APPLICATION OF STACK

AIM:

The code aims to evaluating arithmetic expressions efficiently handles operator precedence And parantheses by using stack.

ALGORITHM:

1. Start
2. Create an empty stack to hold operands.
3. Initialize a variable top to -1 which represents the top of the stack
4. Read the input from user
5. Iterate through each character in the expression
6. If the character is a digit, convert the character to its integer value and push the integer into stack.
7. if the character is an operator, pop the top two operands from the stack and perform the corresponding operation. 8. Get the result and display it
9. End

PROGRAM:

```
/*Evaluation of postfix expression*/
#include<stdio.h>
#include<ctype.h>


int stack[1000],top=-1;


void push(char x){
   top++;
   stack[top]=x;
}
int pop(){
   int x=stack[top];
   top--;
   return x;
}


int main(){
   char ep[1000];
   printf("Enter the postfix expression: ");
   scanf("%s",ep);
   for(int i=0;ep[i]!='\0';i++){
      if(isdigit(ep[i])){
         push((ep[i]-'0'));
```

```
        }
        else {
            int a = pop();
            int b = pop(),c;
            switch(ep[i]){
                case '+':
                    c=a+b;
                    push(c);
                    break;
                case '-':
                    c=a-b;
                    push(c);
                    break;
                case '*':
                    c=a*b;
                    push(c);
                    break;
                case '/':
                    c=a/b;
                    push(c);
                    break;

            }
        }
    }
    printf("The answer is %d".,pop());
    return 0;

}
```

OUTPUT:

```
Enter the postfix expression:  12+
The answer is 3.
```

RESULT:

Thus the program has been successfully executed.