

## DSA LAB 4

1) Evaluate a given prefix expression using stack.

```
#include<stdio.h>
#define MAX 20
typedef struct
{
int data[MAX];
int top;
}stack;
void init(stack *);
int empty(stack *);int full(stack *);
int pop(stack *);
void push(stack *,int);
int evaluate(char x,int op1,int op2);
int evaluate(char x,int op1,int op2)
{
if(x=='+')
return(op1+op2);
if(x=='-')
return(op1-op2);
if(x=='*')
return(op1*op2);
if(x=='/')
return(op1/op2);
if(x=='%')
return(op1%op2);
}
void init(stack *s)
{
s->top=-1;
}
int empty(stack *s)
{
if(s->top==-1)
return(1);
return(0);
}
int full(stack *s)
{
if(s->top==MAX-1)
return(1);
return(0);
}
```

```

}
void push(stack *s,int x)
{
s->top=s->top+1;
s->data[s->top]=x;
}
int pop(stack *s)
{
int x;
x=s->data[s->top];s->top=s->top-1;
return(x);
}
int main()
{
stack s;
char prefix[MAX];
char x=' ';
int op1,op2,val,i=0;
init(&s);
printf("Name:Yashas Kamath ; 200905132 ; Rno: 20 \nEnter the expression (eg:
*+593) with single digit operand and operators only:");
while(x!="\n"){
scanf("%c",&x);
prefix[i]=x;
i++;
}
for(i=i-1;i>-1;i--)
{
x=prefix[i];
if(x>=48 && x<=57)
push(&s,x-'0'); /*x-'0' for removing the effect of ascii */
else
{
op2=pop(&s);
op1=pop(&s);
val=evaluate(x,op1,op2);
push(&s,val);
}
}
val=pop(&s);
printf("\nvalue of expression=%d",val);
return 0;
}

```

```

Name:Yashas Kamath ; 200905132 ; Rno: 20
Enter the expression (eg: *+593) with single digit operand and operators only:*3+21
value of expression=9

```

2) Convert an infix expression to prefix.

```
#include<stdio.h>
#include<math.h>
#include<ctype.h>
#include<string.h>

#define MaxSize 10
typedef struct{
    char stack[MaxSize];
    int top;
}STACK;

void Push(STACK *s,char item){
    if (s->top==MaxSize-1)
    {
        printf("Stack Overflow");
        return;
    }
    s->stack[++(s->top)]=item;
}
char Pop(STACK *s)
{
    return (s->stack[(s->top)--]);
}

int isp(char x){
    switch(x){
        case '^': return 4; //right associative - higher value than icp(^) so as to
                        //pop ^
        case '*':
        case '/': return 2;
        case '+':
        case '-': return 1;
        case ')': return 0;
        case '#': return -1;
    }
}

int icp(char x){
    switch(x){
        case ')': return 5;
        case '^': return 3;
        case '*':
        case '/': return 2;
```

```

        case '+':
        case '-': return 1;
    }
}
void infix_prefix(char infix[], char prefix[])
{
    STACK *s, s1;
    int i=0,j=0;
    char x,y;

    s=&s1;
    s->top=-1;
    strrev(infix);
    Push(s,'#');
    for(x=infix[i];x!='\0'; i++,x=infix[i]){
        if (isdigit(x)){
            prefix[j++]=x;
        }
        else
            if (x == '(')
            {
                y = Pop(s);
                while(y!='') {
                    prefix[j++]=y;
                    y = Pop(s);
                }
            }
            else{
                y=Pop(s);
                while(isp(y)>icp(x)){ // only > to achieve
                    //proper conversion for left associative operators
                    prefix[j++] = y;
                    y = Pop(s);
                }

                Push(s,y); //last operator that is popped
                Push(s,x); //incoming operator
            }
    }

    y=Pop(s); //pop until eos

    while(y!='#'){

```

```

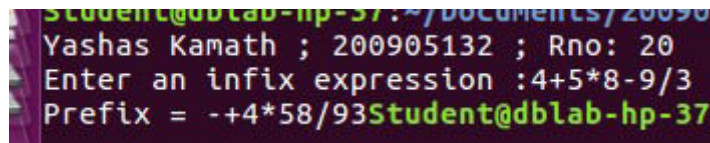
        prefix[j++]= y;
        y = Pop(s);

    }

    prefix[j]='\0';
    strrev(prefix);
}
int main()
{
    char infix[10],prefix[10];
    printf("Yashas Kamath ; 200905132 ; Rno: 20\n");
    printf("Enter an infix expression :");
    scanf("%s",infix);

    infix_prefix(infix,prefix);
    printf("Prefix = %s",prefix);
    return 0;
}

```



```

Student@dblab-hp-37:~/Documents/200905132
Yashas Kamath ; 200905132 ; Rno: 20
Enter an infix expression :4+5*8-9/3
Prefix = -+4*58/93Student@dblab-hp-37

```

3)Implement two stacks in an array.

```

#include<stdio.h>
#include<stdlib.h>
#define maxSize 6
//2 stacks
typedef struct {
    int top1,top2;
    int stack[maxSize];
}STACK;

//inserting elements into the first stack
void push1(STACK *s,int x){
    s->stack[++s->top1]=x;
}

//inserting elements into the second stack
void push2(STACK *s,int x){
    s->stack[--s->top2]=x;
}

//deleting elements from first stack
int pop1(STACK *s){
    return s->stack[s->top1--];
}

//deleting elements from first stack

```

```

int pop2(STACK *s){
    return s->stack[s->top2++];
}

//checking if the array is full
int isFull(STACK *s){
    if(s->top1+1==s->top2)
        return 1;
    return 0;
}

//checking if the first stack is empty
int isEmpty1(STACK *s){
    if(s->top1== -1)
        return 1;
    return 0;
}

//checking if second stack is empty
int isEmpty2(STACK *s){
    if(s->top2==maxSize)
        return 1;
    return 0;
}

int main(){
    int choice,element;
    char cont='y'; // cont checks whether we want to continue or not
    STACK s;
    s.top1=-1;
    s.top2=maxSize;
    printf("Name: Yashas Kamath Roll no: 20 Sec: D1 Reg no: 200905132\n");
    while(cont=='y' || cont=='Y'){
        printf("Enter your choice \n 1. Insert element into first stack \n 2. Insert element into the
second stack \n 3. Delete element from the first stack \n 4. Delete the element from the second
stack\n");
        scanf(" %d",&choice);
        switch(choice){
            case 1: if(isFull(&s)){
                printf("The array is full. ");
                break;
            }
            printf("Enter the element");
            scanf(" %d",&element);
            push1(&s,element);
            break;
            case 2: if(isFull(&s)){
                printf("The array is full. ");
                break;
            }
            printf("Enter the element");
            scanf(" %d",&element);

```

```
    push2(&s,element);
    break;
    case 3: if(isEmpty1(&s)){
        printf("The first stack is empty");
        break;
    }
    printf("The popped element from stack 1 is %d",pop1(&s));
    break;
    case 4: if(isEmpty2(&s)){
        printf("The second stack is empty");
        break;
    }
    printf("The popped element from stack 2 is %d",pop2(&s));
    break;
    default: printf("invalid choice!");
}
printf("\nDo you want to continue? Y/N");
scanf(" %c",&cont);
}
printf("Thankyou!");
}
```

Name: Yashas Kamath Roll no: 20 Sec: D1 Reg no: 200905132

Enter your choice

1. Insert element into first stack
2. Insert element into the second stack
3. Delete element from the first stack
4. Delete the element from the second stack

1

Enter the element 3

Do you want to continue? Y/N y

Enter your choice

1. Insert element into first stack
2. Insert element into the second stack
3. Delete element from the first stack
4. Delete the element from the second stack

1

Enter the element 5

Do you want to continue? Y/N y

Enter your choice

1. Insert element into first stack
2. Insert element into the second stack
3. Delete element from the first stack
4. Delete the element from the second stack

1

Enter the element 8

Do you want to continue? Y/N y

Enter your choice

1. Insert element into first stack
2. Insert element into the second stack
3. Delete element from the first stack
4. Delete the element from the second stack

2

Enter the element 6

Do you want to continue? Y/N y

Enter your choice

1. Insert element into first stack
2. Insert element into the second stack
3. Delete element from the first stack
4. Delete the element from the second stack

2



- 3. Delete element from the first stack
- 4. Delete the element from the second stack

2

Enter the element 9

Do you want to continue? Y/N y

Enter your choice

- 1. Insert element into first stack
- 2. Insert element into the second stack
- 3. Delete element from the first stack
- 4. Delete the element from the second stack

2

Enter the element 4

Do you want to continue? Y/N y

Enter your choice

- 1. Insert element into first stack
- 2. Insert element into the second stack
- 3. Delete element from the first stack
- 4. Delete the element from the second stack

2

The array is full.

Do you want to continue? Y/N y

Enter your choice

- 1. Insert element into first stack
- 2. Insert element into the second stack
- 3. Delete element from the first stack
- 4. Delete the element from the second stack

1

The array is full.

Do you want to continue? Y/N y

Enter your choice

- 1. Insert element into first stack
- 2. Insert element into the second stack
- 3. Delete element from the first stack
- 4. Delete the element from the second stack

3

The popped element from stack 1 is 8

Do you want to continue? Y/N y

Enter your choice

- 1. Insert element into first stack

```
Enter your choice
  1. Insert element into first stack
  2. Insert element into the second stack
  3. Delete element from the first stack
  4. Delete the element from the second stack
2
Enter the element 5

Do you want to continue? Y/N y
Enter your choice
  1. Insert element into first stack
  2. Insert element into the second stack
  3. Delete element from the first stack
  4. Delete the element from the second stack
3
The popped element from stack 1 is 5
Do you want to continue? Y/N y
Enter your choice
  1. Insert element into first stack
  2. Insert element into the second stack
  3. Delete element from the first stack
  4. Delete the element from the second stack
3
The popped element from stack 1 is 3
Do you want to continue? Y/N y
Enter your choice
  1. Insert element into first stack
  2. Insert element into the second stack
  3. Delete element from the first stack
  4. Delete the element from the second stack
3
The first stack is empty
Do you want to continue? Y/N y
Enter your choice
```

```
Enter your choice
  1. Insert element into first stack
  2. Insert element into the second stack
  3. Delete element from the first stack
  4. Delete the element from the second stack
4
The popped element from stack 2 is 5
Do you want to continue? Y/N y
Enter your choice
  1. Insert element into first stack
  2. Insert element into the second stack
  3. Delete element from the first stack
  4. Delete the element from the second stack
4
The popped element from stack 2 is 4
Do you want to continue? Y/N y
Enter your choice
  1. Insert element into first stack
  2. Insert element into the second stack
  3. Delete element from the first stack
  4. Delete the element from the second stack
4
The popped element from stack 2 is 9
Do you want to continue? Y/N y
Enter your choice
  1. Insert element into first stack
  2. Insert element into the second stack
  3. Delete element from the first stack
  4. Delete the element from the second stack
4
The popped element from stack 2 is 6
Do you want to continue? Y/N y
Enter your choice
  1. Insert element into first stack
  2. Insert element into the second stack
  3. Delete element from the first stack
  4. Delete the element from the second stack
4
The second stack is empty
Do you want to continue? Y/N n
Thankyou!Student@dblab-hp-37:~/Documents/200905132/1
```