

CS101: Computer Programming and Utilization, Spring 2020

Lab4 (L1), Tuesday 4th February 2020

Instructions and Program Submissions: You will have to submit all the programs on cs101.bodhi.cse.iitb.ac.in.

Deadline : **Tuesday, 4th February 2020, 10:30 PM**

Q1. Write a program which calculates the value $n! \% m$, where % is the modulo operator($a \% b$ is the remainder obtained on dividing a by b ex: $23 \% 5$ is 3)

Constraints : $n \leq 10^5$, $m \leq 10^9$

Hint : Note that the value a^n can be of the order 100000!, which can't be stored in any variable type in c++. Use repeat statement, long int (or any other integer type variable which can store 10^{14}) and the fact that $(a*b) \% m$ equals $((a \% m) * (b \% m)) \% m$.

Filename for Code: **mod.cpp**

Input Format:

The first and only line of the input consists of **n and m** ,separated by spaces.

Output Format:

Print a single integer, the result of the expression.

Sample Input:

5 7 // n=5, m=7

Sample Output: 1

Q2. Write a program which calculates the Euclidean distance between two n-dimensional points.

Consider 2 n-dimensional points $A = (a_1, a_2, a_3, \dots, a_n)$ and $B = (b_1, b_2, b_3, \dots, b_n)$ then calculate the,

Euclidean distance = $\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2 + \dots + (a_n - b_n)^2}$

Note: To print 2 decimal digit precision, use “**printf**” instead of “**cout**”

```
float answer = 2.342;
printf("%.2f\n", answer); // prints 2.34 followed by a newline
```

Filename for code: **euclidean_distance.cpp**

Input Format:

First-line will be a single integer Q, denoting the number of queries.

For each query, there will be an input for N, denoting the dimension of points followed by N lines for a_i and b_i respectively.

Output Format:

In each line, for every query, the output will be a single value (**2 decimal digit precision**) denoting the euclidean distance for that query.

Input:

```
2      //Number of Queries Q
4      //Query 1: N, Dimension of points A and B
1 2    // a_1 b_1
2 3    // a_2 b_2
3 4    //a_3 b_3p
4 5    //a_4 b_4
2      //Query 2: N, Dimension of points A and B
-1 1   //a_1 b_1
3 4.2 //a_2 b_2
```

Output:

```
2.00    //Result of Query 1 (Notice the 2 decimal digit precision)
2.33    //Result of Query 2
```

Q3. You are given a stream of natural numbers. You need to find the largest number in the stream. Assume that the stream always ends with -1.

The filename for the code: **stream.cpp**

Input Format: A stream of natural numbers separated by spaces

Output Format: A single integer equal to the largest number in the stream

Sample Input:

10 5 6 435 90 -1

Sample Output:

435

Q4. Take the number of queries from input(**Q**). Then for following **Q** queries, given number **A**, print how many of the first 5 prime numbers(2, 3, 5, 7, 11) divide the number **A**.

Hint: Try solving the problem with **switch case** construct.

Filename: **count.cpp**

Sample Input:

```
5      //Number of Queries Q
121    //Query 1: number A
10     //Query 2: number A
13     //Query 3: number A
22     //Query 4: number A
770    //Query 5: number A
```

Sample Output:

```
1      //Query 1 output, divisible by 11
2      //Query 2 output, divisible by 2, 5
0      //Query 3 output, divisible by none
2      //Query 4 output, divisible by 2, 11
4      //Query 5 output, divisible by 2, 5, 7, 11
```