Graphic Era
HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

Lab file

of

DATA STRUCTURE

(PCS 302)

2021-22

Submitted to:

Mr. RAKESH PATRA

GEHU, DDN

Submitted by:

VIPUL CHAUHAN

University Roll No : 2018855

Class Roll. No./Section: 62/A

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

ACKNOWLEDGMENT

I would like to particularly thank my Data Structures with C Lab Faculty Mr. RAKESH PATRA for his patience, support and encouragement throughout the completion of this project work.

At last but not the least I greatly indebted to all other persons who directly or indirectly helped me during this course.

VIPUL CHAUHAN

University Roll No - 2018855

B.Tech 3rd Sem CSE

Session: 2021-22 GEHU, dehradun

Graphic Era
HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

Table of Contents

7	Write a C Program implement STACK using Link List in menu driven form.	
8	Write a C Program implement QUEUE using Link List in menu driven form.	
9	Write a C Program implement priority QUEUE using array in menu driven form.	
10	Write a C Program implement QUEUE using array in menu driven form.	
11	Write a C program to Evaluate Postfix Expression using Stack	
12	Write a C program to create TWO singly linked list L1 and L2 and sort both the list and finally merge both the list such that L2 comes after L1.[use double pointer]	
13	Write C program to create a doubly link list by adding the node right hand side and then check list is in palindrome form or not.	
14	Write a C program to create a circular link list by adding the nodes in right hand side and then print the list	

Graphic Era
HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

DEPARTMENT OF CSE

B.Tech. CSE

STUDENT LAB REPORT SHEET

Name of Student ……..Mob.No……..

Address Permanent ……..

Father's Name ……Occupation ……Mob.No……

Mother's Name …… …..Occupation……Mob.No……..

Section ……Branch……Semester……..Class Roll No…….. Grade A B C

Local Address……Email…….. Marks 5 3

S.No.	Practical	D.O.P.	Date of Submission	Grade (Viva)	Grade (Report File)	Total Marks (out of 10)	Student's Signature	Teacher's Signature
1	Practical-01							
2	Practical-02							
3	Practical-03							
4	Practical-04							
5	Practical-05							
6	Practical-06							
7	Practical-07							
8	Practical-08							
9	Practical-09							
10	Practical-10							
11								

Q1. Write a C program to create an array by inserting N elements in it then find second non repeating element from the array.

```
#include <stdio.h>
int main()
{
   int n,i,j,count=0,flag=0;;
   printf("Enter the size of the array : ");
   scanf("%d",&n);
   int arr[n], arr1[n];
   printf("Enter %d elements : ",n);
   for (i=0;i<n;i++)
   {
      scanf("%d",&arr[i]);
   }
   for (i=0;i<n;i++)
   {
      count=0;
      for (j= 0;j<=n;j++)
      {
         if (arr[i]==arr[j] && i!=j)
         {
            count=1;
         }
      }
      if (count==0)
      {
```

```
            arr1[flag++]=arr[i];
        }
    }
    printf("Non repeating element of array are : ");
    for (i=0;i<flag;i++)
    {
        printf("%d ",arr1[i]);
    }
    if (flag>1)
    {
        printf("\nSecond Non  repeating element of array is : %d",arr1[1]);
    }
    else if (flag==1)
    {
        printf("\nThere is only 1 non repeting element in array i.e %d",arr1[0]);
    }
    else
    {
        printf("\nThere is No non repeting element in the array.");
    }
    return 0;
}
```

Output

Enter the size of the array : 9

Enter 9 elements : 1 2 1 2 3 4 5 6 6

Non repeating element of array are : 3 4 5

Second Non repeating element of array is : 4

Q2. Write a the C program to create an array by inserting N elements in it then find third repeating element from the array.

```
#include <stdio.h>
int main()
{
 int n,i,j;
 printf("Enter the size of array : ");
 scanf("%d",&n);
 int a[n];
 int b=0,visited=0;
 printf("Enter %d elements of an array : ",n);
 for(i=0;i<n;i++)
 {
  scanf("%d",&a[i]);
 }
 for(i=0;i<n;i++)
 {
  int c=0;
  for(j=i+1;j<n;j++)
  {
   if(a[i]!=visited)
   {
    if(a[i]==a[j])
    {
        c++;
        a[j]=visited;
```

```
      }
     }
   }
   if(c>0)
   {
     b++;
   }
   if(b==3)
   {
     printf("The third repeating element is = %d ",a[i]);
     break;
   }
  }
  if(b<3)
   printf("No thrid repeating element");
}
```

Output

Enter the size of array : 7

Enter 7 elements of an array : 1 1 2 2 3 3 4

The third repeating element is = 3

Q3. Write a C program Create a Dynamic array and then Reverse the array using recursion and then finally print the array.

```
#include<stdio.h>
#include<stdlib.h>
//recursion function to make the array reverse
void reverse(int *arr,int start,int end)
{
   if(start<end)
   {
      arr[start]=arr[start]+arr[end];
      arr[end]=arr[start]-arr[end];
      arr[start]=arr[start]-arr[end];
      reverse(arr,start+1,end-1);
   }
   return ;
}
int main()
{
   int n,i;
   printf("Enter the number of elements you want to enter in an array : ");
   scanf("%d",&n);
   //Dynamic array
   int *arr=(int *)malloc(n*sizeof(int));
   printf("Enter %d elements : ",n);
   for(i=0;i<n;i++)
   {
```

```
        scanf("%d",&arr[i]);
    }
    reverse(arr,0,n-1);
    printf("Array after reversing is : ");
    for(i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
    free(arr);
    return 0;
}
```

Output

Enter the number of elements you want to enter in an array : 5

Enter 5 elements : 1 2 3 4 5

Array after reversing is : 5 4 3 2 1

Q4. Write a C Program implement STACK using array in menu driven form.

```
#include<stdio.h>
#define MAX 10
//Function to enter the element in a stack
void push(int *a,int *top)
{
	if(*top>=MAX)
	{
		printf("Stack is full\n");
	}
	else
	{
		(*top)++;
		printf("Enter the element:");
		scanf("%d",&a[*top]);
	}
}
//Function to delete the top most element in a stack
void pop(int *a,int *top)
{
	if(*top==-1)
	{
		printf("Stack is empty\n");
	}
	else
```

```
	{
		printf("The element at index %d that is  %d is being deleted",*top,a[*top]);
		(*top)--;
	}
}
//Function to display the top most element in a stack
void peek(int *a,int *top)
{
	if(*top==-1)
	{
		printf("Stack is empty\n");
	}
	else
	{
		printf("Top most element in a stack is : ");
		printf("%d",a[*top]);
	}
}
//Function to display all the elemnts of the stack
void display(int *a, int *top)
{
	if(*top==-1)
	{
		printf("Stack is empty\n");
	}
	else
	{
```

```
                printf("Elements of stacks are:\n");
                for(int j=*top;j>-1;j--)
                {
                        printf("Element at index %d is %d \n",j,a[j]);
                }
        }
}
int main()
{
        int choice,top=-1;
        int a[MAX];
        printf("Enter the number in the menu to do the task in an stack:\n");
        //Menu driven for performing all the stack functions
        do
        {
                printf("\n\n1-Push\n2-Pop\n3-Peek\n4-Display\n5-Exit\n");
                printf("Enter your choice : ");
                scanf("%d",&choice);
                switch(choice)
                {
                        case 1:
                                push(a,&top);
                                break;
                        case 2:
                                pop(a,&top);
                                break;
                        case 3:
```

```
                peek(a,&top);
                break;
            case 4:
                display(a,&top);
                break;
            case 5:
                break;
            default:
                printf("You enter the wrong choice:\n");
        }
    }while(choice!=5);
    return 0;
}
```

Output

```
Enter the number in the menu to do the task in an stack:
1-Push
2-Pop
3-Peek
4-Display
5-Exit
Enter your choice : 1
Enter the element:2

1-Push
2-Pop
3-Peek
```

4-Display

5-Exit

Enter your choice : 1

Enter the element:4

1-Push

2-Pop

3-Peek

4-Display

5-Exit

Enter your choice : 4

Elements of stacks are:

Element at index 1 is 4

Element at index 0 is 2

1-Push

2-Pop

3-Peek

4-Display

5-Exit

Enter your choice : 2

The element at index 1 that is 4 is being deleted

1-Push

2-Pop

3-Peek

4-Display

5-Exit

Enter your choice : 3

Top most element in a stack is : 2

1-Push

2-Pop

3-Peek

4-Display

5-Exit

Enter your choice : 5

Q5. Write a C Program to Convert Infix to Postfix Expression using Stack.

```
#include<stdio.h>                    //Infix to Postfix
#include<ctype.h>
void push(char x,char *stack,int *top)
{
   stack[++(*top)] = x;
}
char pop(char *stack,int *top)
{
   if(*top == -1)
      return -1;
   else
      return stack[(*top)--];
```

```
}
int priority(char x)
{
   if(x == '(')
      return 0;
   if(x == '+' || x == '-')
      return 1;
   if(x == '*' || x == '/')
      return 2;
   if(x=='^')
      return 3;

   return 0;
}
int main()
{
   char stack[50];
   int top = -1;
   char exp[100];
   char  x;
   int i=0;
   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],stack,&top);
        else if(exp[i] == ')')
        {
            while((x = pop(stack,&top)) != '(')
                printf("%c", x);
        }
        else          //priority
        {
            while(priority(stack[top]) >= priority(exp[i]))
                printf("%c",pop(stack,&top));
            push(exp[i],stack,&top);
        }
        i++;
    }
    while(top != -1)
    {
        printf("%c",pop(stack,&top));
    }
    printf("\n");
    return 0;
}
```

Output

```
Enter the expression : a+b*c/d
abc*d/+
```

Q6. Write a C Program to create singly linked list by adding nodes in the right hand side and delete alternate node from the list and then print the final list.

```
#include<stdio.h>
#include<stdlib.h>
typedef struct node
{
   int data;
   struct node *next;
}node;
void delete(node * head)
{
   if(head==NULL)
   {
      return ;
   }
   node *prev=head;
   node *temp=head->next;
   while(prev!=NULL && temp!=NULL)
   {
      prev->next=temp->next;
      free(temp);
      prev=prev->next;
      if(prev!=NULL)
      {
         temp=prev->next;
      }
```

```
    }
}
int main()
{
    int n,i;
    node *head=NULL,*p=NULL,*temp=NULL;
    printf("Enter the number of elements you want enter in singly linked list : ");
    scanf("%d",&n);
    printf("Enter %d elements in linked list : ",n);
    for(i=0;i<n;i++)
    {
        p=(node *)malloc(1*sizeof(node));
        scanf("%d",&p->data);
        p->next=NULL;
        if(i==0)
        {
            head=temp=p;
        }
        else
        {
            temp->next=p;
            temp=p;
        }
    }
    delete(head);
    printf("Alternate Elements after deletion from the linked list are : ");
    for(temp=head;temp!=NULL;temp=temp->next)
```

```
    {
        printf("%d ",temp->data);
    }
    return 0;
}
```

Output

Enter the number of elements you want enter in singly linked list : 5

Enter 5 elements in linked list : 1 2 3 4 5

Alternate Elements after deletion from the linked list are : 1 3 5

Q7. Write a C Program implement STACK using Link List in menu driven form.

```
#include<stdio.h>
#include<stdlib.h>
#define MAX 10
typedef struct node
{
   int data;
   struct node *next;
}node;
//Function to enter the element in a stack
node* push(node *head)
{
   node *p=(node *)malloc(1*sizeof(node));
         if(p==NULL)
         {
                  printf("Stack is full\n");
         }
         else
         {
                  printf("Enter the element:");
                  scanf("%d",&p->data);
                  if(head==NULL)
                  {
                     p->next=NULL;
                  }
                  else
```

```
                {
                   p->next=head;
                }
                head=p;
        }
        return head;
}
//Function to delete the top most element in a stack
node *pop(node *head)
{
   node *del;
        if(head==NULL)
        {
                printf("Stack is empty\n");
        }
        else
        {
                printf("The element %d is being deleted",head->data);
                del=head;
                head=head->next;
                free(del);
        }
        return head;
}
//Function to display the top most element in a stack
void peek(node *head)
{
```

```
        if(head==NULL)
        {
                printf("Stack is empty\n");
        }
        else
        {
                printf("Top most element in a stack is :");
                printf("%d",head->data);
        }
}
//Function to display all the elemnts of the stack
void display(node *head)
{
   int i=0;
        if(head==NULL)
        {
                printf("Stack is empty\n");
        }
        else
        {
                printf("Elements of stacks are:\n");
                for(node *temp=head;temp!=NULL;temp=temp->next)
                {
                        printf("Element at index %d is %d \n",i,temp->data);
                        i++;
                }
        }
```

```
}
int main()
{
   int choice;
        node *head=NULL;
        printf("Enter the number in the menu to do the task in an stack:\n");
        //Menu driven for performing all the stack functions
        do
        {
                printf("\n\n1-Push\n2-Pop\n3-Peek\n4-Display\n5-Exit\n");
                printf("Enter your choice : ");
                scanf("%d",&choice);
                switch(choice)
                {
                        case 1:
                                head=push(head);
                                break;
                        case 2:
                                head=pop(head);
                                break;
                        case 3:
                                peek(head);
                                break;
                        case 4:
                                display(head);
                                break;
                        case 5:
```

```
                    break;
                default:
                    printf("You enter the wrong choice:\n");
            }
        }while(choice!=5);
        free(head);
        return 0;
}
```

Output

```
Enter the number in the menu to do the task in an stack:
1-Push
2-Pop
3-Peek
4-Display
5-Exit
Enter your choice : 1
Enter the element:23

1-Push
2-Pop
3-Peek
4-Display
5-Exit
Enter your choice : 1
Enter the element:24
```

1-Push

2-Pop

3-Peek

4-Display

5-Exit

Enter your choice : 4

Elements of stacks are:

Element at index 0 is 24

Element at index 1 is 23

1-Push

2-Pop

3-Peek

4-Display

5-Exit

Enter your choice : 2

The element 24 is being deleted

1-Push

2-Pop

3-Peek

4-Display

5-Exit

Enter your choice : 4

Elements of stacks are:

Element at index 0 is 23

1-Push

2-Pop

3-Peek

4-Display

5-Exit

Enter your choice : 5

Q8. Write a C Program implement QUEUE using Link List in menu driven form.

```
#include<stdio.h>
#include<stdlib.h>
typedef struct node
{
   int data;
   struct node *next;
}node;
//Function to enter the element in the Que
node *enqeue(node *head,node *last)
{
        node *p=(node *)malloc(1*sizeof(node));
        if(p==NULL)
        {
                printf("Que is full\n");
        }
        else
```

```
        {
                printf("Enter the number:");
                scanf("%d",&p->data);
                p->next=NULL;
                if(head==NULL)
                {
                   head=p;
                   last=p;
                   return head;
                }
                else
                {
                   last->next=p;
                   last=p;
                }
                return last;
        }
}
//Function to delete the element from the Que
node* deqeue(node *head)
{
   node *del;
        if(head==NULL)
        {
                printf("Que is empty\n");
        }
        else
```

```
        {
                printf("The element %d is being deqeue\n",head->data);
                del=head;
                head=head->next;
                free(del);
        }
        return head;
}
//Function to display the top element of the Que
void peek(node *head)
{
        if(head==NULL)
        {
                printf("Que is empty\n");
        }
        else
        {
                printf("The front element of que is %d\n",head->data);
        }
}
//Function to display elements in the Que
void display(node *head)
{
   int i=0;
   if(head==NULL)
        {
                printf("Que is empty\n");
```

```
        }
        for(node *temp=head;temp!=NULL;temp=temp->next,i++)
        {
                printf("Element at index %d is %d\n",i,temp->data);
        }
}
int main()
{
        node *head=NULL,*last=NULL;
        int n;
        printf("Enter the your choice to perform the following functions \n");
        do
        {
                printf("\n1-Enqeue\n2-Deque\n3-Peek\n4-Display\n5-Exit\nEnter your choice:");
                scanf("%d",&n);
                switch(n)
                {
                        case 1:
                                last=enqeue(head,last);
                                if(head==NULL)
                                {
                                   head=last;
                                }
                                break;
                        case 2:
                                head=deqeue(head);
                                break;
```

```
                case 3:
                        peek(head);
                        break;
                case 4:
                        display(head);
                        break;
                case 5:
                        break;
                default:
                        printf("You entered the wrong choice\n");
                }
        }while(n!=5);
        return 0;
}
```

Output

```
Enter the your choice to perform the following functions
1-Enqeue
2-Deque
3-Peek
4-Display
5-Exit
Enter your choice:1
Enter the number:12

1-Enqeue
2-Deque
```

3-Peek

4-Display

5-Exit

Enter your choice:1

Enter the number:14

1-Enqeue

2-Deque

3-Peek

4-Display

5-Exit

Enter your choice:4

Element at index 0 is 12

Element at index 1 is 14

1-Enqeue

2-Deque

3-Peek

4-Display

5-Exit

Enter your choice:3

The front element of que is 12

1-Enqeue

2-Deque

3-Peek

4-Display

5-Exit

Enter your choice:2

The element 12 is being deqeue

1-Enqeue

2-Deque

3-Peek

4-Display

5-Exit

Enter your choice:4

Element at index 0 is 14

1-Enqeue

2-Deque

3-Peek

4-Display

5-Exit

Enter your choice:3

The front element of que is 14

1-Enqeue

2-Deque

3-Peek

4-Display

5-Exit

Enter your choice:5

Q9. Write a C Program implement priority QUEUE using array in menu driven form.

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 5
typedef struct priorityqueue
{
   int ele;
   int priority;

}que;
int isempty(int rear)
{
   if(rear==-1)
   {
      return 1;
   }
   else
   {
      return 0;
   }
}
int isfull(int rear)
{
   if(rear==MAX-1)
   {
```

```
      return 1;
   }
   else
   {
      return 0;
   }
}
void insert(que *pq,int *rear,int ele,int p)
{
          *rear=*rear+1;
          pq[*rear].ele=ele;
          pq[*rear].priority=p;
}
int gethighestpriority(que*pq,int rear)
{
   int i,p;
   p=-1;
   if(!isempty(rear))
   {
      for(i=0;i<=rear;i++)
      {
         if(pq[i].priority>p)
         {
            p=pq[i].priority;
         }
      }
   }
```

```
   return p;
}
int deletehighestpriority(que *pq,int *rear)
{
   int i,j,p,ele;
   p=gethighestpriority(pq,*rear);
   for(i=0;i<=*rear;i++)
   {
      if(pq[i].priority==p)
      {
         ele=pq[i].ele;
         break;
      }
   }
   if(i<*rear)
   {
      for(j=i;j<*rear;j++)
      {
         pq[j].ele=pq[j+1].ele;
         pq[j].priority=pq[j+1].priority;
      }
   }
   *rear=*rear-1;
   return ele;
}
void display(que *pq,int rear)
{
```

```
   int i;
   printf("Priority queue: \n");
   for(i=0;i<=rear;i++)
   {
      printf("Element: %d Priority: %d\n", pq[i].ele, pq[i].priority);
   }
}
int main()
{
   que pq[MAX];
   int rear=-1;
         int ch,p,ele;
         printf("Enter\n");
   do
   {
      printf("\n1-To insert\n2-To get highest priority\n3-To delete\n4-To display\n5-To exit\n");
      printf("Enter your choice:");
      scanf("%d",&ch);
      switch(ch)
      {
         case 1:
                              if(isfull(rear))
            printf("Priority queue is full.");
           else
                                    {
                                          printf("Enter element to insert :");
                                          scanf("%d",&ele);
```

```
                printf("Enter priority :");
                scanf("%d",&p);
                insert(pq,&rear,ele,p);
            }
        break;
    case 2:
        if(isempty(rear))
        {
            printf("The queue is empty\n");
        }
        else
        {
            p=gethighestpriority(pq,rear);
            printf("The highest priority = %d\n",p);
        }
        break;
    case 3:
        if (isempty(rear))
        {
            printf("Queue is empty\n");
        }
        else
        {
            ele=deletehighestpriority(pq,&rear);
            printf("%d is deleted\n", ele);
        }
        break;
```

```
        case 4:
           display(pq,rear);
           break;
        default:
           break;
      }
   } while (ch != 5);
}
```

Output

```
Enter

1-To insert
2-To get highest priority
3-To delete
4-To display
5-To exit
Enter your choice:1
Enter element to insert :12
Enter priority :13

1-To insert
2-To get highest priority
3-To delete
4-To display
5-To exit
Enter your choice:1
```

Enter element to insert :14

Enter priority :15

1-To insert

2-To get highest priority

3-To delete

4-To display

5-To exit

Enter your choice:4

Priority queue:

Element: 12 Priority: 13

Element: 14 Priority: 15

1-To insert

2-To get highest priority

3-To delete

4-To display

5-To exit

Enter your choice:2

The highest priority = 15

1-To insert

2-To get highest priority

3-To delete

4-To display

5-To exit

Enter your choice:3

14 is deleted

1-To insert

2-To get highest priority

3-To delete

4-To display

5-To exit

Enter your choice:4

Priority queue:

Element: 12 Priority: 13

1-To insert

2-To get highest priority

3-To delete

4-To display

5-To exit

Enter your choice:5

Q10. Write a C Program implement QUEUE using array in menu driven form.

```
#include<stdio.h>
#define MAX 10
//Function to enter the element in the Que
void enqeue(int *q,int *f,int *r)
{
	if(*r==-1 || *f==-1)
	{
		*r=0;
		*f=0;
	}
	if(*r>=MAX)
	{
		printf("Que is full\n");
	}
	else
	{
		printf("Enter the number:");
		scanf("%d",&q[*r]);
		(*r)++;
	}
}
//Function to delete the element from the Que
void deqeue(int *q,int *f,int *r)
{
```

```
	if(*r==-1)
	{
		printf("Que is empty\n");
	}
	else
	{
		printf("The element at index %d that is %d is being deqeue\n",*f,q[*f]);
		for(int i=*f;i<*r;i++)
		{
		   q[i]=q[i+1];
		}
		(*r)--;
	}

}
//Function to display the top element of the Que
void peek(int *q,int *f,int *r)
{
	if(*r==-1)
	{
		printf("Que is empty\n");
	}
	else
	{
		printf("The front element of que is %d\n",q[*f]);
	}
}
```

```
//Function to display elements in the Que
void display(int *q,int *f,int *r)
{
   if(*r==-1)
   {
      printf("Que is empty\n");
   }
        for(int i=*f;i<*r;i++)
        {
                printf("Element at index %d is %d\n",i,q[i]);
        }
}
int main()
{
        int front=-1,rear=-1,arr[MAX],n;
        printf("Enter the your choice to perform the following functions \n");
        do
        {
                printf("\n1-Enqeue\n2-Deque\n3-Peek\n4-Display\n5-Exit\nEnter your choice:");
                scanf("%d",&n);
                switch(n)
                {
                        case 1:
                                enqeue(arr,&front,&rear);
                                break;
                        case 2:
                                deqeue(arr,&front,&rear);
```

```
                        break;
                case 3:
                        peek(arr,&front,&rear);
                        break;
                case 4:
                        display(arr,&front,&rear);
                        break;
                case 5:
                        break;
                default:
                        printf("You entered the wrong choice\n");
            }
        }while(n!=5);
        return 0;
}
```

Output

Enter the your choice to perform the following functions

1-Enqeue

2-Deque

3-Peek

4-Display

5-Exit

Enter your choice:1

Enter the number:12

1-Enqeue

2-Deque

3-Peek

4-Display

5-Exit

Enter your choice:1

Enter the number:14

1-Enqeue

2-Deque

3-Peek

4-Display

5-Exit

Enter your choice:4

Element at index 0 is 12

Element at index 1 is 14

1-Enqeue

2-Deque

3-Peek

4-Display

5-Exit

Enter your choice:2

The element at index 0 that is 12 is being deqeue

1-Enqeue

2-Deque

3-Peek

4-Display

5-Exit

Enter your choice:3

The front element of que is 14

1-Enqeue

2-Deque

3-Peek

4-Display

5-Exit

Enter your choice:5

Q11. Write a C program to Evaluate Postfix Expression using Stack

```
//Evaluate postfix expression
#include<stdio.h>
#include<ctype.h>
int stack[20];
int top = -1;
void push(int x)
{
   stack[++top] = x;
}
int pop()
{
   return stack[top--];
}
int main()
{
   char exp[20];
   char *e;
   int n1,n2,n3,num;
   printf("Enter the expression : ");
   scanf("%s",exp);
   e = exp;
   while(*e != '\0')
   {
      if(isdigit(*e))
      {
```

```
        num = *e - 48;
        push(num);
    }
    else
    {
        n1 = pop();
        n2 = pop();
        switch(*e)
        {
        case '+':
        {
            n3 = n1 + n2;
            break;
        }
        case '-':
        {
            n3 = n2 - n1;
            break;
        }
        case '*':
        {
            n3 = n1 * n2;
            break;
        }
        case '/':
        {
            n3 = n2 / n1;
```

```
            break;
          }
          }
          push(n3);
       }
       e++;
   }
   printf("\nThe result of expression %s  =  %d\n\n",exp,pop());
   return 0;
}
```

Output

Enter the expression : 45+6*

The result of expression 45+6* = 54

Q 12. Write a C program to create TWO singly linked list L1 and L2 and sort both the list and finally merge both the list such that L2 comes after L1.[use double pointer]

```
#include <stdio.h>
#include <stdlib.h>
typedef struct node
{
   int info;
   struct node *next;
}node;
void push(node **head,int val)
{
   node *newNode = malloc(sizeof(node));
   newNode->info = val;
   newNode->next = NULL;
   if (*head == NULL)
      *head = newNode;
   else
   {
      node *lastNode = *head;
      while (lastNode->next != NULL)
      {
         lastNode = lastNode->next;

      }
      lastNode->next = newNode;
```

```
    }
}
void sort(node *head)
{
    node *temp;
    while(head!=NULL)
    {
        temp=head->next;
        while(temp!=NULL)
        {
            if(head->info>temp->info)
            {
                int hold=head->info;
                head->info=temp->info;
                temp->info=hold;
            }
            temp=temp->next;
        }
        head =head->next;
    }
}
void merge(node *l1,node *l2)
{
    while(l1->next!=NULL)
    {
        l1=l1->next;
    }
    l1->next=l2;
```

```
}
void print(node *ptr)
{
   struct node *temp = ptr;
   while (temp != NULL)
   {
      printf("%d ", temp->info);
      temp = temp->next;

   }
}
int main()
{
   int n,num;
   node *l1 = NULL,*l2 = NULL;
   printf("Enter how many number you want to enter in the first linked list : ");
   scanf("%d",&n);
   for(int i=0;i<n;i++)
   {
      printf("Enter the number : ");
      scanf("%d",&num);
      push(&l1,num);
   }
   sort(l1);
   printf("Enter how many number you want to enter in the second linked list : ");
   scanf("%d",&n);
   for(int i=0;i<n;i++)
   {
```

```
        printf("Enter the number : ");
        scanf("%d",&num);
        push(&l2,num);
    }
    sort(l2);
    merge(l1,l2);
    printf("Linked list after merging are : ");
    print(l1);
}
```

Output

Enter how many number you want to enter in the first linked list : 3

Enter the number : 4

Enter the number : 3

Enter the number : 1

Enter how many number you want to enter in the second linked list : 3

Enter the number : 1

Enter the number : 6

Enter the number : 3

Linked list after merging are : 1 3 4 1 3 6

Q 13. Write C program to create a doubly link list by adding the node right hand side and then check list is in palindrome form or not.

```
#include<stdio.h>
#include<stdlib.h>
typedef struct Node
{
   int data;
   struct Node *prev;
   struct Node *next;
}node;
void insert(node **head,node **tail,int value)
{
   node *temp=(node*)malloc(sizeof(node));
   temp->data=value;
   temp->prev=temp->next=NULL;
   if(*head==NULL&&*tail==NULL)
   {
      *head=temp;
      *tail=temp;
   }
   else
   {
      (*tail)->next=temp;
      temp->prev=*tail;
      *tail=temp;
   }
}
```

```
void PalindromeChecker(node *head,node *tail)
{
   int counter=0;
   while(head->next!=NULL||tail->prev!=NULL)
   {
      if(head->data!=tail->data)
      {
         counter++;
         break;

      }
      head=head->next;
      tail=tail->prev;

   }
   if(counter)
      printf("Not in Palindrome Form\n");
   else
      printf("In Palindrome Form\n");
}
int main()
{
   int num,value;
   node *head=NULL,*tail=NULL;
   printf("Enter the total number of nodes to enter : ");
   scanf("%d",&num);
   fflush(stdin);
```

```
    for(int i=0;i<num;i++)
    {
        printf("Enter node %d data: ",(i+1));
        scanf("%d",&value);
        insert(&head,&tail,value);
    }
    PalindromeChecker(head,tail);
    return 0;
}
```

Output

```
Enter the total number of nodes to enter : 5
Enter node 1 data: 12
Enter node 2 data: 12
Enter node 3 data: 13
Enter node 4 data: 12
Enter node 5 data: 12
In Palindrome Form
```

Q14. Write a C program to create a circular link list by adding the nodes in right hand side and then print the list.

```
#include<stdio.h>
#include<stdlib.h>
typedef struct Node
{
   int data;
   struct Node *next;
}node;
void insert(node **last,int value)
{
   node *temp=(node*)malloc(sizeof(node));
   temp->data=value;
   if(temp==NULL)
   temp->next=NULL;
   if(*last==NULL)
   {
      *last=temp;
      (*last)->next=temp;
   }
   else
   {
      temp->next=(*last)->next;
      (*last)->next=temp;
      *last=(*last)->next;
   }
   printf("\n");
}
```

```
void display(node *last)
{
   printf("The elements are : ");
   node *head=last->next;
   while(head!=last)
   {
      printf("%d ",head->data);
      head=head->next;
   }
   printf("%d \n",head->data);
   printf("\n");
}
int main()
{
   node *last;
   int counter=0,choice,value;
   last=NULL;
   printf("Enter\n");
   while(counter==0)
   {
      printf("1.To insert in the circular linked list \n2.To display the circular linked list \n3.Exit\n");
      printf("Enter your choice : ");
      scanf("%d",&choice);
      switch(choice)
      {
         case 1:
            printf("Enter a value : ");
            scanf("%d",&value); insert(&last,value);
            break;
```

```
        case 2:
            display(last);
            break;
        case 3:
            counter++;
            break;
        default:
            printf("WRONG CHOICE\n");
            break;
      }
   }
   return 0;
}
```

Output

Enter

1.To insert in the circular linked list

2.To display the circular linked list

3.Exit

Enter your choice : 1

Enter a value : 12

1.To insert in the circular linked list

2.To display the circular linked list

3.Exit

Enter your choice : 1

Enter a value : 14

1.To insert in the circular linked list

2.To display the circular linked list

3.Exit

Enter your choice : 2

The elements are : 12 14

1.To insert in the circular linked list

2.To display the circular linked list

3.Exit

Enter your choice : 3