Đã bắt đầu vào lúc	Thứ hai, 3 Tháng tư 2023, 2:31 PM
Tình trạng	Đã hoàn thành
Hoàn thành vào lúc	Thứ hai, 3 Tháng tư 2023, 4:35 PM
Thời gian thực hiện	2 giờ 4 phút
Điểm	9,00/9,00
Điểm	**10,00** của 10,00 (**100**%)

Câu hỏi 1

Chính xác

Điểm 1,00 của 1,00

Write function **getPower(int x, int y)** calculate x^y (x and y are 2 non-negative integers and x*y!=0).

For example:

Test	Input	Result
`int x = 2, y = 3;` `printf("%d\n", getPower(x, y));`	2 3	8

Answer: (penalty regime: 0 %)

Reset answer

```
int getPower(int x, int y) {
    //TODO
    int result=1;
    for(int i=1;i<=y;i++){
        result=result*x;
    }
    return result;
}
```

	Test	Input	Expected	Got	
✔	`int x = 2, y = 3;` `printf("%d\n", getPower(x, y));`	2 3	8	8	✔

✔	`int x = 0, y = 0;`
	`scanf("%d%d", &x, &y);`
	`printf("%d\n", getPower(x, y));`
		4 0	1	1	✔

✔	`int x = 0, y = 0;`
	`scanf("%d%d", &x, &y);`
	`printf("%d\n", getPower(x, y));`
		0 100	0	0	✔

✔	`int x = 0, y = 0;`
	`scanf("%d%d", &x, &y);`
	`printf("%d\n", getPower(x, y));`
		2 10	1024	1024	✔

✔	`int x = 0, y = 0;`
	`scanf("%d%d", &x, &y);`
	`printf("%d\n", getPower(x, y));`
		3 6	729	729	✔

Passed all tests! ✔

Question author's solution (C):

```
int getPower(int x, int y) {
    int i = 0;
    int n = 1;
    for (i = 0; i < y; i++) {
        n = n * x;
    }
    return n;
}
```

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 2

Chính xác

Điểm 1,00 của 1,00

Implement function **long callMul(int n)** with a positive integer n input, calculate S = 1^1 + 2^2 + 3^3 + 4^4 +...+ n^n and return the result

For example:

Test	Result
printf("%ld", callMul(3));	32

Answer: (penalty regime: 0 %)

Reset answer

```
// You can implement some supportive functions here (optional)
int getPower(int x, int y) {
    int result = 1;
    for (int i = 1; i <= y; i++) {
        result *= x;
    }
    return result;
}

long callMul(int n) {
    //TO DO:
    int sum=0;
    for(int i=1;i<=n;i++){
        sum += getPower(i,i);
    }
    return sum;
}
```

	Test	Expected	Got	
✔	printf("%ld", callMul(3));	32	32	✔
✔	printf("%lu", sizeof(callMul(1)));	8	8	✔
✔	printf("%ld", callMul(1));	1	1	✔
✔	printf("%ld", callMul(5));	3413	3413	✔
✔	printf("%ld", callMul(9));	405071317	405071317	✔

Passed all tests! ✔

Question author's solution (C):

```
int getPower(int x, int y) {
    int i = 0;
    int n = 1;
    for (i = 0; i < y; i++) {
        n =  n * x;
    }
    return n;
}

long callMul(int n) {
    //TO DO:
    int i = 0;
    long sum = 0l;
    for (i = 0; i < n; i++) {
        sum = sum + getPower(i + 1, i + 1);
    }
    return sum;
}
```

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 3

Chính xác

Điểm 1,00 của 1,00

Implement function **int findMax(int N)** with a positive integer N input, Calculate and return the largest integer m so that 1 + 2 + ... + m < N

For example:

Test	Result
int N = 2; printf("%d", findMax(N));	1

Answer: (penalty regime: 0 %)

Reset answer

```
int findMax(int N) {
    //TO DO:
    int sum=0,i=0;
    while(sum<N){
        i++;
        sum += i;
    }
    return i-1;

}
```

	Test	Expected	Got	
✔	int N = 2; printf("%d", findMax(N));	1	1	✔
✔	int N = 4; printf("%d", findMax(N));	2	2	✔
✔	int N = 100; printf("%d", findMax(N));	13	13	✔
✔	int N = 1024; printf("%d", findMax(N));	44	44	✔
✔	int N = 66; printf("%d", findMax(N));	10	10	✔

Passed all tests! ✔

Question author's solution (C):

```
int findMax(int N) {
    int sum = 0;
    int temp = 0;
    while (sum < N) {
        temp = temp + 1;
        sum = sum + temp;
    }
    return temp - 1;
}
```

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 4

Chính xác

Điểm 1,00 của 1,00

Implement function **int GCD(int a, int b)** to find the greatest common divisor of a and b.

Implement function **int LCM(int a, int b)** to find the least common multiple of a and b.

For example:

Test	Result
int a = 3, b = 6; printf("%d %d", GCD(a, b), LCM(a, b));	3 6

Answer: (penalty regime: 0 %)

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

int LCM(int a, int b) {
    return (a * b) / GCD(a, b);
}
```

	Test	Expected	Got	
✔	int a = 3, b = 6; printf("%d %d", GCD(a, b), LCM(a, b));	3 6	3 6	✔
✔	int a = 10, b = 8; printf("%d %d", GCD(a, b), LCM(a, b));	2 40	2 40	✔
✔	int a = 5, b = 7; printf("%d %d", GCD(a, b), LCM(a, b));	1 35	1 35	✔
✔	int a = 63, b = 981; printf("%d %d", GCD(a, b), LCM(a, b));	9 6867	9 6867	✔
✔	int a = 44, b = 44; printf("%d %d", GCD(a, b), LCM(a, b));	44 44	44 44	✔

Passed all tests! ✔

Question author's solution (C):

```
int GCD(int a, int b) {
    for (int i = a; i > 1; i--) {
        if (a % i == 0 && b % i == 0) return i;
    }
    return 1;
}

int LCM(int a, int b) {
    for (int i = a; i < a * b; i++) {
        if (i % a == 0 && i % b == 0) return i;
    }
    return a * b;
}
```

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 5

Chính xác

Điểm 1,00 của 1,00

Implement function **void printSquare(int n)** to print all the square number from 1 to n.

For example:

Test	Result
printSquare(5);	1 4

Answer: (penalty regime: 0 %)

```
int getPower(int x, int y) {
    int result = 1;
    for (int i = 1; i <= y; i++) {
        result *= x;
    }
    return result;
}

void printSquare(int n){
    for(int i=1;i<=n;i++){
        if(getPower(i,2)<=n){
            printf("%d ",getPower(i,2));
        }
    }
}
```

	Test	Expected	Got	
✔	printSquare(5);	1 4	1 4	✔
✔	printSquare(10);	1 4 9	1 4 9	✔
✔	printSquare(0);			✔
✔	printSquare(100);	1 4 9 16 25 36 49 64 81 100	1 4 9 16 25 36 49 64 81 100	✔
✔	printSquare(500);	1 4 9 16 25 36 49 64 81 100 121 144 169 196 225 256 289 324 361 400 441 484	1 4 9 16 25 36 49 64 81 100 121 144 169 196 225 256 289 324 361 400 441 484	✔

Passed all tests! ✔

Question author's solution (C):

```
int isSquare(int n) {
    for (int i = 1; i <= n / 2 + 1; i++) {
        if (i * i == n) return 1;
    }
    return 0;
}

void printSquare(int n) {
    if (n > 0) printf("%d", 1);
    for (int i = 1; i < n; i++) {
        if (isSquare(i + 1)) printf(" %d", i + 1);
    }
}
```

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 6

Chính xác

Điểm 1,00 của 1,00

Complete the function **int calDays(int day, int month, int year)** with input day, month and year. Calculate what day of year is that and return it.

If the input is not a valid date return -1.

Hint: days in a month, leap years.

For example:

Test	Result
printf("%d", callDays(10, 1, 2021));	10

Answer: (penalty regime: 0 %)

```
int leap_year_or_not (int year) {
    if (year % 4 != 0) {
        return 0;
    } else if (year % 100 != 0) {
        return 1;
    } else if (year % 400 != 0) {
        return 0;
    } else {
        return 1;
    }
}

int callDays(int day, int month, int year){
    //days in month
    int daysInAMonth[] = {31, 28, 31, 30, 31, 30, 31, 31, 30
    if(leap_year_or_not(year)==1){
        daysInAMonth[1]=29;//leap year
    }
    //invalid
    if (month < 1 || month > 12 || day < 1 || day > daysInAM
```

	Test	Expected	Got	
✔	printf("%d", callDays(10, 1, 2021));	10	10	✔
✔	printf("%d", callDays(31, 12, 2021));	365	365	✔
✔	printf("%d", callDays(31, 12, 2020));	366	366	✔
✔	printf("%d", callDays(31, 12, 1900));	365	365	✔
✔	printf("%d", callDays(30, 9, 2021));	273	273	✔
✔	printf("%d", callDays(22, 1, 2000));	22	22	✔
✔	printf("%d", callDays(11, 10, 2021));	284	284	✔
✔	printf("%d", callDays(29, 2, 2016));	60	60	✔
✔	printf("%d", callDays(32, 7, 2016));	-1	-1	✔
✔	printf("%d", callDays(29, 2, 2021));	-1	-1	✔

Passed all tests! ✔

Question author's solution (C):

```
int isLeapYear(int year) {
    return (year % 4 == 0 && year % 100 != 0) || (year % 400
}

int callDays(int day, int month, int year) {
```

```
    if ((month == 4 || month == 6 || month == 9 || month ==
    else if (month == 2) {
        if (isLeapYear(year) && day > 29) return -1;
        else if (!isLeapYear(year) && day > 28)  return -1;
    }
    else if (day > 31 || day < 0 || month > 12 || month < 0

    for (int i = 1; i < month; i++) {
        if (i == 1 || i == 3 || i == 5 || i == 7 || i == 8 |
        else if (i == 2) {
            if (isLeapYear(year)) retVal += 29;
            else retVal += 28;
        }
        else retVal += 30;
```

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 7

Chính xác

Điểm 1,00 của 1,00

Write a function to print “*” with the given n

For example:

Test	Result

`print_star(3);`	`*`
	`* *`
	`* * *`
	`* *`
	`*`

Answer: (penalty regime: 0 %)

Reset answer

```
void print_star(int n)
{
    //TODO
    for(int i=1;i<=2*n-1;i++){
        if(i<=n){
            for (int j=1;j<=i;j++){
                printf("* ");
            }
            printf("\n");
        } else {
            for (int j=1;j<=n-i%n;j++){
                printf("* ");
            }
            printf("\n");
        }
    }
}
```

	Test	Expected	Got	

✔	`print_star(3);`	`*`
		`* *`
		`* * *`
		`* *`
		`*`
			`*`
			`* *`
			`* * *`
			`* *`
			`*`
				✔

✔	`print_star(4);`	`*`
		`* *`
		`* * *`
		`* * * *`
		`* * *`
		`* *`
		`*`
			`*`
			`* *`
			`* * *`
			`* * * *`
			`* * *`
			`* *`
			`*`
				✔

✔	`print_star(5);`	`*`
		`* *`
		`* * *`
		`* * * *`
		`* * * * *`
		`* * * *`
		`* * *`
		`* *`
		`*`
			`*`
			`* *`
			`* * *`
			`* * * *`
			`* * * * *`
			`* * * *`
			`* * *`
			`* *`
			`*`
				✔

	Test	Expected	Got	

✔	`print_star(6);`	`*`
		`* *`
		`* * *`
		`* * * *`
		`* * * * *`
		`* * * * * *`
		`* * * * *`
		`* * * *`
		`* * *`
		`* *`
		`*`
			`*`
			`* *`
			`* * *`
			`* * * *`
			`* * * * *`
			`* * * * * *`
			`* * * * *`
			`* * * *`
			`* * *`
			`* *`
			`*`
				✔

✔	`print_star(7);`	`*`
		`* *`
		`* * *`
		`* * * *`
		`* * * * *`
		`* * * * * *`
		`* * * * * * *`
		`* * * * * *`
		`* * * * *`
		`* * * *`
		`* * *`
		`* *`
		`*`
			`*`
			`* *`
			`* * *`
			`* * * *`
			`* * * * *`
			`* * * * * *`
			`* * * * * * *`
			`* * * * * *`
			`* * * * *`
			`* * * *`
			`* * *`
			`* *`
			`*`
				✔

✔	`print_star(8);`	`*`
		`* *`
		`* * *`
		`* * * *`
		`* * * * *`
		`* * * * * *`
		`* * * * * * *`
		`* * * * * * * *`
		`* * * * * * *`
		`* * * * * *`
		`* * * * *`
		`* * * *`
		`* * *`
		`* *`
		`*`
			`*`
			`* *`
			`* * *`
			`* * * *`
			`* * * * *`
			`* * * * * *`
			`* * * * * * *`
			`* * * * * * * *`
			`* * * * * * *`
			`* * * * * *`
			`* * * * *`
			`* * * *`
			`* * *`
			`* *`
			`*`
				✔

✔	`print_star(9);`	`*`
		`* *`
		`* * *`
		`* * * *`
		`* * * * *`
		`* * * * * *`
		`* * * * * * *`
		`* * * * * * * *`
		`* * * * * * * * *`
		`* * * * * * * *`
		`* * * * * * *`
		`* * * * * *`
		`* * * * *`
		`* * * *`
		`* * *`
		`* *`
		`*`
			`*`
			`* *`
			`* * *`
			`* * * *`
			`* * * * *`
			`* * * * * *`
			`* * * * * * *`
			`* * * * * * * *`
			`* * * * * * * * *`
			`* * * * * * * *`
			`* * * * * * *`
			`* * * * * *`
			`* * * * *`
			`* * * *`
			`* * *`
			`* *`
			`*`
				✔

	Test	Expected	Got	
✔	print_star(10);	* *	* *	✔
✔	print_star(11);	* *	* *	✔
✔	print_star(12);	* *	* *	✔

Passed all tests! ✔

Question author's solution (C):

```c
void print_star(int n)
{
    int i, j;
```

```
    for (i = 1; i <= n; i++)
    {
        for (j = 1; j <= i; j++)
        {
            printf("* ");
        }
        printf("\n");
    }
    for (i = n-1; i >0; i--)
    {
        for (j = 1; j <= i; j++)
        {
            printf("* ");
        }
        printf("\n");
    }

}
```

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 8

Chính xác

Điểm 1,00 của 1,00

Write the function bool completeNum(int N) that checks if a positive integer N is a complete number. N is a complete number if and only if N is equal to the sum of all of its positive divisors (excluding itself)

Input:

- int N: positive integer N to be checked

Output:

- bool: return true if N is a complete number, otherwise return false

For example:

Test	Result
`printf("%s",completeNum(28)? "true":"false");`	`true`

Answer: (penalty regime: 0 %)

Reset answer

```
bool completeNum(int N)
{
    // TODO
    int sum=0;
    for (int i=1;i<N;i++){
        if(N%i==0){
            sum+=i;
        }
    }
    if(sum==N){
        return 1;
    } else {
        return 0;
    }
}
```

	Test	Expected	Got	
✔	`printf("%s",completeNum(28)? "true":"false");`	`true`	`true`	✔
✔	`printf("%s",completeNum(12)? "true":"false");`	`false`	`false`	✔
✔	`printf("%s",completeNum(6)? "true":"false");`	`true`	`true`	✔
✔	`printf("%s",completeNum(2)? "true":"false");`	`false`	`false`	✔
✔	`printf("%s",completeNum(496)? "true":"false");`	`true`	`true`	✔
✔	`printf("%s",completeNum(100)? "true":"false");`	`false`	`false`	✔
✔	`printf("%s",completeNum(49)? "true":"false");`	`false`	`false`	✔
✔	`printf("%s",completeNum(1)? "true":"false");`	`false`	`false`	✔
✔	`printf("%s",completeNum(421)? "true":"false");`	`false`	`false`	✔
✔	`printf("%s",completeNum(124)? "true":"false");`	`false`	`false`	✔

Passed all tests! ✔

Question author's solution (C):

```c
bool completeNum(int N)
{
    // TODO
    int sum = 0;
    for(int i = 1; i < N; i++)
    {
        if(N % i == 0)
            sum += i;
    }
    if(N == sum) return true;
    else return false;
}
```

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 9

Chính xác

Điểm 1,00 của 1,00

Write the function gcdRecursion to perform the greatest common divisor by recursion.

Input:

Two integers p, q respectively (1 ≤ p,q < 10^9).

Output:

The gcdRecursion functions return the greatest common divisor of p, q, respectively.

For example:

Test	Result
`printf("%d",gcdRecursion(6,9));`	3

Answer: (penalty regime: 0 %)

Reset answer

```
int gcdRecursion(int p, int q)
{
    // TODO
    // Euclidean algorithm
    if (q == 0) {
        return p;
    } else {
        return gcdRecursion(q, p % q);
    }

}
```

	Test	Expected	Got	
✔	`printf("%d",gcdRecursion(6,9));`	3	3	✔
✔	`printf("%d",gcdRecursion(5,5));`	5	5	✔
✔	`printf("%d",gcdRecursion(121,135));`	1	1	✔
✔	`printf("%d",gcdRecursion(12,24));`	12	12	✔
✔	`printf("%d",gcdRecursion(62,39));`	1	1	✔
✔	`printf("%d",gcdRecursion(16,39));`	1	1	✔
✔	`printf("%d",gcdRecursion(25,50));`	25	25	✔
✔	`printf("%d",gcdRecursion(40,60));`	20	20	✔
✔	`printf("%d",gcdRecursion(123,123));`	123	123	✔
✔	`printf("%d",gcdRecursion(10,2));`	2	2	✔

Passed all tests! ✔

Question author's solution (C):

```c
int gcdRecursion(int p, int q)
{
    // BEGIN YOUR IMPLEMENTATION [1]
    // TODO
   return (q==0) ? p : gcdRecursion(q,p%q);
    // END YOUR EMPLEMENTATION [1]
}
```

Chính xác

Điểm cho bài nộp này: 1,00/1,00.