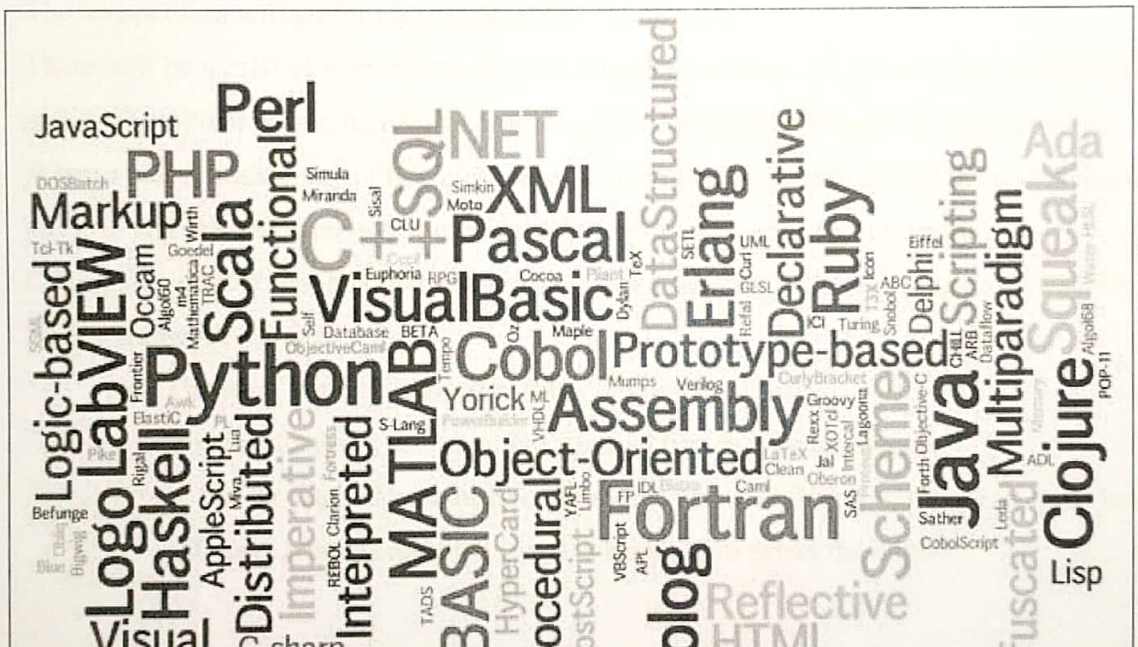
R·I·T Dubai جامعة روتشستر للتكنولوجيا - نيويورك، أمريكا
Rochester Institute of Technology, NY, USA

RIT-DUBAI ENGINEERING COMPETITION 2018

PROGRAMMING COMPETITION



November 1st, 2018

RULES & INSTRUCTIONS

- You are not allowed to use online resources (i.e. Google or stackexchange),
- You are allowed to use "Help" provided by the IDE to get information about built-in classes, functions and methods.
- You can choose any programming language you want (i.e. Java, C++, Python ... etc) to solve the competition problems.
- At the University, we have:
  - Code:Blocks (17.12), Python (2.7), Eclipse Mars.1 Release (4.5.1), JGrasp (version 2.0.1_09) and BlueJ (version 3.1.5).
- You are allowed to use your own laptops and you are welcome to use our computers in our lab.
- The competition will go for about 1 hour and 15 minutes.
- There will be a total of 6 problems that are designed on three levels (easy, med and hard) and each problem have a number of points assigned to it (50, 100 and 150 respectively).
- When a team finishes any of the questions, the team members need to call the competition judge to test the code and give the points (if everything is working properly).
- Calling the judge to test a nonworking code or a wrong code will result in losing 10 points.
- To determine the winner:
  - The team with the highest number of points will win.
  - In case of a draw, the team that finished first will be the winner.
  - In the far-fetched case of two (or more) teams finishing at the exact time, we will have a bonus question (i.e. Tie Breaker) to select the winner.

A Word of advice:

Before making an attempt on any problem, make sure you have considered the input and time constraints as specified in the problem statement and your output is as per the output format. When you are sure, just raise your hand and the judge will be there to check your solution.

Wish You All the Best!!

Problem 1 **50 Points**

Pyramids

In this simple problem, the program should generate a pyramid shape of asterisks based on the number of layers provided by the user.

Example 1:
Sample input:
Number of levels: 2

Sample output:

```
 *
**
```

Example 1:
Sample input:
Number of levels: 3

Sample output:

```
  *
 **
***
```

Example 1:
Sample input:
Number of levels: 5

Sample output:

```
    *
   **
  ***
 ****
*****
```

Problem 2 **50 Points**

Fibonacci Sequence

The Fibonacci sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, … etc. where the next number is found by adding up the two numbers before it.

You program should ask the user to input an integer 'n' for nth sequence number then it should be able to calculate the value of that number.

Example 1:
Sample input:
Enter Fibonacci Sequence index: 5

Output:
Value: 5

Example 2:
Sample input:
Enter Fibonacci Sequence index: 6

Output:
Value: 8

Example 3:
Sample input:
Enter Fibonacci Sequence index: 7

Output:
Value: 13

Problem 3 **100 Points**

Caesar Cipher

The Caesar cipher is one of the earliest known and simplest ciphers. It is a type of substitution cipher in which each letter in the plaintext is 'shifted' a certain number of places down the alphabet. For example, with a shift of 1, A would be replaced by B, B would become C, and so on. The method is named after Julius Caesar, who apparently used it to communicate with his generals.

Your program should ask the user to enter the shift value (any integer from 1 to 25) and then ask the user to enter a text (string with spaces that will terminate by the return character). Then your program will perform Caesar Cipher and produce an encrypted text based on the shift value entered by the user.

Your code shouldn't worry about encrypting numerical (i.e. 1, 2, … etc) or special characters (whitespace, punctuation … etc). Also, you can

Example 1:
Sample input:
```
Enter shift value (1-25): 5
Enter plain text: hello world
```

Output:
```
Encrypted text: mjqqt btwqi
```

Example 2:
Sample input:
```
Enter shift value (1-25): 3
Enter plain text: welcome to rit
```

Output:
```
Encrypted text: zhofrph wr ulw
```

Example 3:
Sample input:
```
Enter shift value (1-25): 9
Enter plain text: programming is fun
```

Output:
```
Encrypted text: yaxpajvvrwp rb odw
```

Problem 4 **100 Points**

Uncompressing Strings

Given a string that is of the form: “nL” where n is the number of repetition (integer from 1 to 9) and L is the letter to be repeated, you program should be able to read the string and un-compress it to a string of repeated alphabets.

Example 1:
Sample input:
Enter compressed string: 5A

Output:
Uncompressed string is: AAAAA

Example 2:
Sample input:
Enter compressed string: 2A3B

Output:
Uncompressed string is: AABBB

Example 3:
Sample input:
Enter compressed string: 1C3Z

Output:
Uncompressed string is: CZZZ

Problem 5 **150 Points**

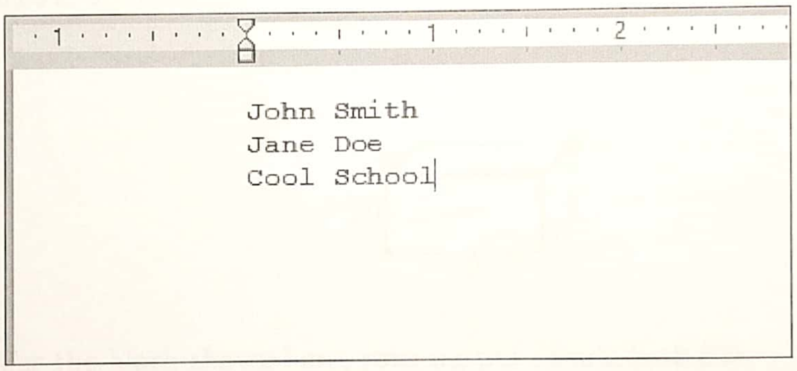
Counter

Write a program that prompts the user for a text filename (example: words.txt - create your own text file and write your names and school name on separate lines), and prints out the total number of characters in the file.

Example:
Sample Input:
Enter filename: words.txt

Sample Output:
Number of characters: 26

Content of Sample file words.txt:



```
John Smith
Jane Doe
Cool School
```

Problem 6 **150 Points**

Bug's Journey

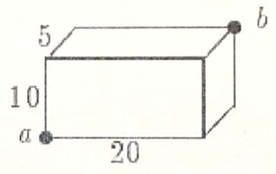
A bug is sitting at vertex ***a*** of a rectangular block and wishes to travel to ***b*** by walking along the surface. Note that the bug can walk anywhere along the surface and not just long the edges.

Your program should ask the user for the length, width and height of the rectangular block (as integers) and then calculates and prints the shortest distance between the two vertices (as double data type – i.e. with decimal point).

Example:
Sample Input:
Enter length: 5
Enter width: 20
Enter height: 10

Sample Output:
Minimum distance is: 25.0

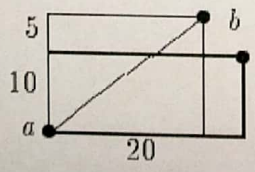
Example:



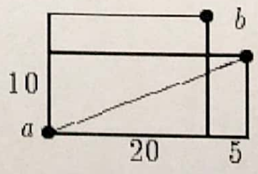
For the block shown here, your output should look like:

```
Length = 5
Width = 20
Height = 10
Minimum distance = 25.0
```

Notes: There are three pairs of possible shortest routes. First unfold the boxes. Then draw lines from a to b and compute the distances. The smaller of the distances is the shortest path. One pair is:



$\sqrt{20^2 + 15^2} = 25.00$



$\sqrt{25^2 + 10^2} = 26.93$