Q.1) Implementation of stack using array

//program to implement stack using arrays

#include<stdio.h>

#include<stdlib.h>

#define M 5

typedef struct stack

{

int top;

int arr[M];

}STK;

void init(STK \*t)

{

t->top=-1;

}

int isFull(STK \*t)

{

return (t->top==M-1);

}

int isEmpty(STK \*t)

{

return (t->top==-1);

}

void push(STK \*t,int d)

{

if(isFull(t))

{

printf("\nOverflow");

return;

}

t->arr[++t->top]=d;

}

void pop(STK \*t)

{

if(isEmpty(t))

{

printf("\nUnderflow");

return;

}

t->top--;

}

void display(STK \*t)

{

int i=t->top;

if(isEmpty(t))

{

printf("\nempty stack");

return;

}

printf("\nStack Data\n");

while(i>=0)

{

printf("%d\n",t->arr[i--]);

}

}

void main()

{

int opt,d;

STK p;

init(&p);

while(1)

{

printf("\nMenu\n1.Push\n2.Pop\n3.Display\n4.exit\noption: ");

scanf("%d",&opt);

if(opt>3)

break;

switch(opt)

{

case 1:if(isFull(&p))

printf("\nOverflow");

else

{

printf("\nData: ");

scanf("%d",&d);

push(&p,d);

}

break;

case 2:if(isEmpty(&p))

printf("\nUnderflow");

else

pop(&p);

break;

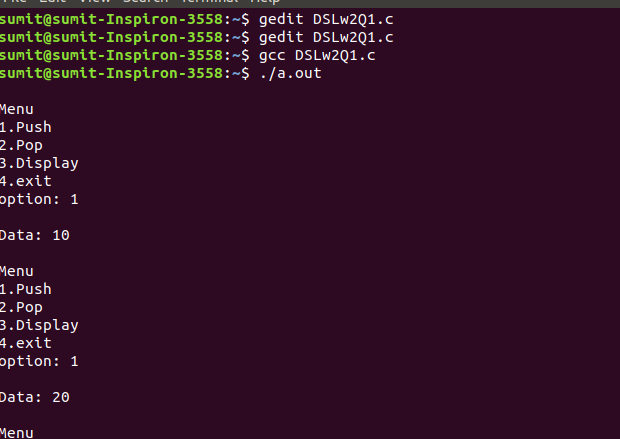
case 3:display(&p);

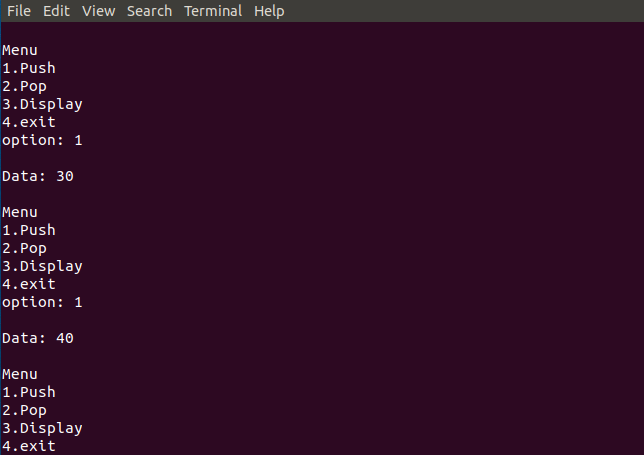
break;

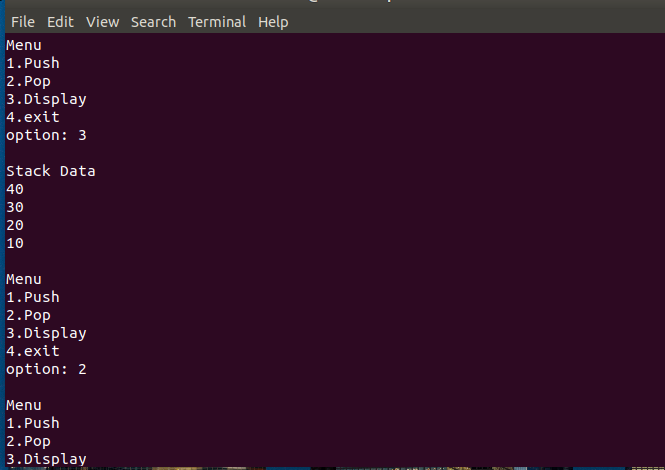
}

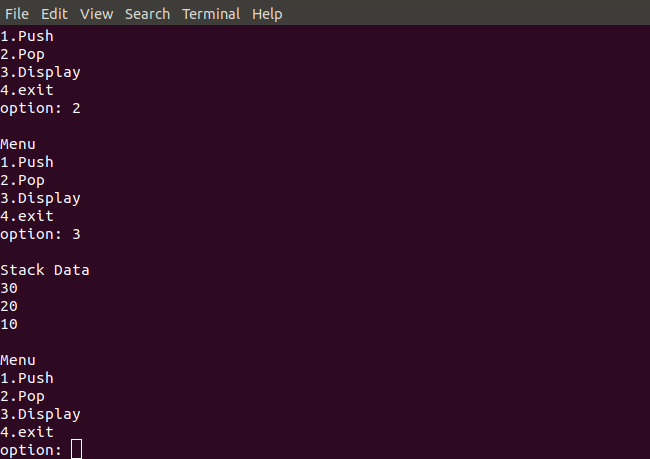
}

}









Q.2) conversion of infix expression to postfix expression

//program to convert infix expression to postfix

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct stack

{

int top;

char opr[50];

}S;

void init()

{

S.top=-1;

}

void push(char ch)

{

S.opr[++S.top]=ch;

}

void pop()

{

S.top--;

}

int isEmpty()

{

return (S.top==-1);

}

int preced(char stktop,char opr)

{

if(stktop!='(' && opr==')')

return 1;

else

return 0;

}

void main()

{

char istr[50],pstr[50];

int i=0,j=0;

init();

printf("\nEnter infix expression: ");

scanf("%s",istr);

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

{

if(istr[i]>=65 && istr[i]<=90)

pstr[j++]=istr[i];

else

{

while(!isEmpty() && preced(S.opr[S.top],istr[i]))

{

pstr[j++]=S.opr[S.top];

pop();

}

if(isEmpty() || istr[i]!=')')

push(istr[i]);

else

pop();

}

i++;

}

while(!isEmpty())

{

pstr[j++]=S.opr[S.top];

pop();

}

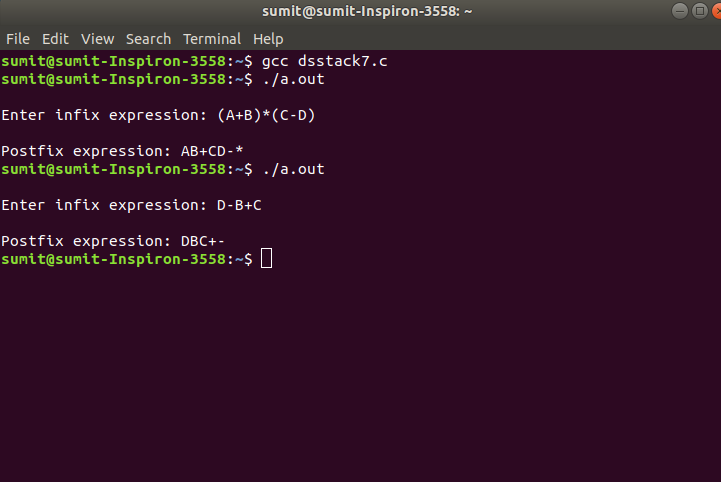
pstr[i]='\0';

printf("\nPostfix expression: %s",pstr);

printf("\n");

return;

}



Q.3) Evaluation of Postfix expression

//program for evaluation of postfix expression

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<ctype.h>

struct stack

{

int top;

int arr[10];

}S;

void init()

{

S.top=-1;

}

void push(int val)

{

S.top++;

S.arr[S.top]=val;

}

int isEmpty()

{

return(S.top==-1);

}

int pop()

{

return S.arr[S.top--];

}

void main()

{

char pstr[40],ch,val;

int i=0,a,b,res,index;

init();

printf("\nEnter Postfix string: ");

scanf("%s",pstr);

i=0;

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

{

ch=pstr[i];

if(isdigit(ch)){

push(ch-'0');

}

else if(ch=='+'||ch=='-'||ch=='\*'||ch=='/')

{

b=pop();

a=pop();

switch(ch)

{

case '+':val=a+b;

break;

case '-':val=a-b;

break;

case '\*':val=a\*b;

break;

case '/':val=a/b;

break;

}

push(val);

}

i++;

}

printf("\nResult:%d\n",pop());

return;

}

