Approach

***COL106: Assignment 2***

***2019CS10722***

***SACHIN***

1. Postfix expression is taken as input in one line in the form of string and stored in memory.
2. Maximum 1024 characters can be taken as input. If more given, computation will be done on first 1024 characters and warning “ *Your input is too big, answer might be frivolous.”* will be thrown.
3. Address of input string is loaded in $t0 and character at that address in $t1.
4. Equality check is done on $t1 and accordingly pop and push is done in stack.
5. If $t1 is digit between 0-9 then it is pushed in stack, and head pointer is changed accordingly.
6. If $t1 is operator +,-,\* then appropriate is computation is done on last two numbers of stack, they are popped and the computation is pushed back.
7. Once $t1 becomes empty or “\n” then computation stops.
8. If any other value of $t1 is encountered then error “*INVALID-INPUT: Input is not in postfix”* will be thrown and computation stops.
9. Now that computation is done, our answer will be at head pointer of stack (if input was correct postfix expression) otherwise number stored at head is the partial computation of postfix subset of input. My code will print this partial computation at this step with a warning “*Your input is not complete postfix, ans above is partial answer.”* Otherwise correct answer is printed.

Wrong Input

1. The case when input containing any character other than {0-9,+,-,\*} is stated in approach section above.
2. The case when operators is not equal to (number of digits -1)

Test Cases

1. Once my code was done, I first tested it with the testcase given in question. Code gave correct answer in that, then I used induction to prove the correctness of my code.
2. Proof by induction on number of digits in