“C PROGRAM FOR INFIX TO POSTFIX CONVERSION USING STACK”

#include<stdio.h>

#define SIZE 50

char stack[SIZE];

int top=-1;

void push(char item)

{

if(top>=SIZE-1)

printf("\n stack overflow,push not possible \n");

else

{

top++;

stack[top]=item;

}

}

char pop()

{

char item;

item=stack[top];

top--;

return(item);

}

int is\_operator(char symbol)

{

if(symbol=='^'||symbol=='\*'||symbol=='/'||symbol=='+'||symbol=='-')

return 1;

else

return 0;

}

int precedence(char symbol)

{

if(symbol=='^')

return 3;

else if(symbol=='\*'|| symbol=='/')

return 2;

else if(symbol=='+'||symbol=='-')

return 1;

else

return 0;

}

int main()

{

char infix[SIZE],postfix[SIZE],item,temp;

int i=0,j=0;

printf("\n enter the arithmetic notation in infix notation:");

gets(infix);

while(infix[i]!='\0')

{

item=infix[i];

if(item=='(')

{

push(item);

}

else if(item>='A' && item<='Z' || item>='a' && item<='z')

{

postfix[j]=item;

j++;

}

else if(is\_operator(item)==1)

{

temp=pop();

while(is\_operator(temp)==1 && precedence(temp)>=precedence(item))

{

postfix[j]=temp;

j++;

temp=pop();

}

push(temp);

push(item);

}

else if(item==')')

{

temp=pop();

while(temp!='(')

{

postfix[j]=temp;

j++;

temp=pop();

}

}

else

{

printf("\n invalid arithmetic expression\n");

getch();

exit(0);

}

i++;

}

while(top>-1)

{

postfix[j]=pop();

j++;

}

postfix[j]='\0';

puts(postfix);

getch();

return 0;

}