**EXP 7 - SHIFT-REDUCE Parsing**

**Name: -** Sai Mohit Ambekar

**Reg No: -** RA1911031010137

**Class: -** CSE-IT (L2 Section)

**Aim:**

To write a program for the implementation of shift reduce parsing.

**Problem Statement:**

Consider the following grammar:

C -> 0C1 | 01

Parse the following input string using shift-reduce parser: 010101

**Code:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int z = 0, i = 0, j = 0, c = 0;

char a[16], ac[20], stk[15], act[10];

void check()

{

strcpy(ac, "REDUCE TO E -> ");

for (z = 0; z < c; z++)

{

if (stk[z] == '4')

{

printf("%s4", ac);

stk[z] = 'E';

stk[z + 1] = '\0';

printf("\n$%s\t%s$\t", stk, a);

}

}

for (z = 0; z < c - 2; z++)

{

if (stk[z] == '2' && stk[z + 1] == 'E' &&

stk[z + 2] == '2')

{

printf("%s2E2", ac);

stk[z] = 'E';

stk[z + 1] = '\0';

stk[z + 2] = '\0';

printf("\n$%s\t%s$\t", stk, a);

i = i - 2;

}

}

for (z = 0; z < c - 2; z++)

{

if (stk[z] == '3' && stk[z + 1] == 'E' &&

stk[z + 2] == '3')

{

printf("%s3E3", ac);

stk[z] = 'E';

stk[z + 1] = '\0';

stk[z + 1] = '\0';

printf("\n$%s\t%s$\t", stk, a);

i = i - 2;

}

}

return;

}

int main()

{

printf("Grammar is:-\nC->0C1|01\n ");

strcpy(a, "010101");

c = strlen(a);

strcpy(act, "SHIFT");

printf("\nstack \t input \t action");

printf("\n$\t%s$\t", a);

for (i = 0; j < c; i++, j++)

{

printf("%s", act);

stk[i] = a[j];

stk[i + 1] = '\0';

a[j] = ' ';

printf("\n$%s\t%s$\t", stk, a);

check();

}

check();

if (stk[0] == 'E' && stk[1] == '\0')

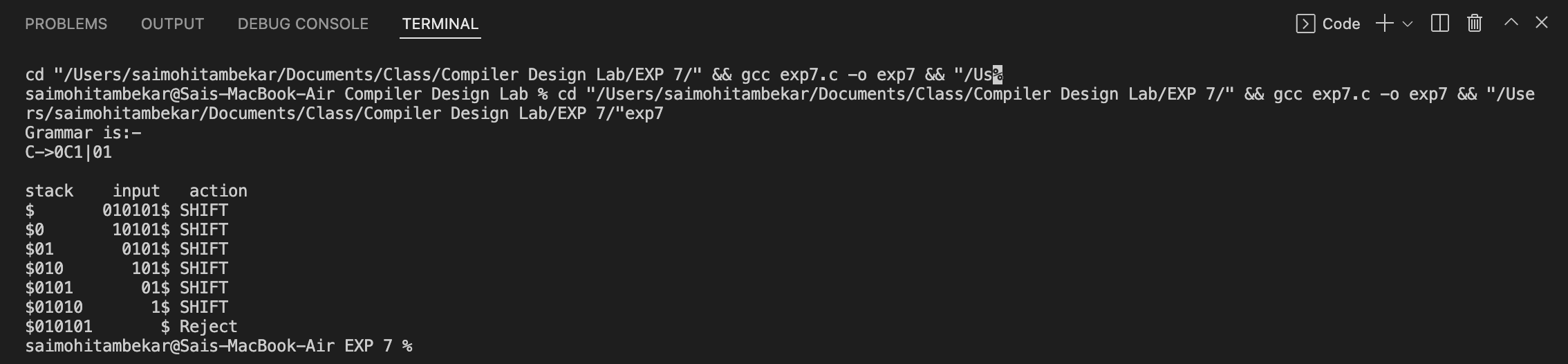
printf("Accept\n");

else

printf("Reject\n");

}

**Output:**

****

**Result:**

Shift reduce parsing has been implemented successfully.