# Teramatrix Assignment

**TeraMatrix Technologies P.L**
**Company Overview**

Teramatrix's data aggregation platform is responsible for collecting data from various end points, normalise the data and make it available for end user to visualise and analyse.

**ASSIGNMENT 1 :**
**PROGRAMMING ASSIGNMENT**

Coding Assignment

In this assignment we expect the working code with documentation. We'll evaluate your Programming skills, problem solving skills, the complexity (space and time)

Q. Write a program to generate COMPLEX PASSWORD for a user having following functionality

1. Able to generate simple password as well as complex password
2. Min Length and Max Length
3. Number of lower case characters (min, max)
4. Number of upper case characters (min, max)
5. Number of numerics  (min, max)
6. Number of special characters  (min, max)

Please code it how you generally code.

————————————————————————————————

**ASSIGNMENT 2 :**
**PROGRAMMING ASSIGNMENT**

Without Using any CONDITIONAL STATEMENT [i.e if-else, switch] print the first EVEN DIGIT from the RIGHT, if exists else print -1.

Input
File Q1.in first line will have an Integer value [x], where $0 < x < 100$
From Next line till the EOF will contain an integer number [y], where $0 < y < 1000000$, one integer (i.e one test case input) per line.
Output
File should be Q1.out; it will have first case as number of test cases [x]

And then one output per line, represented as
 Case#<output number> = <Output>

<u>Sample Input</u>
4
123
123456
654321
3


<u>Sample Output</u>
4
Case #1 = 2
Case#2 = 6
Case#3 = 2


————————————————————————————————————

ASSIGNMENT 3 :
PROGRAMMING ASSIGNMENT

Mr. Walker is a person who is known for his ability to walk fast. He accepts an attractive offer to get as much free land as he can cover by a walk on an open uneven huge piece of land within a specified time and following certain conditions. A walk is a sequence of paths, starting at a marked spot on the land and ending at the same spot where the walk starts. The first path in a walk begins at the marked spot and extends in one of the four directions: North (N), South (S), East (E) or West (W). It ends at the point where the direction changes. Each  of the  other  paths  in  a  walk  begins  at  the  point  where  the  previous  path ends  and  ends either at a point where the direction changes or when the walk ends. Mr. Walker may change directions, as  and  when  he  feels  like, depending  on  the  constraint  of  time  and/or  quality/quantity  of  the  land  he decides  to  get.  When  he  changes  a  direction  he  must  keep  the  direction always  to  one  of  the  four directions  N,  S,  E  or W.  Paths  are  distinct  and nonintersecting;  for  example,  a  path  in  E  or  W  direction  cannot  cross another in N or S direction.

Given a walk, you are required to write a program that finds the area of land covered by the walk.

<u>Input</u>
The input consists of multiple test cases.
First line will be number of test cases, followed by the test cases, one per line.
The line gives a walk defined by a sequence of paths. A direction followed, without any space, by a distance represents a path. A direction is denoted by one of the four letters N, S, E or W while a distance is measured in meters and is denoted by an integer.
A space character appears between two paths in the given sequence.

<u>Output</u>
For each test case print the area of land covered by the given walk.

<u>Sample Input</u>
3
N3 W4 S8 E4 N5
W6 N2 E9 S6 W3 N4
W6 N2 W3 S4 W5 S4 E14 N6

<u>Sample Output</u>
3
Case #1 = 32
Case #1 = 30
Case #1 = 80

———————————————————————————————————————

<u>Application: Online Debate Portal</u>

Application Summary: ODP is an open web based debate portal, where registered users can start debating (for the motion or against the motion) on a provided list of running debates.

<u>Application Modules:</u>
· Debate List
· User Profile
· Debate Result

<u>Application Constraints:</u>
· Application will have ACL based on user roles.
· Application will support Administrator, Moderator, Debater and Guest roles
· Application will only allow registered and logged in users to debate
· Guest users can only read the debates list but cannot participate in any debate
· Moderators will be responsible for maintaining the debate content, manage an altercation, keeping track of user infractions and banning a volatile user
· Administrators can post a new debate topic, close an old topic, add or delete a user, assign user a particular role, choose the winners of a debate, close a running debate and if required open a closed debate
· Debaters will be able to post their views over a debate, they can challenge another debater over any running debate topic, can choose the debate side  (for the motion or against the motion) once per debate topic.
· Debates will be one on one, if the participants changes the same debate topic will host a new debate automatically
· Other debaters can vote up or vote down a debater's point of view
· User profile will be visible to all, and it will carry information regarding, debates participated in, debates won, debates lost, points earned

· A user with good number of points will be chosen as moderator by the site administrator
· Debater can suggest a new debate topic
· Debaters should be searchable
· Debate List should be searchable

Assignment Details:
· Draw and Application Architecture for the Debate Portal
· Class diagram for the Application
· Draw an ER schema OR Database Schema OR Database Diagrams
· Draw a Class Structure OR UML diagram
· Clearly write all the possible functions or API definition
· Draw a workflow OR flowchart for the application
· Prototype the Application GUI
· Project plan and timelines are worked upon

Application Technology:
· Language: Python
· Platform: Any
· Database: Any
· Framework: Django
· Other Technology Stack (Of your choice)