1.Write a program to insert or delete an element from an array given a position.

**PROGRAM CODE**

#include<stdio.h>

#include<stdlib.h>

void enter(int n,int arr[])

{

printf("Enter elements: ");

for (int i = 0; i < n; i++)

scanf("%d", &arr[i]);

}

void disp(int n,int arr[])

{

printf("\nThe elements are: ");

for(int i=0;i<n;i++)

printf("%d ",arr[i]);

}

void ins(int \*n,int arr[])

{

int num,pos;

printf("Enter element to be inserted: ");

scanf("%d",&num);

printf("Enter position: ");

scanf("%d",&pos);

(\*n)++;

for(int i=(\*n)-1;i>=pos;i--)

arr[i]=arr[i-1];

arr[pos-1]=num;

disp((\*n),arr);

}

void del(int \*n,int arr[])

{

int pos;

printf("Enter position to be deleted: ");

scanf("%d",&pos);

pos-=1;

for(int i=pos;i<(\*n)-1;i++)

arr[i]=arr[i+1];

(\*n)--;

disp((\*n),arr);

}

void main()

{

char rpy;

int ch,n,arr[20];

printf("Enter number of elements: ");

scanf("%d",&n);

do

{

printf("\nMENU\n");

printf("\n1. Enter Elements\n2. Display elements\n");

printf("3. Insert an element\n4. Delete an element\n5. Exit");

printf("\nEnter your choice: ");

scanf("%d",&ch);

switch(ch)

{

case 1: enter(n,arr);

break;

case 2: disp(n,arr);

break;

case 3: ins(&n,arr);

break;

case 4: del(&n,arr);

break;

case 5: exit(0);

break;

default: printf("\nINVALID INPUT\n");

}

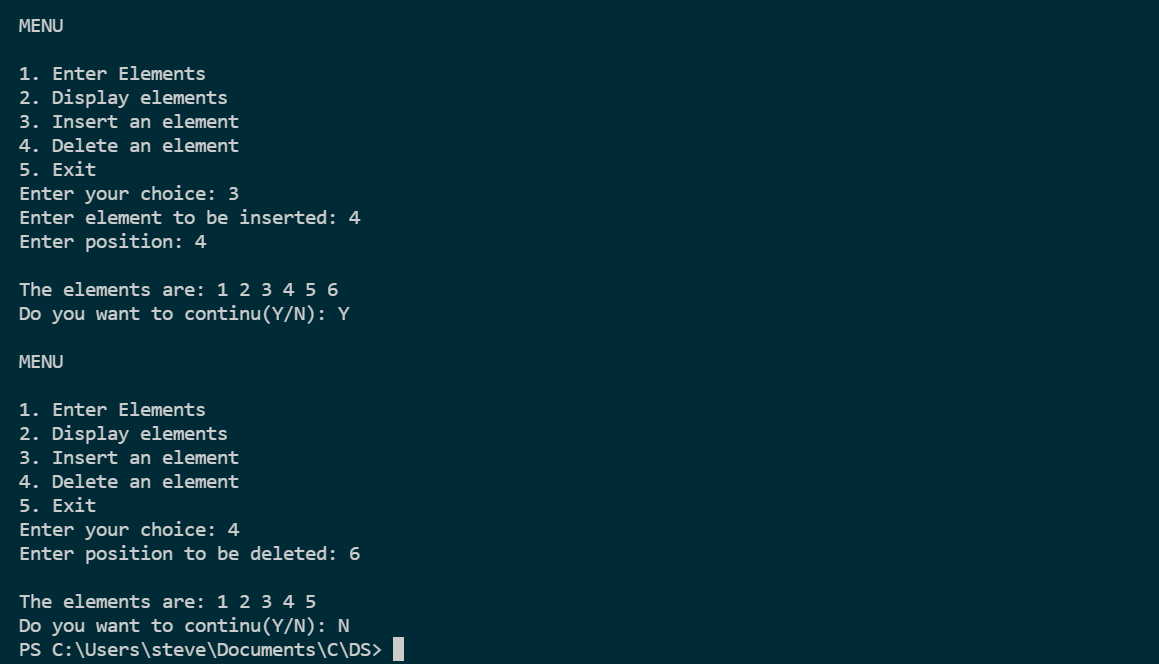
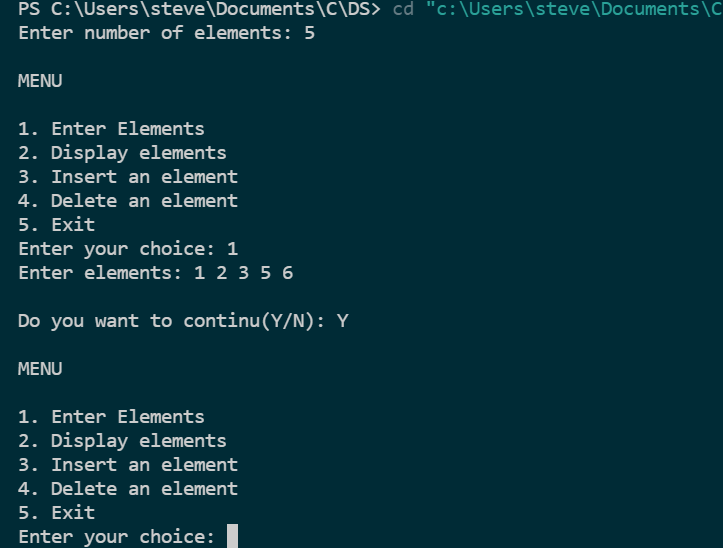
printf("\nDo you want to continu(Y/N): ");

scanf("%s",&rpy);

}while(rpy=='Y' || rpy=='y');

}

**OUTPUT**



2. Perform polynomial addition using arrays

**PROGRAM CODE**

#include<stdio.h>

#include<stdlib.h>

void main()

{

int i,poly1[15],poly2[15],poly3[15],deg1,deg2,deg3;

printf("Enter highest degree of polynomial 1: ");

scanf("%d",&deg1);

printf("Enter Coefficients\n");

for(i=0;i<=deg1;i++)

{

printf("X^%d:",i);

scanf("%d",&poly1[i]);

}

printf("\nEnter highest degree of polynomial 2: ");

scanf("%d", &deg2);

printf("Enter Coefficients\n");

for (i = 0; i <= deg2; i++)

{

printf("X^%d:",i);

scanf("%d", &poly2[i]);

}

printf("\nADDING BOTH POLYNOMIALS\n");

if(deg1>deg2)

{

for(i=0;i<=deg1;i++)

{

if (i<=deg2)

poly3[i]=poly1[i]+poly2[i];

else if( i>deg2)

poly3[i]=poly1[i];

}

deg3=deg1;

}

else if(deg1<deg2)

{

for (i = 0; i <=deg2; i++)

{

if (i <= deg1)

poly3[i] = poly1[i] + poly2[i];

else if (i > deg1)

poly3[i] = poly2[i];

}

deg3=deg2;

}

else

{ for (i = 0; i <= deg1; i++)

poly3[i] = poly1[i] + poly2[i];

deg3=deg1;

}

printf("Sum of polynomials: ");

for(i=0;i<=deg3;i++)

{

if(i==0)

printf("%d ",poly3[i]);

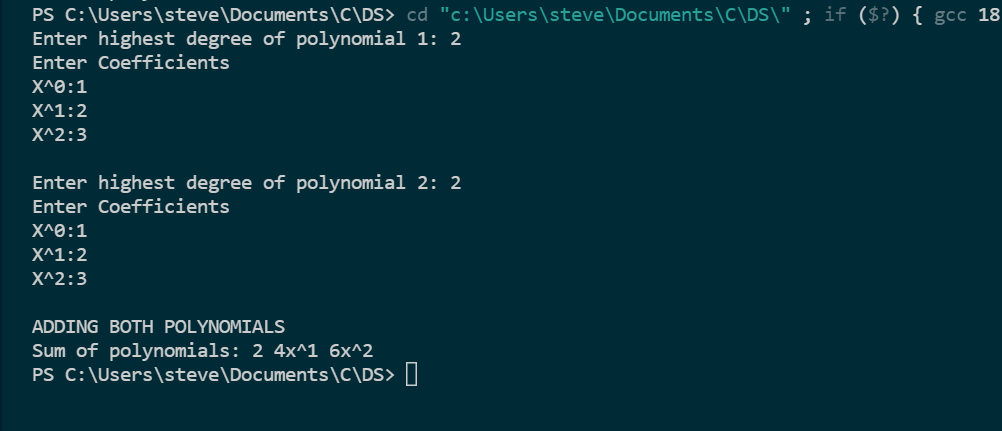
else

printf("%dx^%d ",poly3[i],i);

}

}

**OUTPUT**



3. Write a menu driven program to perform various stack operations.

**PROGRAM CODE**

#include <stdio.h>

#include <stdlib.h>

void main()

{

int stack[10],ch,i,item,top=-1,max;

printf("Enter the maximum size:");

scanf("%d", &max);

while (1)

{

printf("\nMENU");

printf("\n1.Push\n2.Pop\nn3.Display\n4.Exit\nEnter your choice:");

scanf("%d", &ch);

switch (ch)

{

case 1:

if (top>=max - 1)

{

printf("\nStack overflow");

}

else

{

printf("\nEnter the item:");

scanf("%d", &item);

top = top + 1;

stack[top] = item;

}

break;

case 2:

if (top == -1)

printf("\nstack underflow");

else

{

item = stack[top];

top = top - 1;

printf("\ndeleted item is %d\n", item);

}

break;

case 3:

if (top == -1)

printf("\nNo elements in stack.");

else

{

printf("\nElements are: ");

for (i = top; i >= 0; i--)

printf(" %d", stack[i]);

}

break;

case 4:

exit(0);

default:

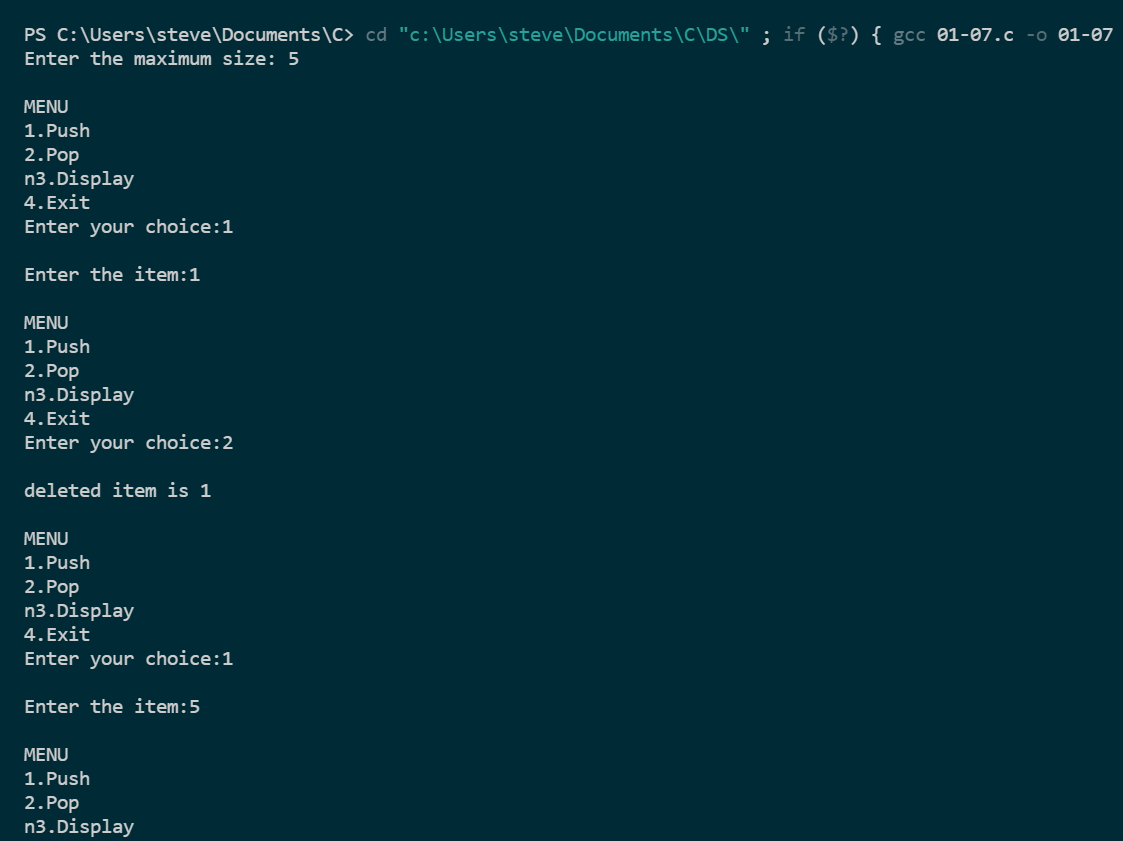
printf("\nInvalid choice");

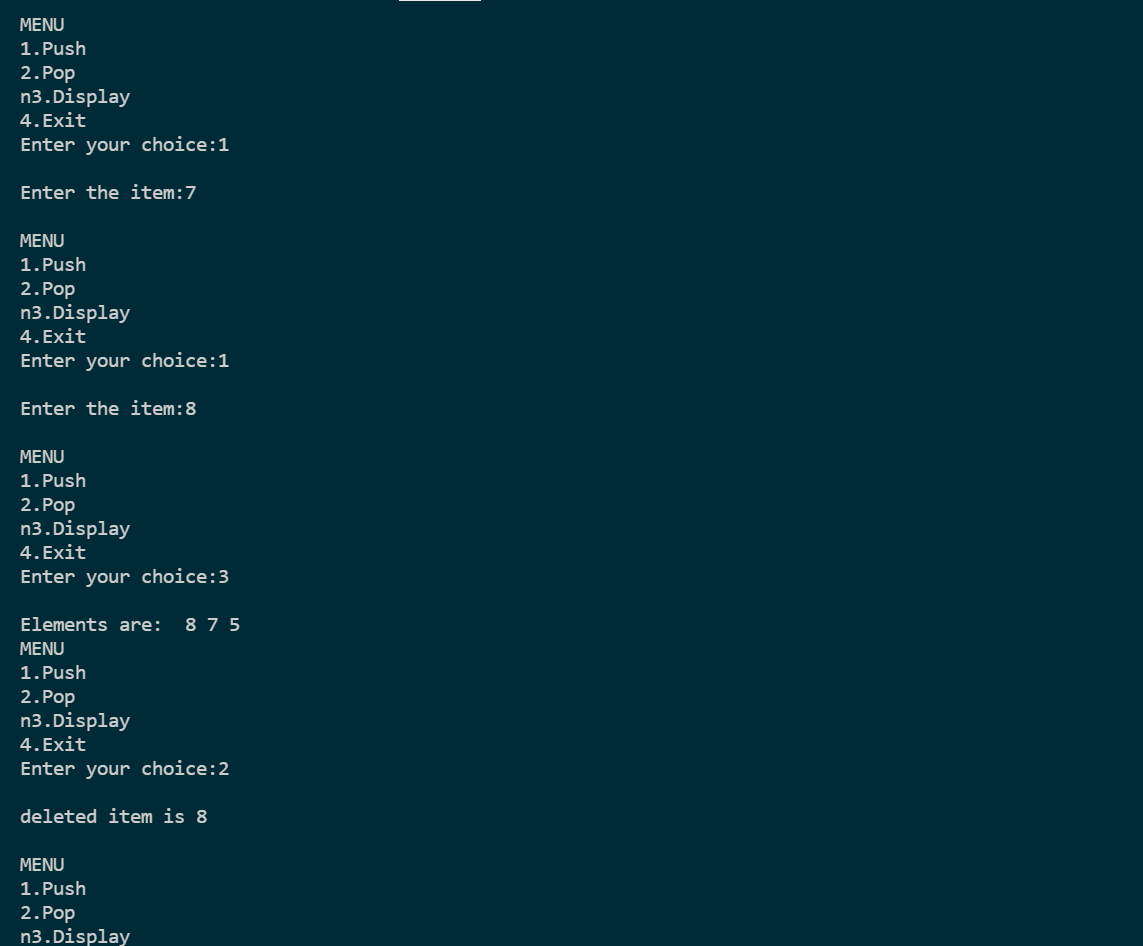
}

}

}

**OUTPUT**





4. Program to convert infix to postfix expression using stack

**PROGRAM CODE**

#include <stdio.h>

#include <ctype.h>

char stack[100];

int top = -1;

void push(char x)

{

stack[++top] = x;

}

char pop()

{

if (top == -1)

return -1;

else

return stack[top--];

}

int check(char x)

{

if (x == '(')

return 0;

if (x == '+' || x == '-')

return 1;

if (x == '\*' || x == '/')

return 2;

return 0;

}

int main()

{

int i=0;

char exp[100],x;

printf("Enter the expression : ");

scanf("%s", exp);

printf("\n");

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

{

if (isalnum(exp[i]))

printf("%c",exp[i]);

else if (exp[i] == '(')

push(exp[i]);

else if (exp[i] == ')')

{

while ((x = pop()) != '(')

printf("%c ", x);

}

else

{

while (check(stack[top]) >= check(exp[i]))

printf("%c ", pop());

push(exp[i]);

}

i++;

}

while (top != -1)

{

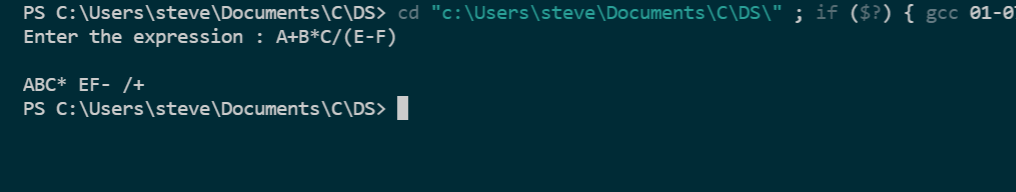
printf("%c",pop());

}

return 0;

}

**OUTPUT**



5. Reverse a string using stack and check whether it is a palindrome or not

**PROGRAM CODE**

#include<stdio.h>

#include<string.h>

#define max 100

int top=-1,stack[max];

void push(char x)

{

if (top== max-1)

printf("stack overflow");

else

stack[++top]=x;

}

char pop()

{

if(top!=-1)

return stack[top--];

}

void main()

{

int i,len,flag=0;

char str[30],rev[30];

printf("Enter a string: ");

scanf("%s",str);

len=strlen(str);

for(i=0;i<len;i++)

push(str[i]);

for(i=0;i<len;i++)

if(str[i]!=pop())

flag=1;

if (flag==0)

printf("\nIt is a palindrome");

else

printf("\nIt is not a palindrome");

}

**OUTPUT**

