PLACEMENTS KEY'S

Coding Challenge #35 (Question)

Day - 18

Character count in given String

Given an Input string aaaabbccdee. Count the number of same consecutive characters and give the output as a4b2c2d1e2.

Assume: Maximum Length of the string is 20.

Sample Input 0:

Aaaabbccdee

Sample Output 0:

A4b2c2d1e2

Sample Input 1:

AAAZZZZZCCBA

Sample Output 1:

A3Z5C2B1A1

PLACEMENTS KEY'S

Coding Challenge #35 (C Solution)

Day - 18

```
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
int main(int c,char **arg)
{
        char a[30];
        scanf("%s",a);
        int count=1,i,len=0;
        for(len=0;a[len]!='\0';len++);
        if(len>20)
        {
                printf("Invalid Input");
                return 0;
        }
        for(i=1;i<=len;i++){
            if(tolower(a[i])==tolower(a[i-1])){
                        count++;
                }
                else{
                        printf("%c%d",toupper(a[i-1]),count);
                        count=1;
                }
  }
}
```

PLACEMENTS KEY'S

Coding Challenge #35 (JAVA Solution)

Day - 18

```
import java.util.*;

import java.lang.Math;

class Main {

      static void print(String s)

    {

      for (int i = 0; i < s.length(); i++)

{

         int count = 1;

        while (i + 1 < s.length() && s.charAt(i) == s.charAt(i + 1)) {

           i++;

           count++;

         }

        System.out.print(s.charAt(i)

                  + "" + count);

     }

     }
```

PLACEMENTS KEY'S

Coding Challenge #35 (JAVA Solution contd.)

Day - 18

```
public static void main(String args[])
    {
      String str1;
      Scanner sc=new Scanner(System.in);
      str1=sc.nextLine();
       if(str1.length()>20)
      {
         System.out.println("Invalid Input");
         System.exit(0);
       }
       else
            print(str1);
    }
}
```

PLACEMENTS KEY'S

Coding Challenge #36 (Question)

Day - 18

Roots of a quadratic equation

Write a program to find the roots of a given quadratic equation.

The equation will be in the form of $ax^2 + bx + c = 0$. The input will be 3 integers a, b and c and the output will be the roots of the equation. The roots need to be floating-point integers with 2 precision digits.

Output format example:
root1 = -1.00 root2 = -6.00

Sample Input 0:

1
-1
-6

Sample Output 0:

root1 = 3.00 root2 = -2.00

Sample Input 1:

1
7
6

Sample Output 1:

root1 = -1.00 root2 = -6.00

PLACEMENTS KEY'S

Coding Challenge #36 (C Solution)

Day - 18

```c
#include<stdio.h>
#include<math.h>
int main()
{
  float a, b, c, determinant, r1, r2, real, img;
  scanf("%f %f %f",&a, &b, &c);
  determinant = b*b - 4*a*c;
  if (determinant  > 0)
  {
          r1 = (-b + sqrt(determinant)) / (2 * a);
          r2 = (-b - sqrt(determinant)) / (2 * a);
          printf("root1 = %.2f  root2 = %.2f", r1 , r2);
  }
  else if (determinant == 0)
  {
          r1 = r2 = -b / (2*a);
    printf("root1 = %.2f  root2 = %.2f", r1 , r2);
  }
else
  {
          real = -b/(2*a);
    img = sqrt(-determinant) / (2*a);
    printf("root1 = %.2f + %.2fi  root2 = %.2f - %.2fi", real, img, real, img);
  }
  return 0;
}
```

PLACEMENTS KEY'S

Coding Challenge #36 (JAVA Solution)

Day - 18

```
import java.util.*;
import java.lang.*;
class Main
 {
 public static void main(String args[])
   {
      Scanner sc=new Scanner(System.in);
      float a, b, c, x1, x2, discriminant, realPart, imaginaryPart;
     a=sc.nextFloat();
      b=sc.nextFloat();
     c=sc.nextFloat();
     discriminant = b*b - 4*a*c;
           if (discriminant > 0)
        {
            x1 = (-b + (float)Math.sqrt(discriminant)) / (2*a);
             x2 = (-b - (float)Math.sqrt(discriminant)) / (2*a);
             System.out.printf("root1 = %.2f  ",x1);
            System.out.printf("root2 = %.2f",x2);
      }
```

PLACEMENTS KEY'S

Coding Challenge #36 (JAVA Solution contd.)

Day - 18

```
else if (discriminant == 0)
{
    x1 = (-b + (float)Math.sqrt(discriminant)) / (2*a);
     System.out.printf("root1 = %.2f  ",x1);
     System.out.printf("root2 = %.2f ",x1);
  }
  else
{
    realPart = -b/(2*a);
    imaginaryPart =(float)Math.sqrt(-discriminant)/(2*a);
    System.out.printf("root1 = %.2f + %.2fi  ",realPart,imaginaryPart);
     System.out.printf("root2 = %.2f - %.2fi",realPart,imaginaryPart);
  }
  }
 }
```