Program 2

Write A Program to convert a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide)

```
Code:
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#define MAX 5
char stack[MAX];
int top = -1;
void push(char c) {
  if (top \le MAX - 1) {
    stack[++top] = c;
}
char pop() {
  if (top >= 0) {
    return stack[top--];
```

```
}
  }
char peek() {
  if (top >= 0) {
    return stack[top];
  return '\0';
}
int precedence(char c) {
  switch (c) {
    case '+': return 1;
    case '-': return 1;
     case '*': return 2;
     case '/': return 2;
     case '^': return 3;
     default: return 0;
int isOperator(char c) {
```

```
\text{return } c == \text{'+'} \parallel c == \text{'-'} \parallel c == \text{'*'} \parallel c == \text{'/'} \parallel c == \text{'-'};
}
void infixToPostfix(const char *infix, char *postfix) {
  int i = 0, j = 0;
   while (infix[i]) {
      if (isalnum(infix[i])) {
         postfix[j++] = infix[i];
      } else if (infix[i] == '(') {
         push(infix[i]);
      } else if (infix[i] == ')') {
         while (top != -1 && peek() != '(') {
            postfix[j++] = pop();
         pop();
      } else if (isOperator(infix[i])) {
         while (top != -1 && precedence(peek()) >= precedence(infix[i])) {
            postfix[j++] = pop();
         }
         push(infix[i]);
     i++;
   }
```

```
while (top != -1) {
     postfix[j++] = pop();
  }
  postfix[j] = '\0';
}
int main() {
  char infix[MAX], postfix[MAX];
  printf("Enter an infix expression: ");
  scanf("%s", infix);
  infixToPostfix(infix, postfix);
  printf("Postfix expression: %s\n", postfix);
  return 0;
}
```

```
Enter an infix expression: abcd^e-fgh*+^*+i-
Postfix expression: abcde^fgh*-^*+i+-
```

WAP to convert a given valid parantheoized infix orithmetic	inteleckledet =-1;
require to coulix exercision. The expression tomics of small	(1 Create Mistack)
character operands and the binary operations of (plus), Trainers), Trainers	
and / (divide)	Pants I Memory allocation for the
	rewry;
# include (4dio m)	
# include < Andlibin)	for link , 0; i < lenj i++)
# include < ftring h)	10 Feb.
int o rec (chare)	: 11(00 to' 22 (2 "2") (C) = (A" 22 (6 = "Z")
	(C)=10' 28 CC(9'))
if ((=='A')	(1):10 11 (2):11
yelluru 3;	result (result Index ++)=c;
elec if ((=='1'))	Your Creat work of you
Veloria 2 i	Uscif (c='(3')
elect (c== (+) ((c==-)	users (c. c.s.)
yelum 1;	stack [1+ stack Index]=C;
· ·)
return -1;	eluit (1==1)
}	
that anotativity (churc)	while (stack Index >= 0 ff stack [stack Index]!
and the state of t	:()
((c:: '\)	
tchen 'L'	reme [rante Index 1+7 = stack [stack Index -]
rchurn't';	
	stact Index;
void intirtologix (comt charms)	
	lle e
int low: strike(s);	All Carlos Charges Mour Areas
char * result (char *) malloc (len *);	white (space Index >= 0 SD (prec(c) < prec(shark)
(har* stack = ((har *) mulloc (ten);	1 prec (c) == prec (glack Impack Index)) &&
int roult indeco;	associativity (c) = = (L'))

result [result] soles ++] glock [++ stock Index] = C; } while [stock Index >= 0) result [result Index ++] = shock [stock Index] result [result Index] = '\0'; print [(7-5\n'' + result); free [result]; free (stock); into moint) into moint) into moint) char exp[] = [as b + + (c^3 - c)^ (lig + h)] - (c		07
stock [i+ stock Index]: C; } while [stock Index]: C) routh [routh Index]: '\0'; print (7:s\n', result); free (stoct); int maint) char exp[]: [a+6*+(c^3-c)^(lig*h)]-c insix la Postfic(exp); yournel; Quaput:	Assisted to section to a manager to a	par Street
stock [i+ stock Index]: C; } while [stock Index]: C) routh [routh Index]: '\0'; print (7:s\n', result); free (stoct); int maint) char exp[]: [a+6*+(c^3-c)^(lig*h)]-c insix la Postfic(exp); yournel; Quaput:	Tarana.	
stock [i+ stock Index]: C; } while [stock Index]: C) routh [routh Index]: '\0'; print (7:s\n', result); free (stoct); int maint) char exp[]: [a+6*+(c^3-c)^(lig*h)]-c insix la Postfic(exp); yournel; Quaput:	regula [result Index ++]	
white (stace I when >= 0) Touth [routh Index + 7] : space (stock Index] Touth [routh Index] = (0); print (1-s \n	3	
white (stace I when >= 0) Touth [routh Index + 7] : space (stock Index] Touth [routh Index] = (0); print (1-s \n	Slack [1+ stack Index] = C	Land mark
regults [result Index +]: space (Stock Index] regults [result Index]: 100; print (1-\$ \n" result); free (result); free (start): int maint) char exp[]: \asb + (c^3 -c)^(lig + h)\-(injx In Prostic (exp); yourno; geom user Output:	200 1 March 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	A . 1 - 181 1 C
regults [result Index +]: space (Stock Index] regults [result Index]: 100; print (1-\$ \n" result); free (result); free (start): int maint) char exp[]: \asb + (c^3 -c)^(lig + h)\-(injx In Prostic (exp); yourno; geom user Output:	3 dear	11 21 4
regults [result Index +]: space (Stock Index] regults [result Index]: 100; print (1-\$ \n" result); free (result); free (start): int maint) char exp[]: \asb + (c^3 -c)^(lig + h)\-(injx In Prostic (exp); yourno; geom user Output:	white [Hack Index >: 1	p)
reput Cremet lodes]: (0); print (7:5/m", remet); free (remet); free (start); int maint) char exp[] = (a + 6 + (c^3 - c)^(lig + h)) - (infin 10 footpite(exp); yournel; Quapute:	5	
reput Cremet lodes]: (0); print (7:5/m", remet); free (remet); free (start); int maint) char exp[] = (a + 6 + (c^3 - c)^(lig + h)) - (infin 10 footpite(exp); yournel; Quapute:	routh Fronth Index +	+] : stack [Stock Index]
intermit (1.5 h., Yearne); free (start); free (start); char exp[] = as b * + (c^3 - c)^ (lig * h) - i inter 10 Postfic (exp); your no! Output:	A Comment	Challanter
intermit (1.5 h., Yearne); free (start); free (start); char exp[] = as b * + (c^3 - c)^ (lig * h) - i inter 10 Postfic (exp); your no! Output:	result Created Index] = (\0);
inte main() char exp[] = as b * (c^d - c)^(lig*h)-i intis locatic(exp); yournoi Quapute:	print (1-5 10")	(exut);
int maint) (that exp[] = \as 6 + \((c^3 - c)^\((lig + h)\)- ingin 10 foot ((up)), yether 0; Quaput: Quaput:		
int maint) (that exp[] = \as 6 + \((c^3 - c)^\((lig + h)\)- ingin 10 foot ((up)), yether 0; Quaput: Quaput:	bru (result);	
int maint) that exp[] = as 6.4 (c^3-c)^(lig+h))-i ingix la-footfillup); yether no; Quaput: Quaput:	free (start).	Cautay (A17) Separah 1919
interment) char exp[] = as b * (c^d - c)^(lig * h) - i intis lo Postfic (exp); your noi Ocapute:	and the second	}
char exp[] · as b · * (c^3 - c) ^ (lig * h) - i infix la Postfic(exp); yearn 0; Acapute: Ocapute:	The state of the s	Constant to
ingia to Post (cep); good when your when the Chapute	int main()	
ingia to Post (cep); good when your when the Chapute	11 Charles	wedget awall? I vot
ingia to Post (cep); good when your when the Chapute	char exp[] = 1	a+ 6+ + (c/d -c) ^ (fig + h) \-i
John Bon Bon Bon	mile Burne in	a (1 street)
adput	ingix Infortic (exp);	and how
adput	yehrno;	Real way
Output:	The state of the s	To the state of th
Output:		least.
1,		and the second
abid'e-fant+1++11.	adput:	and the same of
abide-fant+n++i-		
	abid'e-fant+n++i-	Norman 1 1 Jam
		1

amehic .	intolack lasex =-1;
	(L Cremit Mistack)
le * proly;	The state of the s
	Print ("Hemeny allocation failed");
1000	return:
	the state of the s
	for line = 0; i < lenj itt)
	(01 2001 1)
	Char cas (i);
	1 1 (10 to 88 (L = 2)) ((>= (A' 88 (C = 'Z')))
	(C): '0' 10 (Z;9'))
	E CONTRACTOR OF
	rout (rout Index >+)=c;
	the contract of the contract o
200	Useif (c='(3')
	S CONTRACTOR OF THE CONTRACTOR
	stack [++ stack Index]=C;
	1 In walk to the second
	eluit (c==")
0.00	A STATE OF THE PARTY OF THE PAR
23.3	while (stack Index >= 0 ld stack [stack Index]!
13.00	3()
2.00	1
	runt frault Index 1+7 - stack [stack Index -]
	7
	Stack Index;
	3,050,000,000
	elie
	1
	17 (n. 12. 7:0 (0 (//) / C. //)
	while (charalhole 7:0 & (prec(c) < nrec(sharalshor)
1	1 pres (c) == pres (storex Interest Index) 28