**Bharatiya Vidya Bhavan’s** 

**SARDAR PATEL INSTITUTE OF TECHNOLOGY**

(Autonomous Institute Affiliated to University of Mumbai)

Munshi Nagar, Andheri (W), Mumbai – 400 058.

Department of AI-ML

| **Experiment** | 0 |
| --- | --- |
| **Aim** | Implement the given problem statement using a stack. |
| **Objective** | Given as input a sequence of opening and closing brackets, check whether it is balanced or not.  In particular, check whether the first bracket is an opening bracket and for every opening bracket,  there should be a matching closing bracket.  For example, ((())) is balanced but )()( is not. |
| **Name** | Balla Mahadev Shrikrishna |
| **UCID** | 2023300010 |
| **Class** | A |
| **Batch** | A |
| **Date of Submission** | 09/08/24 |

| **Explanation of the technique used** | For each closing bracket, an opening bracket should exist in the stack for the i/p to be balanced. After iterating over the whole array, it is checked if there exists any unclosed opening bracket. If so, the i/p is declared unbalanced. |
| --- | --- |
| **Program(Code)** | #include <bits/stdc++.h>  using namespace std;  class Stackk{  int topp;  int n=20;  char arr[20];  public:  Stackk(){  topp=-1;  }  void push(char n){  if(topp>=19){  cout<<"Overflow\n";  return;  }  topp++;  arr[topp]=n;  }  void pop(){  if(topp<0){  cout<<"Underflow\n";  return;  }  topp--;  return;  }  char top(){  if(topp<0){  cout<<"underflow"<<endl;  return -1;  }  return arr[topp];  }  int size(){  return topp+1;  }  };  int main(){  string s;  cin>>s;  Stackk stk;  for(auto c: s){  if(c=='('){  stk.push('(');  }  else if(c==')'){  if(stk.top()!='('){  cout<<"unbalanced"<<endl;  return -1;  }  else stk.pop();  }  }  if(stk.size()!=0){  cout<<"unbalanced"<<endl;  return -1;  }  cout<<"balanced"<<endl;  return 0;  } |
| **Output** |  |
| **Conclusion** | Implemented the logic to check if a sequence of brackets is balanced or not using a stack. |