(fahr));  
        fahr += 1;  
    }  
  
    return 0;  
}  
  
double celsius(double f) {  
    return 5 * (f - 32) / 9;  
}
```

Bài tập 9.7

- Cho số nguyên dương N có k chữ số. Viết chương trình kiểm tra xem tất cả k chữ số đều là số chẵn hoặc đều là số lẻ.

Lời giải

```
#include <stdio.h>
int DigitAllSame(int n){
    int digit;
    int count =0;
    int flagEven,flagOdd;
    flagEven=1; flagOdd=1;
    while (n>0 && count<5){
        digit = n%10;
        n= n/10;
        count++;
        if (digit%2 == 0) {
            flagEven= flagEven*1;
            flagOdd= flagOdd*0;
        }
        else {
            flagEven= flagEven*0;
            flagOdd= flagOdd*1;
        }
    }
    printf("count = %d\n", count);
    if (count>=5) return -1;
    if (flagEven || flagOdd) return 1;
    else return 0;
}
```

```
main() {
    printf("Hello.\n");
    printf("So %d co gia tri ham la
%d\n", 44668, DigitAllSame(46668));
}
```

Bài tập 9.8

- Chương trình Vietnam Idol có 5 giám khảo. Mỗi giám khảo cho điểm thí sinh từ 0 tới 10. Điểm cuối cùng là điểm trung bình của 3 điểm sau khi loại đi điểm cao nhất và thấp nhất.
- Viết chương trình tính điểm của thí sinh sử dụng các hàm sau:
 - void getJudgeData() nhận điểm của một giám khảo thông qua con trỏ tới biến và kiểm tra tính đúng đắn
 - void calcScore() tính và hiển thị điểm
 - double findLowest() tính điểm thấp nhất
 - double findHighest() tính điểm cao nhất

```
#include <stdio.h>
#include <stdlib.h>
```

Lời giải

```
// function to get Judge's score
void getJudgeData(double *);
// function to calculate competitor's score
void calcScore(double, double, double, double, double);
double findLowest(double, double, double, double, double);
double findHighest(double, double, double, double, double);

int main() {
    double s1, s2, s3, s4, s5;

    getJudgeData(&s1);
    getJudgeData(&s2);
    getJudgeData(&s3);
    getJudgeData(&s4);
    getJudgeData(&s5);
    calcScore(s1, s2, s3, s4, s5);
    return 0;
}
```

Lời giải

```
void getJudgeData(double *s) {  
    do {  
        printf("Enter a judge's score: "); scanf("%f",s);  
    } while (*s < 0 || *s > 10);  
}
```

```
double findLowest(double s1, double s2, double s3, double s4, double s5)  
{  
    double min = s1;  
    if (s2 < min) min = s2;  
    if (s3 < min) min = s3;  
    if (s4 < min) min = s4;  
    if (s5 < min) min = s5;  
    return min;  
}
```


Lời giải

```
double findHighest(double s1, double s2, double s3, double  
    s4, double s5) {  
    double max = s1;  
    if (s2 > max) max = s2;  
    if (s3 > max) max = s3;  
    if (s4 > max) max = s4;  
    if (s5 > max) max = s5;  
    return max;  
}
```

Lời giải

```
void calcScore(double s1, double s2, double s3, double s4,  
double s5) {  
    double sum = s1 + s2 + s3 + s4 + s5;  
    double max = findHighest(s1, s2, s3, s4, s5);  
    double min = findLowest(s1, s2, s3, s4, s5);  
    sum -= (max + min);  
    printf("Max = %1.2f\n", max);  
    printf("Min = %1.2f\n", min);  
    printf("Final score: %1.2f\n", sum / 3);  
}
```

Bài tập 9.9

- Viết hàm *isLeapYear* nhận vào năm là tham số và kiểm tra có phải năm nhuận
- Năm nhuận:
 - chia hết cho 4 nhưng không chia hết cho 100 OR
 - chia hết cho 400
 - VD: 1996 và 2000 là năm nhuận, nhưng 1900, 2002 và 2100 không phải năm nhuận

Bài tập 9.10

- Viết hàm xác định số nguyên là số chẵn hay số lẻ.
 - Tham số: số nguyên
 - Giá trị trả về của hàm: 0 (số chẵn) hoặc 1 (số lẻ)
- Nhập một số nguyên N từ bàn phím, sử dụng hàm trên để kiểm tra tính chẵn / lẻ của N.

Bài tập 9.11

- Viết hàm tính $n!$
- Sử dụng hàm này:
 - Nhập giá trị N từ bàn phím
 - Tính tổng $S = 1! + 2! + \dots + N!$ sử dụng hàm trên.

Bài tập 9.12

- Viết hàm chuyển đổi một số từ hệ cơ số 10 sang hệ cơ số 2.
 - Tham số: một số ở hệ cơ số 10, ví dụ: 47
 - Kết quả của hàm: một số ở hệ cơ số 10 có các chữ số mô tả số trên ở hệ cơ số 2. Ví dụ: $47 \Rightarrow 101111$
- Sử dụng hàm này:
 - Nhập một số N từ bàn phím.
 - Chuyển số này sang số M (dạng biểu diễn ở hệ cơ số 2)

Bài tập 9.13

- Viết hàm giải phương trình bậc 1: $a.x + b = 0$
 - Tham số: a, b
 - Giá trị trả về bởi hàm: x
- Sử dụng hàm này.

Bài tập 9.14

- Viết hàm xác định số Fibonacci thứ N :
 - Tham số: số nguyên N
 - Giá trị trả về của hàm: số Fibonacci thứ N
- Nhập một số nguyên N từ bàn phím
- Hiển thị số Fibonacci thứ N

Bài tập 9.15

- Viết hàm xác định hàm e^x bằng cách dùng vòng lặp while hoặc do...while sử dụng khai triển Taylor với độ chính xác $1e-M$
 - Tham số: số thực x , số nguyên dương M
 - Giá trị trả về của hàm: e mũ x
- Sử dụng hàm này để tính số E với độ chính xác M (M được nhập từ bàn phím)

f có đạo hàm cấp $n+1$ trong (a, b) chứa x_0 :

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \dots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n + R_n$$

$$R_n = \frac{f^{(n+1)}(c)}{(n+1)!}(x-x_0)^{n+1}, \quad c \text{ nằm giữa } x \text{ và } x_0$$

(khai triển Taylor đến cấp n trong lân cận x_0)

Bài tập 9.16

- Viết hàm xác định hàm $\arcsin(x)$ bằng cách dùng vòng lặp while hoặc do...while sử dụng khai triển Taylor với độ chính xác $1e-M$
 - Tham số: số thực x , số nguyên dương M
 - Giá trị trả về của hàm: e mũ x
- Sử dụng hàm này để tính số PI với độ chính xác M (M được nhập từ bàn phím)



25 YEARS ANNIVERSARY
SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you
for your
attentions!**



soict.hust.edu.vn/



fb.com/groups/soict

