Experiment XI

**Aim**: Write a program to implement a shift-reduce parser for a given language.

Algorithm

1. Start
2. Copy the grammar into the array.
3. Enter the input string.
4. Parse the input string from left to right.
5. If the current element/group of elements is present on the right side of any rule, reduce it by replacing it with the left-hand side symbol..
6. After parsing the entire string, only the start symbol remains in the stack, string is accepted, else rejected.
7. Stop

Output

GRAMMAR is -

E->2E2

E->3E3

E->4

Enter input string: 23432

stack input action

$ 23432$ SHIFT

$2 3432$ SHIFT

$23 432$ SHIFT

$234 32$ REDUCE TO E -> 4

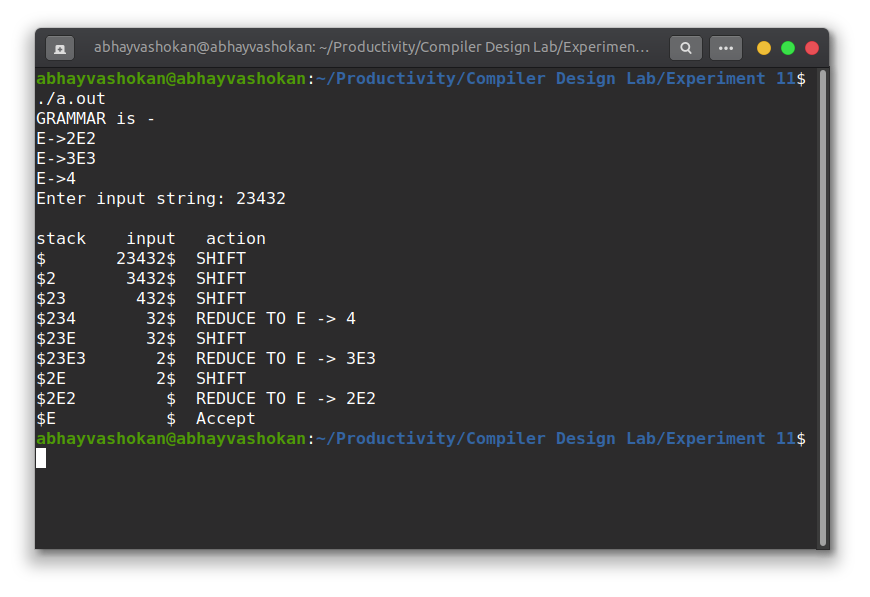
$23E 32$ SHIFT

$23E3 2$ REDUCE TO E -> 3E3

$2E 2$ SHIFT

$2E2 $ REDUCE TO E -> 2E2

$E $ Accept

Screenshot

Readme

1. Compile and run the C program using the command

**gcc 2Abhay-P11.c && ./a.out**

2. Enter the input string

3. The program returns whether the string is accepted or rejected.

**Result**: Successfully implemented shift-reduce parser in C.