NAME: AGNIV PRAMANICK **SECTION:** A **USN:** 1NT21IS017 **DATE:** 29/12/22

Q. Write a program for evaluation of infix to prefix.

THEORY

Prefix expression:

Another way to describe anything is with a prefix notation, which does not require knowledge about precedence or associativity but does when used with an infix notation. It is also known as **polish notation**. In prefix notation, an operator comes before the operands.

The syntax of prefix notation is given below:
<operator> <operand> <operand>

Example:

a b + c d + *

Evaluate the given PREFIX expression:

+, -, *, 2, 2, /, 16, 8, 5
Let's reverse the given prefix expression.
Expression: 5, 8, 16, /, 2, 2, *, -, +

Symbol Scanned	Stack
5	5
8	5, 8
16	5, 8, 16
/	5, 2
2	5, 2, 2
2	5, 2, 2, 2
*	5, 2, 4
-	5, 2
+	7

ANSWER IS 7.

ALGORITHM

Step 1: Initialize a pointer 'S' pointing to the end of the expression.

Step 2: If the symbol pointed by 'S' is an operand then push it into the stack.

Step 3: If the symbol pointed by 'S' is an operator then pop two operands from the stack. Perform the operation on these two operands and stores the result into the stack.

Step 4: Decrement the pointer 'S' by 1 and move to step 2 as long as the symbols left in the expression.

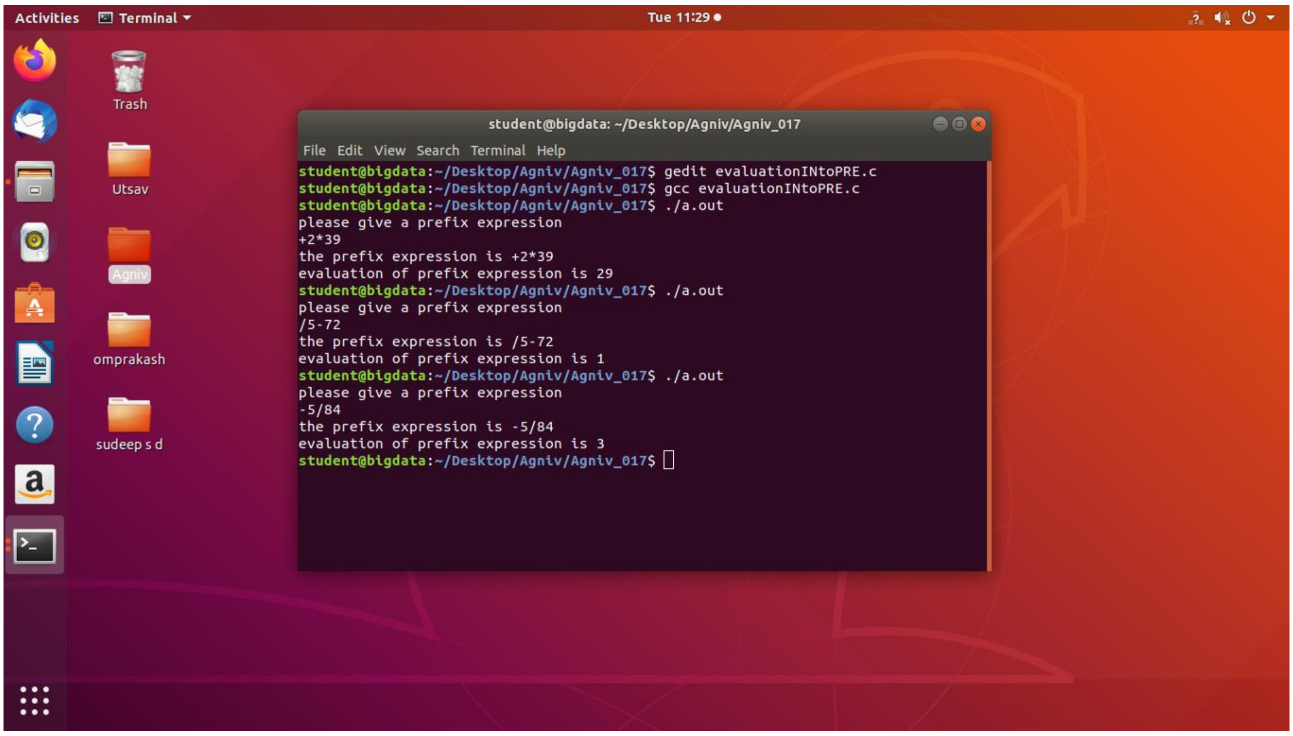
Step 5: The final result is stored at the top of the stack and return it.

Step 6: End

CODE

```
//to evaluate the prefix expression after converting it from infix
#include<stdio.h>
#include<string.h>     //library function inserted
#include<stdlib.h>

//char stack
char stack[50];  //create a stack array of 50 size
int top=-1;        // value of -1 is given to top

//---------------------------------------------------------------------------------------------------
// creating a user defined function to push elements
void push(int item)
{
 stack[++top]=item; // incrementing top, equating the array to item
}

//---------------------------------------------------------------------------------------------------
// creating a user defined function to pop elements
int pop()
{
 return stack[top--];
}

int isoperand(char c)
{
return isdigit(c);
}

//---------------------------------------------------------------------------------------------------
//evaluation of PREFIX

// creating a user defined function to evaluate elements
int prefixeval(char exp[])
 {
  int a,b;      //declaring two operands
  int i;      //giving i a value of 0

  for(i=strlen(exp)-1; i>=0; i--)
  {
    if(isoperand(exp[i]))
    {
      push(exp[i]-'0');
     }
     else
     {
     a=stack[top];
     pop();
     b=stack[top];
     pop();


   switch(exp[i]) // created a switch case
   {
    case'+': push(a+b); //addition
            break;
    case'-': push(a-b); //subtraction
            break;
    case'*': push(a*b); //multiplication
            break;
    case'/': push(a/b); //division
            break;
    case'^': push(a^b); //power
            break;

    }
   }

  }
  return stack[top];    // return the stack array value
  }

 //-------------------------------------------------------------------------------------
// main function started here
void main()
{
char prefix[30];
printf("please give a prefix expression \n");  //giving prefix the expression
scanf("%s",prefix);
printf("the prefix expression is %s \n",prefix); //printing the prefix expression.
printf("evaluation of prefix expression is %d \n",prefixeval(prefix));  //printing the evaluated answer
}

//code is ended
```

OUTPUT


Activities
Terminal
Tue 11:29
Trash
Utsav
Agniv
omprakash
sudeep s d
student@bigdata: ~/Desktop/Agniv/Agniv_017
File Edit View Search Terminal Help

```
student@bigdata:~/Desktop/Agniv/Agniv_017$ gedit evaluationINtoPRE.c
student@bigdata:~/Desktop/Agniv/Agniv_017$ gcc evaluationINtoPRE.c
student@bigdata:~/Desktop/Agniv/Agniv_017$ ./a.out
please give a prefix expression
+2*39
the prefix expression is +2*39
evaluation of prefix expression is 29
student@bigdata:~/Desktop/Agniv/Agniv_017$ ./a.out
please give a prefix expression
/5-72
the prefix expression is /5-72
evaluation of prefix expression is 1
student@bigdata:~/Desktop/Agniv/Agniv_017$ ./a.out
please give a prefix expression
-5/84
the prefix expression is -5/84
evaluation of prefix expression is 3
student@bigdata:~/Desktop/Agniv/Agniv_017$
```