CodeChef CGC Chapter

Weekly-Challenges

Week 1

Problem -1 LUCKFOUR

Problem:
Count to number of 4s in the given number.

Trick:
Input the digit as a string instead of taking it as a integer.

Solution:
Iterate over each digit(character) in order to find the total count of digit 4.

Pseudo Code

```
Input string NUMBER
Set COUNT := 0
Loop for i from 0 to LENGTH(NUMBER):
    If NUMBER[i] == ‘4’ then Set COUNT := COUNT + 1
Output COUNT
```

Time Complexity

O(n)

Space Complexity

O(1)

Problem -2 HS08TEST

Problem:
Check if amount to be withdrawn is multiple of 5 and lesser than available balance (along with the added cost of transaction).

Solution:
Check if X is multiple of 5 and X + 0.5 <= Y,
If both the conditions are true then print Y - X - 0.5 else print Y.

Pseudo Code

```
Input integer $X$ and $Y$
If $X\%5==0$ and $X+0.5 <= Y$ then Set $Y := Y - X - 0.5$
Output $Y$
```

Time Complexity

O(1)

Space Complexity

O(1)

Problem -3 MUFFINS3

Problem:
Find the size of package for which the leftover cupcakes will be the maximum. If multiple such solutions exist then print the largest one.

Trick:
Find the minimum package size such that only one individual package can be made out of all the cupcakes.

Solution:
Calculate $\lfloor N / 2 \rfloor + 1$

Pseudo Code

```
Input integer N
Output $\lfloor N / 2 \rfloor + 1$
```

Time Complexity

$O(1)$

Space Complexity

$O(1)$

Problem -4 SUMTRIAN

Problem:
Compute the largest of the sums of numbers that appear on the paths starting from the top towards the base. Next element for each number is either directly below or one place to the right.

Trick:
Instead of finding the path downwards, move upwards with highest summations.

Solution:
Move from the last layer towards the first adding max of two to its predecessor.

Pseudo Code

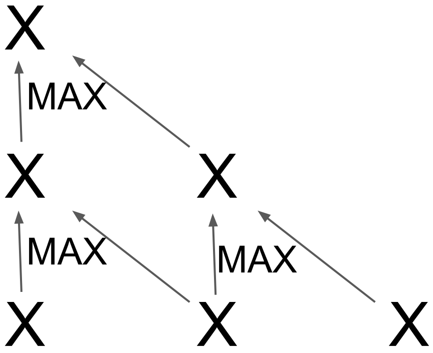
```
Input integer S
Input integer[S][S] TRIAN
Loop for i from S-1 to 1:
    Loop for j from 0 to i-1:
        TRIAN[i-1][j] = TRIAN[i-1][j] + max(TRIAN[i][j], TRIAN[i][j+1])
Output TRIAN[0][0]
```

Time Complexity

$O(n^2)$

Space Complexity

$O(1)$


X
MAX
X
X
MAX
MAX
X
X
X