Instructions:

1. Please write the program in C language.
2. Use the program name as "$A\langle assignmentNo\rangle\_\langle rollNo\rangle$.c". For example A2_15XXXXXX.c for the program of assignment-2 submitted by 15XXXXXX. If a group is submitting the assignment, then the program name should be "$A\langle assignmentNo\rangle\_\langle rollNo1\rangle\_\langle rollNo2\rangle$.c".
3. Upload the latest program in your google drive and share it to "**assignments.ds@gmail.com**". The procedure is as follows.
   - Right click the uploaded file and click "share".
   - Select "Advanced" button at the bottom right corner.
   - Give "assignments.ds@gmail.com" as "view access".
   - Untick the Checkbox for "Notify People" in order to avoid email notification.
   - Click "Ok" and then "Done".
4. The code can be deleted and submitted again with the same name, any time before 11:59 pm of the last date of submission. The latest file will be used for evaluation. Any file submitted after the deadline will not be evaluated.
5. As the file is only shared as view access to the evaluator, students are not supposed to delete the file in their own drive after the deadline, which may cause deletion of the shared file in evaluator's drive. Hence keep the files in the drive unchanged till the end of the semester. This is required to maintain the ownership of the file.
6. An Input test case file is also attached to the assignment notification email. This input can be given to the program using the following syntax.
   ./a.out < $\langle inputFileName\rangle.txt$
   The code snippet for reading this input is given below.
7. Please refrain from copying the code.
8. Last date of submission: **23th August 2016 11:59 pm.**

Code Snippet to read the input test case

You may use the following code to read the input file.

```
#include<stdio.h>

void main(){
    int n,i;
    char c[20][100];
    scanf("%d\n",&n);

    printf("\nThe number of expressions, n = %d",n);

    for(i=0;i<n;i++){
        gets(c[i]);
```

```
        printf("\nc[%d]=%s",i,c[i]);
    }
}
```

Question

Implement a **stack** data structure and its associated functions. Evaluate the expressions given in the input test case using stack. The expressions can contain operators from the set $\{+, -, *, /, \wedge\}$. You must incorporte the precedence and associativity rules for these operators.

The ouput should contain the answer for each expression (on a seperate line). The sample input and output are given below.

Sample Input

2
1 + 2 * 4 − 3^2
2^2 − 1

Output

0
3

The End.