1. Problem Name: Exceptional

Problem Category: No Category [Print]

Problem Setter: Khairul Anam Mubin, Intake: 38

Problem: There is no input of this problem. You have to print anything except “Hello World”.

Solution:

There is a solution which is to print nothing.

1. Problem Name: BIUPC Division

Problem Category: Implementation [If-else]

Problem Setter: Faisal Mahmud Rifat, Intake: 38

Problem: You will be given an intake. You have to find the division of that intake.

Solution:

If the given intake is 39 to 42 then it is Senior division,

If the given intake is 43 to 46 then it is Junior division,

else for other intakes are out of the division.

1. Problem Name: Count Odds

Problem Category: Math

Problem Setter: Khairul Anam Mubin, Intake: 38

Problem: You will be given a range L, R. You have to find the number of odd integers in range L, R(inclusive).

Solution 1 (Loop): As the input range is small so we can iterate through L to R, and count the odd numbers.

Solution 2(Math): Calculate ((R + 1) / 2) – (L / 2) and output the result.

1. Problem Name: Meme

Problem Category: String

Problem Setter: Jannatul Ferdows Modina, Intake: 35

Problem: You are given 5 strings. You have to find if all the strings are length of five or not.

Solution:

Let We’ve a variable cnt = 0 initially. For each of the input strings we have to find its length. If the length is equal to 5 then cnt = cnt + 1, that means increment 1 to the cnt. Lastly, we have to check if cnt is equal to 5 or not. If cnt is equal to five then output “Mim” else output “Pocha Dim”

1. Problem Name: Ghor Mapa Shikho

Problem Category: Math

Problem Setter: Walid Khan Jim, Intake: 35

Problem: You will be given N floating point numbers maximum of 2 digits after the decimal point. You have to separate this given floating number from decimal point as feet and inches. After it sum up the inputs continuously one by one and have to say the current total feet and inches.

Solution:

Let total\_feet = 0 and total\_inches = 0 initially.

There will be input like 7.4 . We have to separate 7 and 4 from decimal point and let 7 is the feet and 4 is the inches. We can separate these in C language by taking the input like scanf(“%d.%d”,&feet, &inches); . After this, update total\_feet = total\_feet + feet and total\_inches = total\_inches + inches. As 12 inch = 1 feet, so we have to update total\_feet = total\_feet + total\_inches / 12 and update total\_inches = total\_inches % 12. After it output the total\_feet and total\_inches for each line of input.

1. Problem Name: Divisible Number

Problem Category: String + Number Theory

Problem Setter: H.M Safayat Jakir, Intake: 37

Problem: You are given a number N. You have to check if the number is divisible by 18 or not.

Solution:

As the input is very large around 1000 digits so we have to take the inputs in string. Let that is s. Now, there are some divisible rules in number theory. For 18, the rule is if the sum of the digits of the number is divisible by 9 and if the last digit is even, then whole number is divisible by 18.

So, let sum = 0 initially, we have to iterate through the string and let ith character of that is s[i]. now, we will sum the whole string like sum = sum + s[i] – ‘0’ (subtracting ‘0’ because it will convert a character to integer). Let last character is last\_digit = s[len – 1] – ‘0’ (here len is the length of the string s).

Now, if the sum % 9 == 0 and last\_digit % 2 == 0 then it will be divisible 18 else not.

1. Problem Name: Giving Candies

Problem Category: Combinatorics

Problem Setter: Khairul Anam Mubin, Intake: 38

Problem: Given N box of candies and ith box have ai number of candies. Each of the box have different types of candies. If you want to give some candies to your friend how many ways you can do it?

Solution: Let, the box has ai number of candies. If we say, how many ways we can take from first box? there will a1 ways. Again, if we say, how many ways we can take from box first and second? Then there comes choices. We can take or not to take. If we take 1 candy from first box then we can take or not to take from second box. But if we take from second box, then there are a2 choices. And there also a choice not to take from the second. So, there are a2 + 1 choices. Now there are a1 candies in first box, so there will be a1 + 1 choices as the second box. So, there are total of (a1 + 1) \* (a2 + 1) choices. But from first and second, we have to take at least one because if I don’t give any candy to my friend that won’t be a way.

So, considering the above discussion, we can do it in (a1 + 1) \* (a2 + 1) \* … \* (an + 1) – 1 way.

As the answer could be quite large, we have to output modulo 1e9 + 7

1. Problem Name: Jumping Frog

Problem Category: DP

Problem Setter: Khairul Anam Mubin, Intake: 38

Problem: A frog currently at position 1 and have to reach in position N. He can jump from position i to i \* 2 position with cost a, i ^ 1 with cost b, i + 1 with cost c, i – 1 with cost d.

You have to find the minimum cost to reach n.

Solution:

Here we see, for odd i, i ^ 1 gives i – 1 and for even i, i ^ 1 gives i + 1. And i \* 2 gives longest jump only in even positions. And it can be happened that in a jump we have reached a position which is greater than n. So, in that case, we can come back in n with jump i ^ 1 or i – 1. So that means in every way we have to try to reach n and minimizing the cost and here comes the concept of DP.

Let, array dp[] size 2 \* n + 1 initially have some bigger value. Set dp[1] = 0, as from 1 to 1 cost should be 0.

Now, for each i = 1 to 2 \* n

dp[i \* 2] = min(dp[i \* 2] , dp[i] + a)

dp[i ^ 1] = min(dp[i ^ 1] , dp[i] + b)

dp[i + 1] = min(dp[i + 1], dp[i] + c)

dp[i – 1] = min(dp[i – 1], dp[i] + d)

After this, the required minimized cost will be in dp[n].

1. Problem Name: After pandemic tour

Problem Category: Segment tree + binary Search

Problem Setter: Khairul Anam Mubin, Intake: 38

Problem: There are M groups of students. Also, there are N buses in a row. For each bus you know the number of free seats. You task is to assign seats for group of students. All students of a group want to ride in the same bus. The group will come to you one after another, and you know for each group the number of seats it requires. You always assign a group to the first bus having enough seats. After this, the number of free seats in the bus decreases.

Solution: We can maintain prefix maximum with segment tree and binary search for the required bus where we have to assign the ith group. After this, update this position in segment tree.