

## REC-CIS

[Finish review](#)

## Question 1

Correct

Marked out of  
1.00[Flag question](#)

## Objective

In this challenge, we're going to use loops to help us do some simple math. Check out the Tutorial tab to learn more.

## Task

Given an integer,  $n$ , print its first 10 multiples. Each multiple  $n \times i$  (where  $1 \leq i \leq 10$ ) should be printed on a new line in the form:  $n \times i = \text{result}$ .

## Input Format

A single integer,  $n$ .

## Constraints

$$2 \leq n \leq 20$$

## Output Format

Print 10 lines of output; each line  $i$  (where  $1 \leq i \leq 10$ ) contains the **result** of  $n \times i$  in the form:

$n \times i = \text{result}$ .

## Sample Input

2

## Sample Output

$$2 \times 1 = 2$$

$$2 \times 2 = 4$$

$$2 \times 3 = 6$$

$$2 \times 4 = 8$$

$$2 \times 5 = 10$$

$$2 \times 6 = 12$$

$$2 \times 7 = 14$$

$$2 \times 8 = 16$$



REC-CIS

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n;
5     scanf("%d",&n);
6     for(int i=1;i<=n;i++)
7     {
8         printf("%d x %d = %d\n",n,i,n*i);
9     }
10 }
```

	Input	Expected	Got	
✓	2	2 x 1 = 2 2 x 2 = 4 2 x 3 = 6 2 x 4 = 8 2 x 5 = 10 2 x 6 = 12 2 x 7 = 14	2 x 1 = 2 2 x 2 = 4 2 x 3 = 6 2 x 4 = 8 2 x 5 = 10 2 x 6 = 12 2 x 7 = 14	✓

## Question 2

Correct

Marked out of 1.00

Flag question

A nutritionist is labeling all the best power foods in the market. Every food item arranged in a single line, will have a value beginning from 1 and increasing by 1 for each, until all items have a value associated with them. An item's value is the same as the number of macronutrients it has. For example, food item with value 1 has 1 macronutrient, food item with value 2 has 2 macronutrients, and incrementing in this fashion.

The nutritionist has to recommend the best combination to patients, i.e. maximum total of macronutrients. However, the nutritionist must avoid prescribing a particular sum of macronutrients (an 'unhealthy' number), and this sum is known. The nutritionist chooses food items in the increasing order of their value. Compute the highest total of macronutrients that can be prescribed to a patient, without the sum matching the given 'unhealthy' number.

Here's an illustration:

Given 4 food items (hence value: 1, 2, 3 and 4), and the unhealthy sum being 6 macronutrients, on choosing items 1, 2, 3  $\rightarrow$  the sum is 6, which matches the 'unhealthy' sum. Hence, one of the three needs to be skipped. Thus, the best combination is from among:

- $2 + 3 + 4 = 9$
- $1 + 3 + 4 = 8$
- $1 + 2 + 4 = 7$

Since  $2 + 3 + 4 = 9$ , allows for maximum number of macronutrients, 9 is the right answer.

Complete the code in the editor below. It must return an integer that represents the maximum total of macronutrients, modulo  $1000000007$  ( $10^9 + 7$ ).

It has the following:

$n$ : an integer that denotes the number of food items

REC-CIS

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     long long int n,t,i,nut=0;
5     scanf("%lld %lld",&n,&t);
6     for(i=1;i<=n;i++)
7     {
8         nut=nut+i;
9         if(nut==t)
10             nut=nut-1;
11     }
12     printf("%lld",nut%1000000007);
13
14 }
```

	Input	Expected	Got	
✓	2 2	3	3	✓
✓	2 1	2	2	✓
✓	3 3	5	5	✓

## Question 3

Correct

Marked out of 1.00

Flag question

Determine all positive integer values that evenly divide into a number, its factors. Return the  $p^{\text{th}}$  element of your list, sorted ascending. If there is no  $p^{\text{th}}$  element, return 0.

For example, given the number  $n = 20$ , its factors are  $(1, 2, 4, 5, 10, 20)$ . Using **1-based indexing** if  $p = 3$ , return 4. If  $p > 6$ , return 0.

Complete the code in the editor below. The function should return a long integer value of the  $p^{\text{th}}$  integer factor of  $n$ .

It has the following:

$n$ : an integer

$p$ : an integer

**Constraints**

$$1 \leq n \leq 10^{15}$$

$$1 \leq p \leq 10^9$$

Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer  $n$ , the number to factor.

The second line contains an integer  $p$ , the 1-based index of the factor to return.



## REC-CIS

```
1 #include<stdio.h>
2 long pthFactor(long n,long p)
3 {
4     int count=0;
5     for(long i=1;i<=n;++i)
6     {
7         if(n%i==0)
8         {
9             count++;
10            if(count==p)
11                return i;
12        }
13    }
14    return 0;
15 }
16 int main()
17 {
18     int n,p,o;
19     scanf("%d",&n);
20     scanf("%d",&p);
21     o=pthFactor(n,p);
22     printf("%d",o);
23 }
```

	Input	Expected	Got	
✓	10 3	5	5	✓
✓	10 5	0	0	✓
✓	1 1	1	1	✓