//Write a program to convert an prefix expression to its corresponding infix expression.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

#include <conio.h>

char opnds[50][80],oprs[50];

int topr=-1,topd=-1;

pushd(char \*opnd)

{

strcpy(opnds[++topd],opnd);

}

char \*popd()

{

return(opnds[topd--]);

}

pushr(char opr)

{

oprs[++topr]=opr;

}

char popr()

{

return(oprs[topr--]);

}

int empty(int t)

{

if( t == 0) return(1);

return(0);

}

void main()

{

char prfx[50],ch,str[50],opnd1[50],opnd2[50],opr[2];

int i=0,k=0,opndcnt=0;

printf("Give an Expression = ");

gets(prfx);

printf(" Given Prefix Expression : %s\n",prfx);

while( (ch=prfx[i++]) != '\0')

{

if(isalnum(ch))

{

str[0]=ch; str[1]='\0';

pushd(str); opndcnt++;

if(opndcnt >= 2)

{

strcpy(opnd2,popd());

strcpy(opnd1,popd());

strcpy(str,"(");

strcat(str,opnd1);

ch=popr();

opr[0]=ch;opr[1]='\0';

strcat(str,opr);

strcat(str,opnd2);

strcat(str,")");

pushd(str);

opndcnt-=1;

}

}

else

{

pushr(ch);

if(opndcnt==1)opndcnt=0; /\* operator followed by single operand\*/

}

}

if(!empty(topd))

{

strcpy(opnd2,popd());

strcpy(opnd1,popd());

strcpy(str,"(");

strcat(str,opnd1);

ch=popr();

opr[0]=ch;opr[1]='\0';

strcat(str,opr);

strcat(str,opnd2);

strcat(str,")");

pushd(str);

}

printf(" Infix Expression: ");

puts(opnds[topd]);

getch();

}