

[Dashboard](#) / [My courses](#) / [SFP:HK2-2019-2020](#) / [Hàm](#) / [Hàm](#)

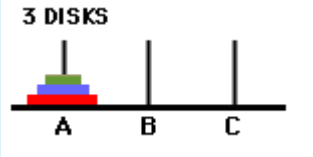
Started on	Wednesday, 13 May 2020, 3:32 PM
State	Finished
Completed on	Friday, 15 May 2020, 9:28 AM
Time taken	1 day 17 hours
Marks	6.00/8.00
Grade	7.50 out of 10.00 (75%)

Question **1**

Correct

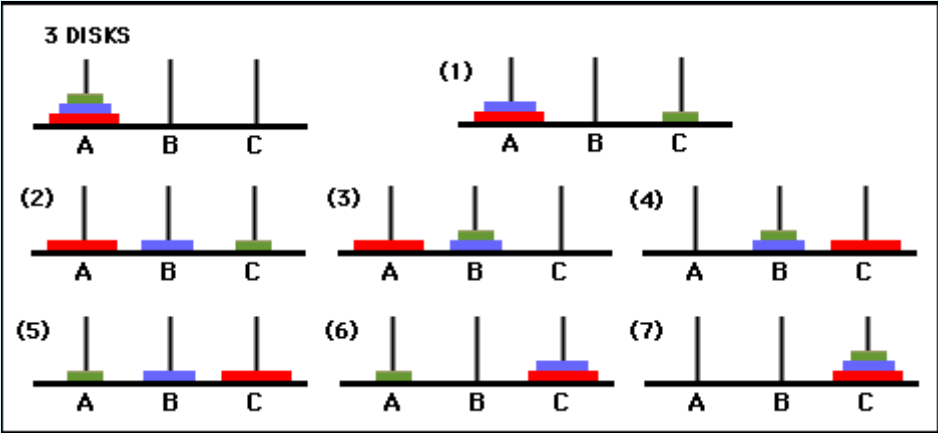
Mark 1.00 out of 1.00

Có 3 cây cọc và một chồng n đĩa có kích thước từ nhỏ đến lớn (không trùng kích thước). Các đĩa đều có lỗ bên trong để chồng vào các cây cọc. Ban đầu n đĩa được chồng vào cây cọc thứ nhất, đĩa lớn nhất ở dưới, và kích thước đĩa giảm dần từ dưới lên:



Nhiệm vụ của chúng ta là phải di chuyển n cái đĩa từ cọc A sang cọc C với ràng buộc là: Có một cái cọc trung gian là B, mỗi lần chỉ được di chuyển 1 cái đĩa từ một cọc đến cọc khác và đĩa to hơn không được xếp lên trên đĩa nhỏ hơn.

Ví dụ dưới đây cho thấy cách di chuyển 3 cái đĩa từ A sang C:



Viết 1 hàm đệ quy để mô tả các bước chuyển n đĩa từ cọc A sang cọc C với B là cọc trung gian.

Khuôn dạng (Prototype)

- Tên hàm (function name): **Move**
- Tham số (parameters):
 - n** - kiểu int
 - A, B, C** - kiểu char
- Kiểu trả về (return type): **void**

Thân hàm (Body)

- Dùng đệ quy để mô tả các bước chuyển đĩa.

Chú ý

- Giá trị của các tham số luôn hợp lệ, không cần kiểm tra.
- Chỉ viết hàm, KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH
- Xem thêm chi tiết trong phần **For example**.

Gợi ý

- Nếu (số đĩa chuyển là 1 đĩa) --> Thực hiện chuyển từ A sang C
- Ngược lại, xu hướng giải bài toán sẽ là
 - Chuyển n-1 đĩa từ A sang B, lúc đó C là trung gian
 - Chuyển 1 đĩa từ A sang C, lúc đó B là trung gian
 - Chuyển n-1 đĩa từ B sang C, lúc đó A là trung gian

For example:

Test	Result
Move(3, 'A', 'B', 'C')	A -> C A -> B C -> B A -> C B -> A B -> C A -> C
Move(1, 'A', 'B', 'C')	A -> C

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 void Move(int n, char A, char B, char C){
2     if(n==1){
3         printf("%c -> %c\n",A,C);
4     }
5     else {
6         Move(n-1, A, C, B);
7         Move(1, A, B, C);
8         Move(n-1, B, A, C);
9     }
10 }
```

	Test	Expected	Got	
✓	Move(3, 'A', 'B', 'C')	A -> C A -> B C -> B A -> C B -> A B -> C A -> C	A -> C A -> B C -> B A -> C B -> A B -> C A -> C	✓
✓	Move(1, 'A', 'B', 'C')	A -> C	A -> C	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

Write a recursive function to find the Greatest Common Divisor (GCD) of two numbers a and b.

Prototype

- function name: GCD
- parameters: two integers a and b ($0 \leq |a|, |b| \leq 10^6$)
- return type: int

Body

- compute the greatest common divisor of a and b.

Notes:

- Input is always valid.
- The greatest common divisor is no negative.

See examples for more details in input/output.

Hint

Using recursive definition of GCD

- $GCD(a, b) = a$, if $b = 0$
- $GCD(a, b) = GCD(b, a \% b)$, if $b \neq 0$

For example:

Test	Result
printf("%d", GCD(0, 4));	4
printf("%d", GCD(1, -5));	1
printf("%d", GCD(6, 9));	3
printf("%d", GCD(-12, 8));	4
printf("%d", GCD(36, 33));	3
printf("%d", GCD(-1000, -2000));	1000

Answer: (penalty regime: 33.3, 66.7, ... %)

1

2

3

4

5

6

7

8

```
int GCD(int a, int b){
    a=abs(a);
    b=abs(b);
    if(b==0){
        return a;
    }
    return GCD(b,a%b);
}
```

	Test	Expected	Got	
✓	printf("%d", GCD(0, 4));	4	4	✓
✓	printf("%d", GCD(1, -5));	1	1	✓
✓	printf("%d", GCD(6, 9));	3	3	✓
✓	printf("%d", GCD(-12, 8));	4	4	✓
✓	printf("%d", GCD(36, 33));	3	3	✓

	Test	Expected	Got	
✓	printf("%d", GCD(-1000, -2000));	1000	1000	✓

Passed all tests! ✓

Question author's solution:

```
int GCD(int a, int b) {
    if (b == 0)
        return a > 0 ? a : -a;
    return GCD(b, a%b);
}
```

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 0.00 out of 1.00

Viết hàm kiểm tra số n có phải là số hoàn thiện không.

Khuôn dạng (Prototype)

- Tên hàm (function name): **isPerfect**
- Tham số (parameters):
 - n** - số cần kiểm tra, kiểu: int ($2 \leq n \leq 10^6$)
- Kiểu trả về (return type): **int**

Thân hàm (Body)

- Nếu n là số hoàn thiện trả về 1
- Ngược lại trả về 0.

Chú ý

- Giá trị của các tham số luôn hợp lệ, không cần kiểm tra.
- Chỉ viết hàm, KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH
- Xem thêm chi tiết trong phần **For example**.

Gợi ý

- Xem lại bài tập kiểm tra số hoàn thiện trong phần vòng lặp.

For example:

Test	Result
if (isPerfect(6)) printf("YES"); else printf("NO");	YES
if (isPerfect(28)) printf("YES"); else printf("NO");	YES
if (isPerfect(1000)) printf("YES"); else printf("NO");	NO

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 int isPerfect(int n){
2     int sum = 0; //khai bao biem sum
3     for(int i=1;i<=n/2;i++){
4         if(n%i==0)
5             sum+=i;
6     }
7     if(sum==n) return 1;
8     return 0;
9 }
```

	Test	Expected	Got	
✓	if (isPerfect(6)) printf("YES"); else printf("NO");	YES	YES	✓

	Test	Expected	Got	
✓	if (isPerfect(24)) printf("YES"); else printf("NO");	NO	NO	✓
✓	if (isPerfect(28)) printf("YES"); else printf("NO");	YES	YES	✓
✓	if (isPerfect(496)) printf("YES"); else printf("NO");	YES	YES	✓
✓	if (isPerfect(8128)) printf("YES"); else printf("NO");	YES	YES	✓
✓	if (isPerfect(1000)) printf("YES"); else printf("NO");	NO	NO	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.00/1.00**.

Question **4**

Correct

Mark 0.33 out of 1.00

Viết hàm tính $n!!$ ($0 \leq n \leq 25$) theo công thức sau:

$$n!! = \begin{cases} 1 * 3 * 5 * \dots * n, & n \text{ lẻ} \\ 2 * 4 * 6 * \dots * n, & n \text{ chẵn} \end{cases}$$

Khuôn dạng (Prototype)

- Tên hàm (function name): **Factorial2**
- Tham số (parameters):
 - n** - kiểu int
- Kiểu trả về (return type): **long long**

Thân hàm (Body)

- Áp dụng công thức trên để tính giá trị của $n!!$

Chú ý

- Giá trị của các tham số luôn hợp lệ, không cần kiểm tra.
- Chỉ viết hàm, KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH
- Xem thêm chi tiết trong phần **For example**.

For example:

Test	Result
<pre>int i; for(i=1;i<=25;i++) printf("%lld\n",Factorial2(i));</pre>	1
	2
	3
	8
	15
	48
	105
	384
	945
	3840
	10395
	46080
	135135
	645120
	2027025
	10321920
	34459425
	185794560
	654729075
	3715891200
<pre>printf("%lld",Factorial2(24));</pre>	13749310575
	81749606400
	316234143225
	1961990553600
	7905853580625
	1961990553600

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 long long Factorial2(int n){
2     if(n==1 || n==2){
3         return n;
4     }
5     return n*Factorial2(n-2);
6 }
7 }
```


	Test	Expected	Got	
✓	<pre>int i; for(i=1;i<=25;i++) printf("%lld\n",Factorial2(i));</pre>	1 2 3 8 15 48 105 384 945 3840 10395 46080 135135 645120 2027025 10321920 34459425 185794560 654729075 3715891200 13749310575 81749606400 316234143225 1961990553600 7905853580625	1 2 3 8 15 48 105 384 945 3840 10395 46080 135135 645120 2027025 10321920 34459425 185794560 654729075 3715891200 13749310575 81749606400 316234143225 1961990553600 7905853580625	✓
✓	<pre>printf("%lld",Factorial2(24));</pre>	1961990553600	1961990553600	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.33/1.00**.

Question **5**

Correct

Mark 1.00 out of 1.00

Write a recursive function that converts a decimal number into a binary number and prints the binary number.

Prototype

- function name: **dec2bin**
- parameters:
 - an integer n - the number (1 <= n <= 64)
- return type: **none**

Body

- recursively print the binary bits of n.

Notes:

- Input is always valid.

See examples for more details in input/output.

Hint

- Using recursive definition of **dec2bin(n)**:
- if n = 0, do nothing
 - else **recursively call dec2bin(n/2)** and then **print n%2**.

For example:

Test	Result
dec2bin(3);	11
dec2bin(1);	1
dec2bin(14);	1110
dec2bin(64);	1000000

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 void dec2bin(int n){
2     if(n==0){
3     }
4     else{
5         dec2bin(n/2);
6         printf("%d",n%2);
7     }
8 }
```

	Test	Expected	Got	
✓	dec2bin(3);	11	11	✓
✓	dec2bin(1);	1	1	✓
✓	dec2bin(14);	1110	1110	✓
✓	dec2bin(64);	1000000	1000000	✓

Passed all tests! ✓

Question author's solution:

```
void dec2bin(int n) {  
    if (n == 0)  
        return;  
    dec2bin(n/2);  
    printf("%d", n%2);  
}
```

Correct

Marks for this submission: 1.00/1.00.

Question **6**

Correct

Mark 1.00 out of 1.00

Viết hàm **đệ quy** để in các chữ số của số nguyên theo thứ tự từ phải sang trái.

Khuôn dạng (Prototype)

- Tên hàm (function name): **in_nguoc**
- Tham số (parameters):
 - **n** - Số cần in, kiểu int
- Kiểu trả về (return type): **void**

Thân hàm (Body)

- In các chữ số của n theo thứ tự từ phải sang trái, mỗi chữ số cách nhau 1 khoảng trắng.

Chú ý

- Giá trị của các tham số luôn hợp lệ, không cần kiểm tra.
- Chỉ viết hàm, KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH
- Phải sử dụng đệ quy, không được dùng vòng lặp for, while, ...
- Xem thêm chi tiết trong phần **For example**.

Gợi ý

- Nếu n = 0, không in gì cả
- Ngược lại
 - In số cuối của n
 - Gọi đệ quy in các chữ số của n/10

For example:

Test	Result
in_nguoc(1346);	6 4 3 1
in_nguoc(401);	1 0 4

Answer: (penalty regime: 33.3, 66.7, ... %)

1

2

3

4

5

6

7

8

```
void in_nguoc(int n){
    if(n==0){
    }
    else{
        printf("%d ",n%10);
        in_nguoc(n/10);
    }
}
```

	Test	Expected	Got	
✓	in_nguoc(1346);	6 4 3 1	6 4 3 1	✓
✓	in_nguoc(100000);	0 0 0 0 0 1	0 0 0 0 0 1	✓
✓	in_nguoc(401);	1 0 4	1 0 4	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **7**

Correct

Mark 1.00 out of 1.00

Viết hàm **đệ quy** tính lũy thừa bậc n của a:

$a^n = a * a * \dots * a$, tích của n chữ a.

Ví dụ:

- $2^3 = 8$
- $3^4 = 81$
- $2^{10} = 1024$

Khuôn dạng (Prototype)

- Tên hàm (function name): **power**
- Tham số (parameters):
 - **a** - cơ số, kiểu: float ($0 \leq |a| \leq 5$)
 - **n** - số mũ, kiểu: int ($0 \leq n \leq 10$)
- Kiểu trả về (return type): **double**

Thân hàm (Body)

- Tính và trả về a^n .

Chú ý

- Giá trị của các tham số luôn hợp lệ, không cần kiểm tra.
- Chỉ viết hàm, KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH
- Không được sử dụng vòng lặp để tính
- Xem thêm chi tiết trong phần **For example**.

Gợi ý

- Tìm công thức đệ quy cho a^n
 - $a^0 = 1$
 - $a^n = a^{n-1} * a$

For example:

Test	Result
printf("%.4f\n", power(2, 3));	8.0000
printf("%.4f\n", power(3.14, 10));	93174.4045
printf("%.4f\n", power(-2.0, 5));	-32.0000
printf("%.4f\n", power(-3.123, 0));	1.0000

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 double power(float a, int n){
2     if(n==0){
3         return 1;
4     }
5     if(n==1){
6         return a;
7     }
8     return a*power(a,n-1);
9 }
```

Test	Expected	Got	
------	----------	-----	--

	Test	Expected	Got	
✓	printf("%.4f\n", power(2, 3));	8.0000	8.0000	✓
✓	printf("%.4f\n", power(3.14, 10));	93174.4045	93174.4045	✓
✓	printf("%.4f\n", power(5.0, 10));	9765625.0000	9765625.0000	✓
✓	printf("%.4f\n", power(-2.0, 5));	-32.0000	-32.0000	✓
✓	printf("%.4f\n", power(-3.123, 0));	1.0000	1.0000	✓
✓	printf("%.4f\n", power(-0.5, 10));	0.0010	0.0010	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **8**

Correct

Mark 0.67 out of 1.00

You are given a function

```
int max2(int a, int b);
```

which returns the bigger between **a** and **b**.

Write a C function that takes 3 integer numbers and returns the biggest using the function **max2**.

Prototype

- function name: **max3**
- parameters: three integers **a**, **b** and **c**.
- return type: **integer**

Body

- return the biggest among **a**, **b** and **c**.

Notes:

- Input is always valid.

See examples for more details in input/output.

For example:

Test	Result
printf("%d\n", max3(7, 9, 10));	10
printf("%d\n", max3(10, 9, 7));	10
printf("%d\n", max3(5, 5, 2));	5
printf("%d\n", max3(5, 12, 5));	12

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 int max3(int a, int b, int c){
2     int max=a;
3     if(max<b){
4         max=b;
5     }
6     if(max<c){
7         max=c;
8     }
9     return max;
10 }
```

	Test	Expected	Got	
✓	printf("%d\n", max3(7, 9, 10));	10	10	✓
✓	printf("%d\n", max3(10, 9, 7));	10	10	✓
✓	printf("%d\n", max3(5, 5, 2));	5	5	✓
✓	printf("%d\n", max3(5, 12, 5));	12	12	✓

Passed all tests! ✓

Question author's solution:

```
int max3(int a, int b, int c) {
    return max2(max2(a, b), c);
}
```

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.67/1.00**.

[◀ Một số bài tập vòng lặp nâng cao](#)

Jump to...

▾

[Bài tập về mảng ▶](#)