D. Y. PATIL COLLEGE OF ENGINEERING & TECHNOLOGY

KASABA BAWADA, KOLHAPUR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Academic Year: 2017 - 2018)

A

Mini Project Report

On

"Bob vs ATM"

Submitted by:

Roll No	Name
58	Apurv Kumbhar
63	Dnyaneshwar Raut
66	Ruturaj Sawant
68	Rajahmed Sayyad

Under the guidance of

Prof. Mrs. D. R. Patil

Class: SE (CSE) **Div.: A** **Batch: S4**

D. Y. PATIL COLLEGE OF ENGINEERING & TECHNOLOGY

KASABA BAVADA, KOLHAPUR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DY PATIL
COLLEGE of
ENGINEERING & TECHNOLOGY
KASABA BAWADA

CERTIFICATE

This is to certify that the mini project group consisting of following members have satisfactorily completed the mini project work entitled **"Bob vs ATM"** at SE (CSE) semester-IV as prescribed in the syllabus of Shivaji University for academic year 2017-2018.

Name	Exam Seat no.
1. Kumbhar Apurv Avinash.	4230
2. Raut Dnyaneshear Mahadeo.	4265
3. Sawant Ruturaj Ravindra.	4268
4. Sayyad Rajahmed Bashir.	4135

Date:

Place: Kolhapur

Prof.Mrs. D. R. Patil
(Project Guide)

Prof.Dr.G. A. Patil
(HOD)

(Prof.Mrs. D. R. Patil , Prof.Ms. A. R. Nagaonkar)
(Project coordinators)

Prof.Dr.A.N. Jadhav
(Principal)

(External)

ACKNOWLEDGMENT

This project work entitled **"Bob vs ATM"** was a formidable task, but with collective effort of our group and active guidance made it possible for us to complete.

First of all we would like to thank **Prof. Dr. G. A. Patil (H.O.D., Department of Computer Science)** delineating us with project work.

We would also like to thank him for the support and interest that he has shown in bringing out this project and our course co-ordinator **Prof. Mrs. D. R. Patil** and **Prof. Miss. A. R. Nagaonkar** for their guidance and cooperation. We would also like to express our most humble and deepest gratitude to **Prof. Mrs. D. R. Patil** for providing us with the right guidance at the time of need it was for her presence and active guidance that we were able to complete the project work.

We would like to thank all our friend their help, ideas, criticism, and also their encouragements for preparation of this project work. Any further ideas and constructive criticisms on our shall be highly welcomed.

Date:

Place: Kolhapur

Name **Sign.**

1. Kumbhar Apurv Avinash.
2. Raut Dnyaneshear Mahadeo.
3. Sawant Ruturaj Ravindra.
4. Sayyad Rajahmed Bashir.

INDEX

1. Introduction

The string validation is a technique in which we check that input string meets the certain criteria. If the entered string meets the given set of criteria then the input string is valid otherwise the given input string is invalid.

In this project, we are validating a string consisting of parentheses. We are validating the input string against the set of predefined valid sequence of parentheses. The valid sequences of parentheses in this project is “(())” and “(()())”. The invalid sequence of parentheses in this project is “()()” and “(())()”. We are validating string against the valid and invalid sequences. We will erase valid sequences of parentheses and keep the invalid sequence as it is. Therefor if the given string is valid, then it becomes empty otherwise invalid.

If we manually find the string is valid or not it will take more time and there also be some human errors. A system is created to validate given string and output is calculated within fraction of time with great efficiency.

2. Problem Statement

To create a system to identify valid sequence of parentheses from given string and erasing found valid sequence until it reduces to empty string or non-empty string.

3. Objectives

1. To develop a system for user, that will choose the number of string that want to enter and taking that strings as input.

2. To develop user friendly application for the user.

3. To find the valid and invalid strings.

4. Flow Chart

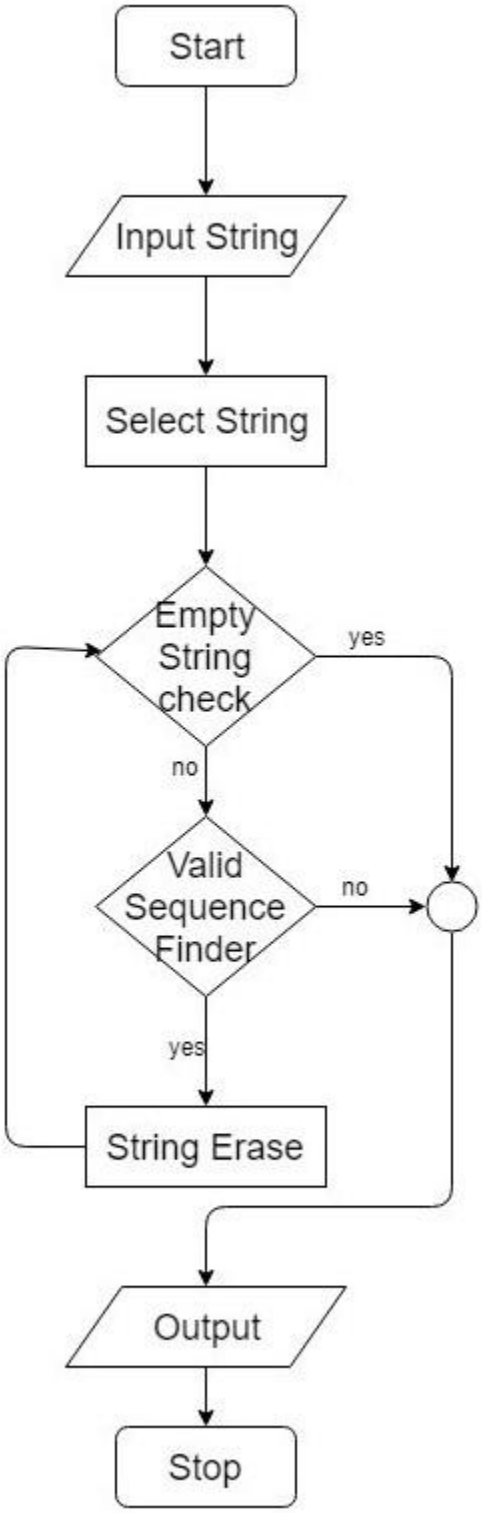

Fig. Flow chart

Explanation

When the program starts, the welcome screen is displayed, then user is asked for number of strings and strings. Then strings are selected one by one, further the string is checked, if input string is empty then output is displayed as “ATM Won”. If the selected string is non-empty the string is checked for predefined valid sequences. If valid sequence is found then that predefined valid sequence is erased. This process is repeated until string gets empty or no valid sequence is found. If the resultant string is empty, then “USER Won” is displayed. If the resultant string is non-empty then “ATM Won” is displayed. This process is repeated for every string. After processing every string, the program is terminated.

5. Algorithm

1. START
2. Take input
    a) Name of user
    b) Number of strings (T)
    c) Repeat c for i=0 and i<T
    d) Input string
3. Repeat 4 to 5 for i=0 and i<T
4. Call count function to find count of opening parentheses and closing parentheses.
   If count of opening parentheses != count of closing parentheses, then:
        Print user won.
   Else
        a) Repeat b while string contain valid sequence.
        b) Find and Erase valid sequence.
5. Display Result
   If string is empty, then:
        Print ATM won.
   Else
        Print user won.
6. STOP

6. Functions and Constructs

1. Header Files

1. iostream

This header file is used for basic input and output operations.

2. string

This header file is used because it has some member function which are required to perform operations on string.

3. cstdio

This header file is used for getch() function.

4. graphics.h

This header file is included for graphics functionality.

2. Built-in Functions

1. string::find()

size_t find (const string& str, size_t pos = 0) const noexcept;

The find function searches the range start to end for value specified by val. It return the position of the first character of the first match.

2. string::erase()

iterator erase (const_iterator first, const_iterator last);

Iterators specifying a range within the string to be removed: (first,last). i.e., the range includes all the characters between first and last, including the character pointed by first but not the one pointed by last.

3. string::length()

size_t length() const noexcept;

Return the length of string.

4. string::empty()

bool empty() const noexcept;

Returns true if the string length is 0, false otherwise

5. string::begin()

iterator begin() noexcept;

Returns an iterator pointing to the first character of the string.

6. initwindow()

int initwindow(int width, int height, const char* title="Windows BGI");

The function initializes the graphics system by opening a graphics window of the specified size.

7. readimagefile()

void readimagefile(const char* title=NULL,int left=0, int right=0, int right=INT_MAX, int bottom=INT_MAX);

The function reads a BMP, GIF, JPG, ICO, EMF or WMF image file and displays it in part of the current active window.

8. closegraph()

void closegraph();

closegraph function closes the graphics mode, deallocates all memory allocated by graphics system and restores the screen to the mode it was in before you called initgraph.

9. getch()

int getch();

It reads a single character from keyboard.

3. User Defined Functions

1. count()

int count(string)

It takes string as input and return 1 if number of opening brackets and closing brackets is equal. Otherwise it returns 0.

4. Constraints

1. $1 <= T <= 10000$
2. $2 <= \text{length of } S <= 10000$
3. The sum of all lengths of *S* will be not more than 1000000
4. *S* will be a valid bracket sequence

7. Input and Output

1. Input

1. The first line of the input contains an integer *T* denoting the number of test cases. The description of *T* test cases follows.

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

3. **Sample Input 1**

Enter your Name : Ruturaj

Ruturaj enter number of strings you want to enter : 2

Enter Strings

Enter String no 1:(())

Enter String no 2:(())()

4. **Sample Input 2**

Enter your Name : Ruturaj

Ruturaj enter number of strings you want to enter : 3

Enter Strings

Enter String no 1:((())(()))

Enter String no 2:(()())(())

Enter String no 3:()()

2. Output

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

2. **Sample Output for Sample Input 1**

Results

String (()) is Valid So ATM Won

String (())() is invalid So Ruturaj Won

3. **Sample Output for Sample Input 2**

Results

String ((())(())) is invalid So Ruturaj Won

String (()())(()) is Valid So ATM Won

String ()() is invalid So Ruturaj Won

8. Results

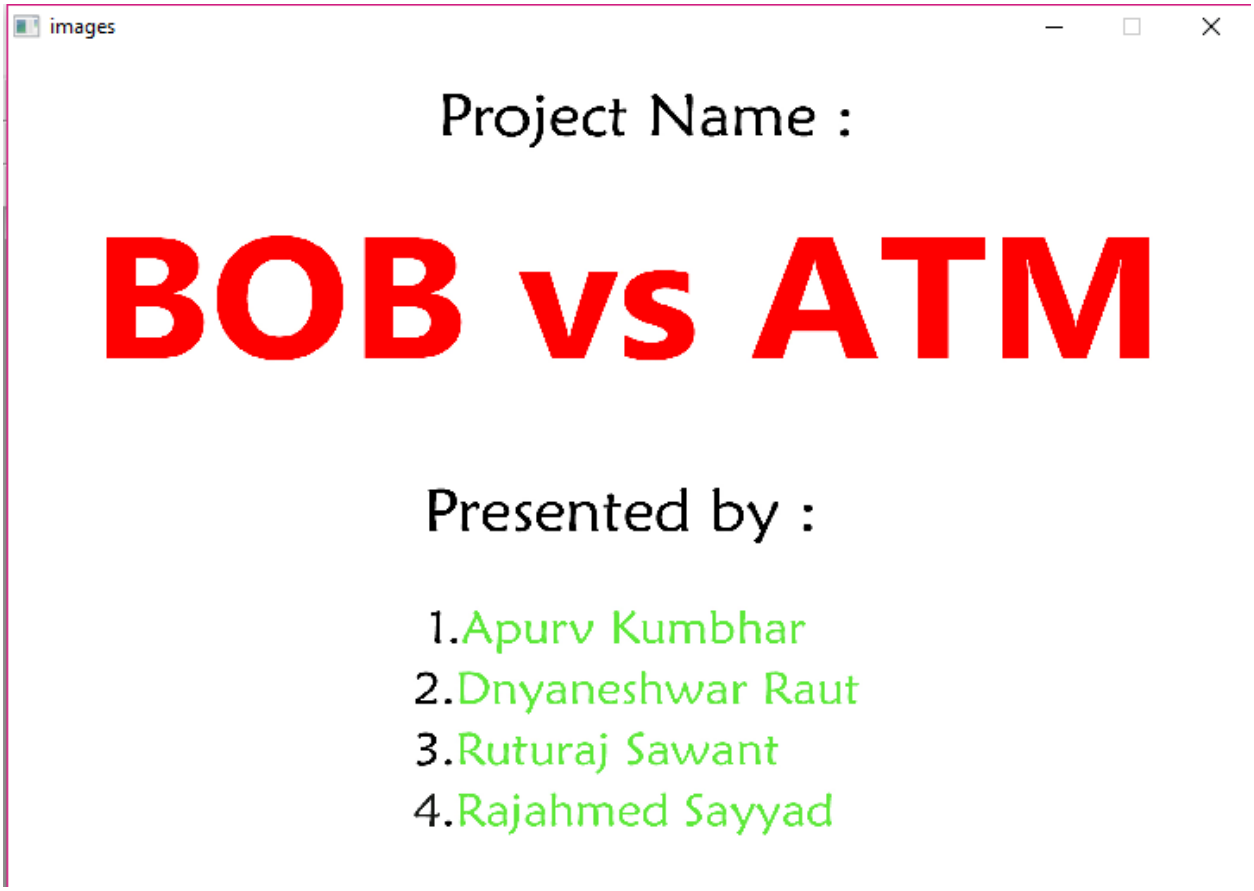

Fig. (1)

Screenshot in fig (1) shows picture when program is started, which is the welcome screen created using graphics.

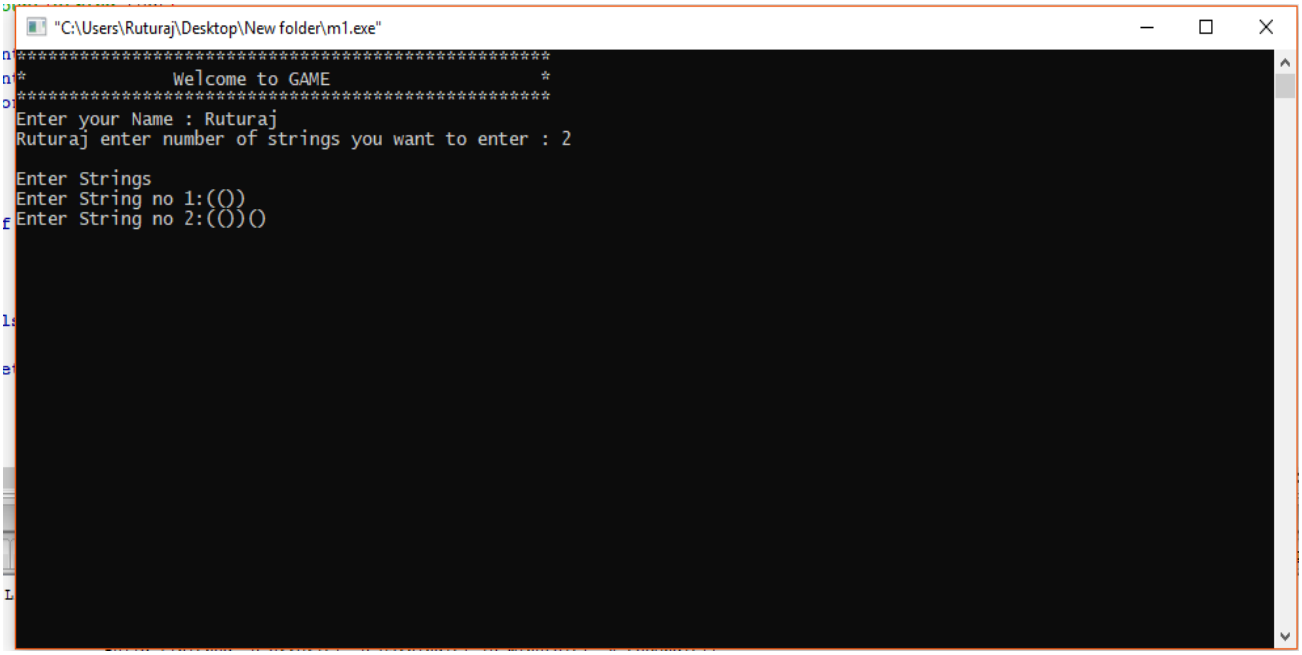

Fig. (2)

Screenshot in fig (2) shows picture after welcome screen displayed, which instructs the user and ask for name, number of strings and strings.

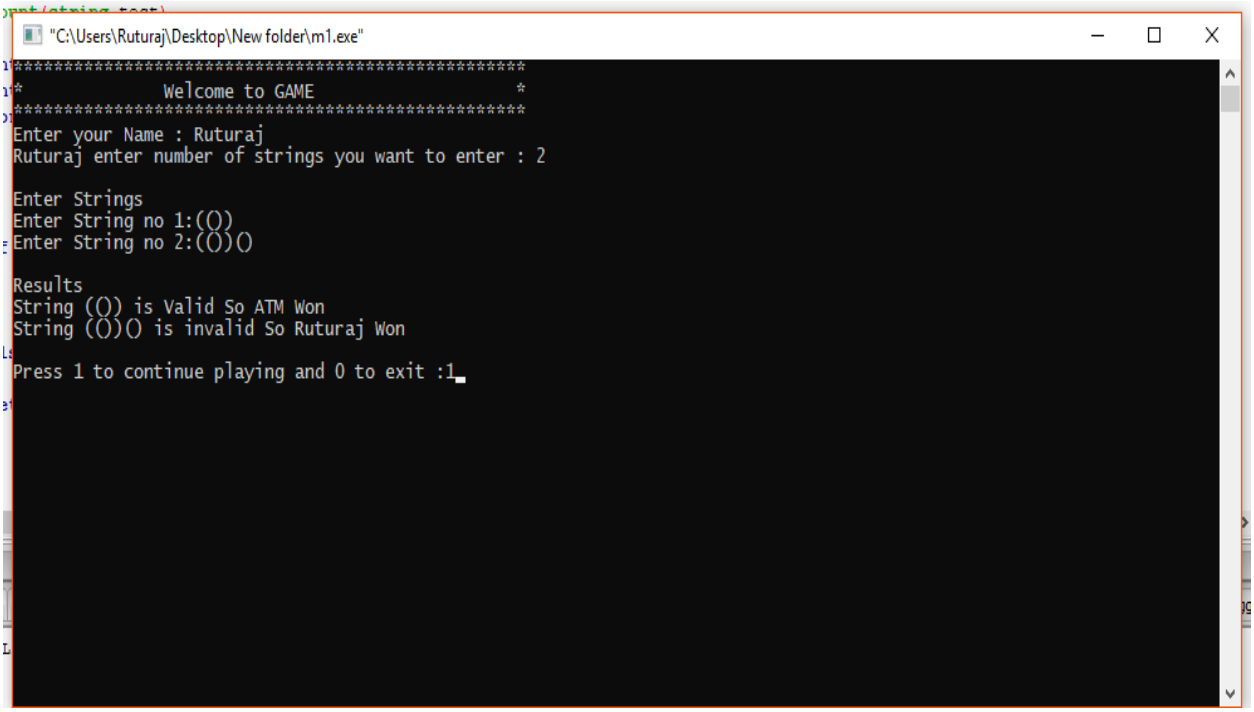

Fig. (3)

Screenshot in fig (3) shows picture when input is taken, which displays the output for given users input and further asks user for continue to play or exit the game.

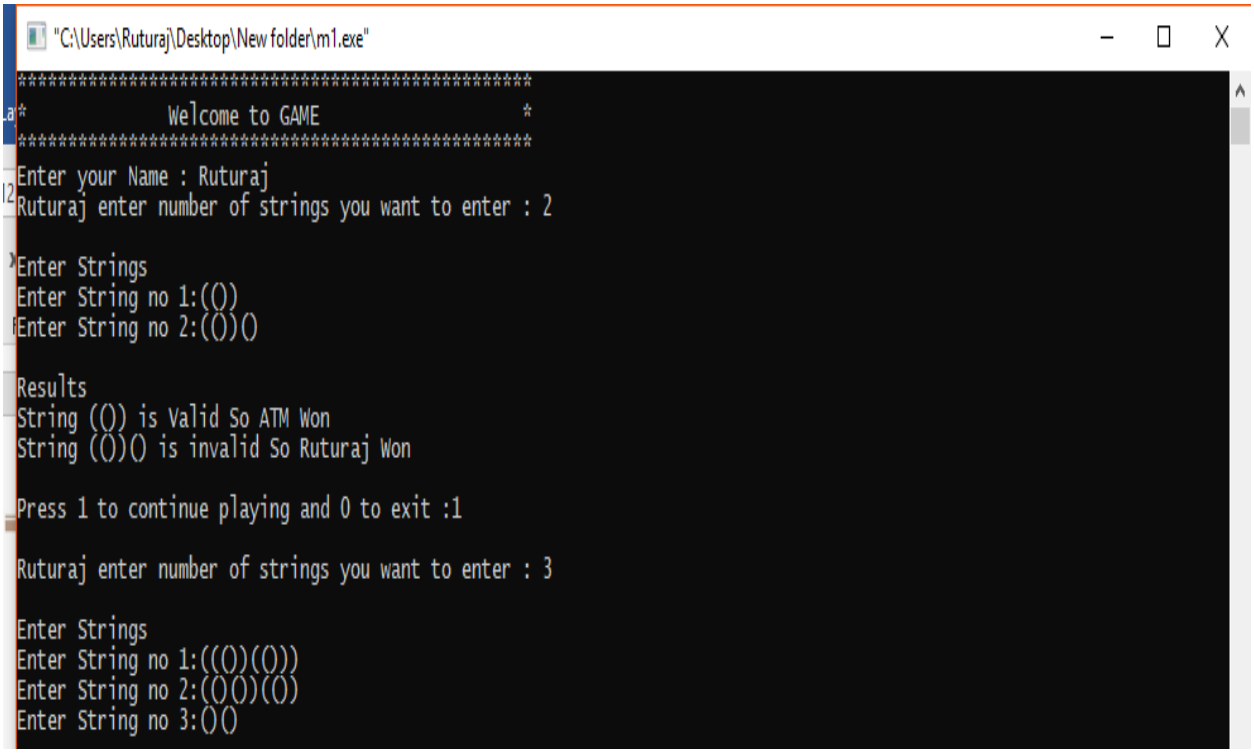

Fig. (4)

Screenshot in fig (4) shows picture, which displays the takes input from user for second time.

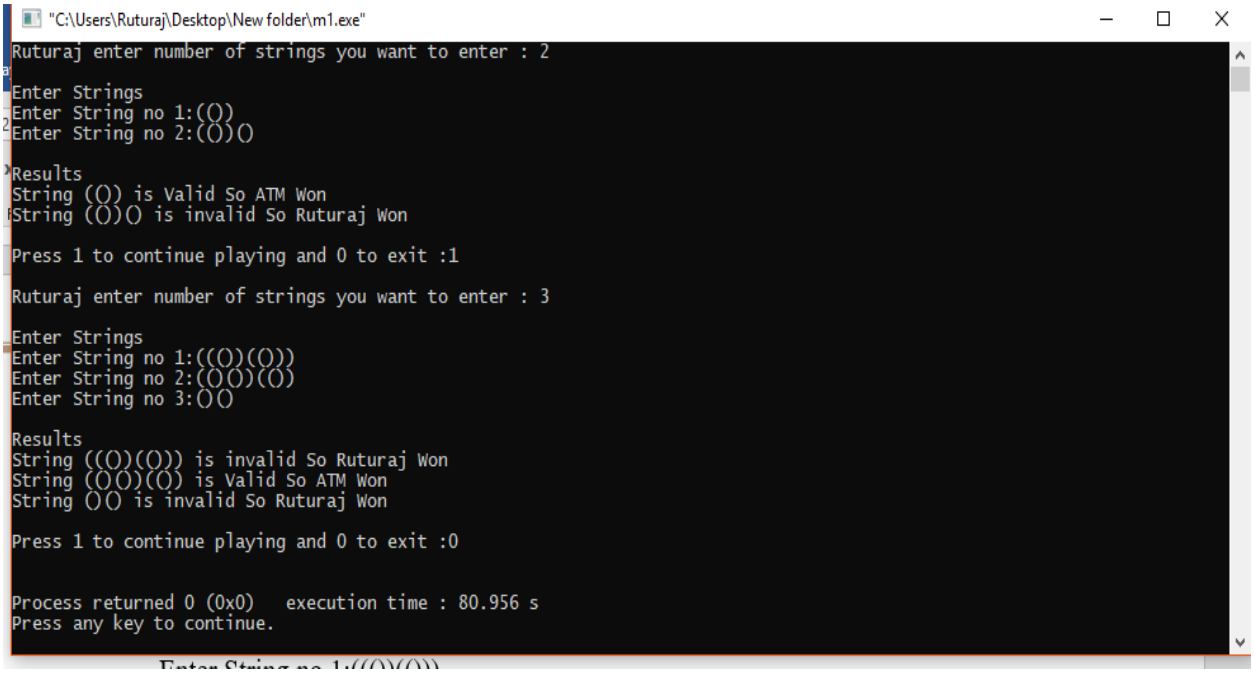

Fig. (5)

Screenshot in fig (5) shows picture, which displays the output for given users input and further asks user for continue to play or exit the game and user exits the game.

9. Software Requirement Specification

1. Hardware Requirements

1. RAM : 1 GB
2. Processor : Pentium 4 onwards

2. Software Requirements

1. Language : C++
2. Operating system : Windows 10
3. Compiler : GNU G++

10. Conclusion

This Mini Project is developed to determine whether given bracket sequence is valid or not using specified valid bracket sequence definitions. It checks validity of string using predefined rules and with help of some functions. It will validate the given set of input string in fraction of second without any errors.

11. References

1. Books

1. C++: The Complete Reference by Herbert Schildt

2. Websites

1. https://icpc.baylor.edu/
2. http://en.cppreference.com/w/
3. https://stackoverflow.com/questions/30472509/stdstringerase-erases-everything-after-1st-char-found-with-stdstringfi
4. https://www.geeksforgeeks.org/c-plus-plus/
5. http://www.informit.com/articles/article.aspx?p=328647