**PROGRAM – 7**

**AIM :** WAP to implement Recursive descent parser using predefined grammar .

**PROGRAM :**

#include <bits/stdc++.h>

using namespace std ;

int F(string str , int i) {

if(str[i] == 'a') {

if(i == str . length() - 1) {

return 1 ;

} else {

if(F(str , i + 1) == 1)

return 1 ;

else

return 0 ;

}

} else {

return 0 ;

}

}

int T(string str , int i) {

if(str[i] == 'b') {

if(i == str . length() - 1) {

return 1 ;

} else {

if(T(str , i + 1) == 1)

return 1 ;

else

return 0 ;

}

} else {

return 0 ;

}

}

void E(string str) {

if(str[0] == '+') {

int val = F(str , 1) ;

if(val == 1)

cout << "Valid string ...\n" ;

else {

val = T(str , 1) ;

if(val == 1)

cout << "Valid string ...\n" ;

else

cout << "Invalid string ...\n" ;

}

} else {

cout << "Invalid string ...\n" ;

}

}

int main() {

/\*

Predefined Grammar -

E->+F|+T

F->aF|$

T->bT|$

\*/

cout << "Enter any string to parse -\n" ;

string str ;

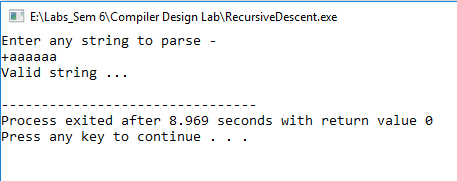
cin >> str ;

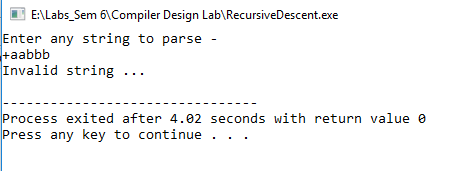
E(str) ;

return 0 ;

}

**Output :**

****

****