**D Y PATIL COLLEGE OF ENGINEERING & TECHNOLOGY**

**KASABA BAVADA, KOLHAPUR**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

(Academic Year: 2017 - 2018)



A

Project Synopsis On

**“Bob vs ATM”**

Submitted by:

|  |  |  |
| --- | --- | --- |
| **S.N** | **Name** | **Roll. No.** |
| 1 | Dnyaneshwar Mahadeo Raut | 63 |
| 2 | Ruturaj Ravindra Sawant | 66 |
| 3 | Apurv Avinash Kumbhar | 58 |
| 4 | Rajahmed Bashir Sayyad | 68 |

**Under the guidance of**

**Prof. Mrs. D. R. Patil**

**Class: SE Div.: A Batch: S4**

**Introduction :**

Due to shortage of ATMs, a bank decided to convert some arcade gaming machines into ATMs. Unfortunately, due to a bug, the converted machine will only dispense cash if the withdrawer can win the game. The game is played over a valid bracket sequence. Formally a valid bracket sequence is defined as:

1. An empty string is valid

2. If *A* is a valid bracket sequence, then (*A*) is also valid

3. If *A* and *B* are valid, *AB* is also valid

In each move a player can erase such a non-empty valid sub-string of the remaining brackets such that they cannot be formed by concatenating 2 valid sequences. E.g. a player can erase (()) or (()()) but not ()() or (())(). The ATM always plays first. The player that is not able to make a move will lose the game. Assume that both the players play optimally.The first line of the input contains an integer *T* denoting the number of test cases. The description of *T* test cases follows.

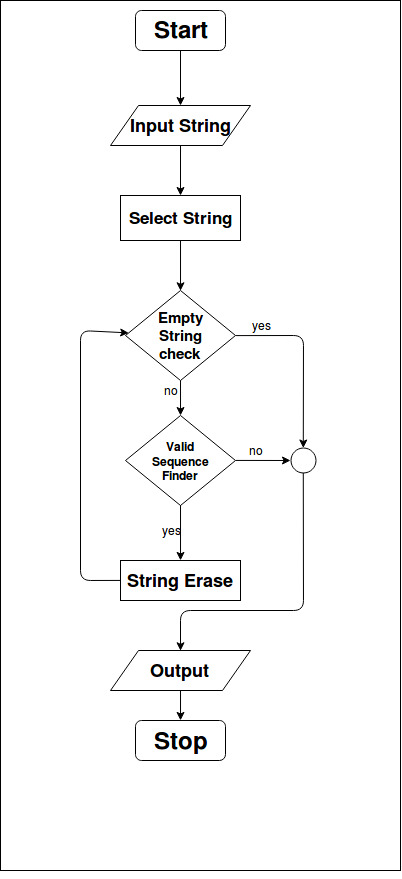
Each of the test case consists of a single line containing the string *S* (the initial bracket sequence).

For each test case, output a single line containing the winner — either ‘ATM’ or ‘Bob’ without quotes.

**Problem Statement:**

Create a program to identify valid sequence of parentheses from given string and erasing found valid sequence until it reduces to invalid or empty string ?

**Flow Chart:**



**Fig. Flow chart**

**Input:**

* The first line of the input contains an integer *T* denoting the number of test cases. The description of *T* test cases follows.
* Each of the test case consists of a single line containing the string *S* — the initial bracket sequence.

**Output:**

* For each test case, output a single line containing the winner — either ‘ATM’ or ‘Bob’ without quotes.

**Constraints :**

* 1 *<= T <=* 10000
* 2  *<= length of S <=* 10000
* The sum of all lengths of *S* will be not more than 1000000
* *S* will be a valid bracket sequence

**Applications:**

* To find correct algorithm and program to beat ATM game’s LOGIC.
* To find correct and wrong bracket sequences.

**Software Requirement Specification:**

**Hardware Requirements: -**

* + RAM : 1 GB
  + Processor : Pentium 4 onwards

**Software Requirements: -**

* + Language : C++
  + Operating system : Windows 10 / Ubuntu 16.04
  + Compiler : GNU G++

**Reference:**

* [www.acmicpc.com](http://www.acmicpc.com)
* [www.google.com](http://www.google.com)
* [www.stackoverflow.com](http://www.stackoverflow.com)
* [www.cppreference.com](http://www.cppreference.com)
* [www.geeksforgeeks.com](http://www.geeksforgeeks.com)

**Conclusion:**

Program is created to determine whether given bracket sequence is valid or not using specified valid bracket sequence definitions.