Question (1)

//write a program to implement stack with push,pop,peep and update operation.

```
#include<stdio.h>
#include<conio.h>
#define n 5
int top=-1;
int s[n];

void push(int data)
{
	if(top==n-1)
	{
		printf("\n\t\tStack is overflow");
		return;
	}
	top++;
	s[top]=data;
}
int pop()
{
	int data;
	if(top==-1)
	{
		printf("\n\t\tStack is underflow");
		return 0;
	}
```

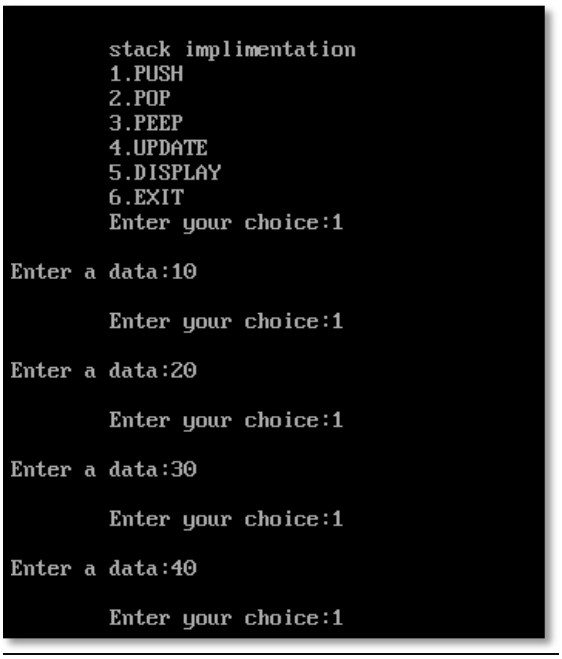
```
        data=s[top];
        top--;
        return data;
}
int peep(int p)
{
        if(top-p+1<0)
        {
                printf("\n\t\tStack is underflow in peep");
                return 0;
        }
        else
                return s[top-p+1];
}
void update(int p,int data)
{
        if(top-p+1<0)
        {
                printf("\n\t\tStack is underflow in update");
                return;
        }
        else
                s[top-p+1]=data;
}
void disp()
{
```

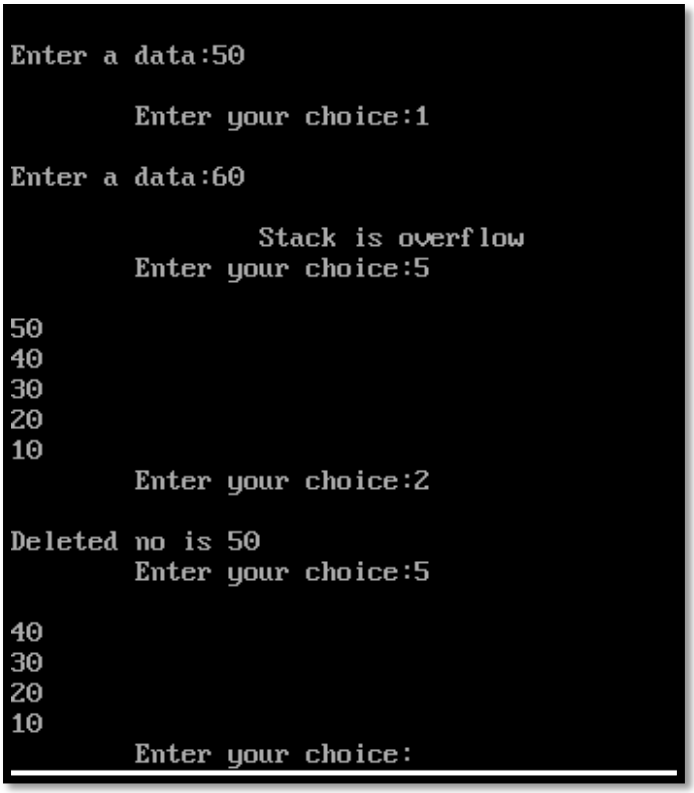
```
        int i;
        for(i=top;i>=0;i--)
                printf("\n%d",s[i]);
}
void main()
{
        int e,ch,p;
        clrscr();
        printf("\n\tstack implimentation");
        printf("\n\t1.PUSH");
        printf("\n\t2.POP");
        printf("\n\t3.PEEP");
        printf("\n\t4.UPDATE");
        printf("\n\t5.DISPLAY");
        printf("\n\t6.EXIT");
        do
        {
        printf("\n\tEnter your choice:");
        scanf("%d",&ch);
        switch(ch)
        {
                case 1:
                        printf("\nEnter a data:");
                        scanf("%d",&e);
                        push(e);
                        break;
```

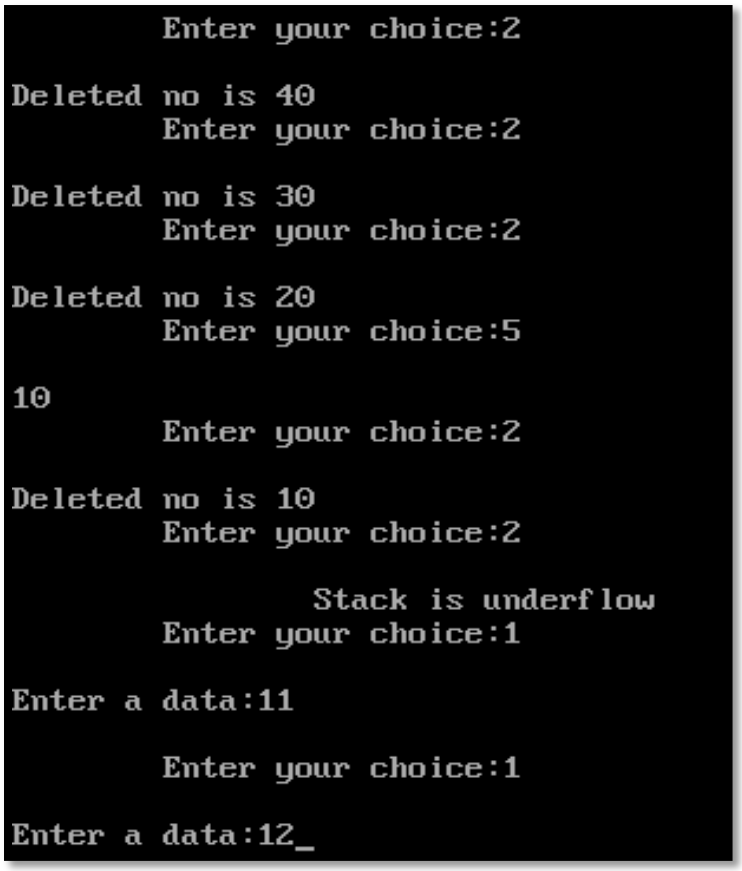
```
case 2:
        e=pop();
        if(e!=0)
                printf("\nDeleted no is %d",e);
        break;
case 3:
        printf("\nEnter position betwee 1 to %d:",top+1);
        scanf("%d",&p);
        e=peep(p);
        if(e!=0)
                printf("\n no is %d",e);
        break;
case 4:
        printf("\nEnter position:");
        scanf("%d",&p);
        printf("Enter data to change:");
        scanf("%d",&e);
        update(p,e);
        break;
case 5:
        disp();
        break;
case 6:
        printf("\nBYE BYE");
        break;
default:
```

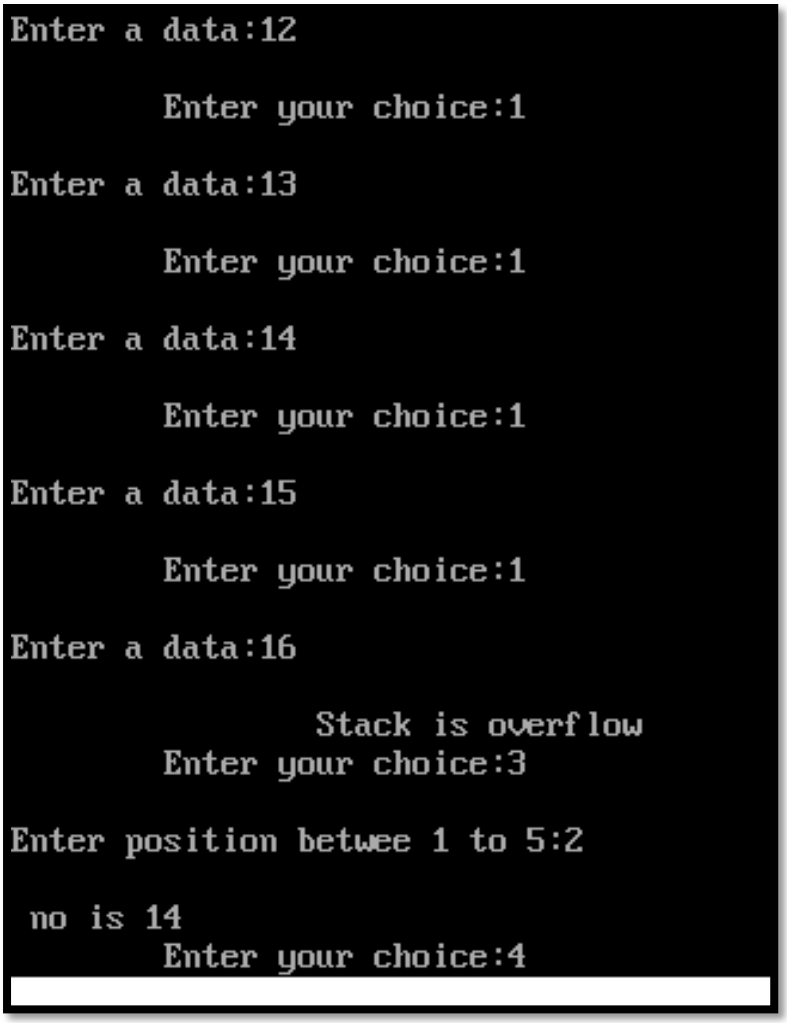
```
                printf("\nPLEASE ENTER VALID CHOICE");
        }
        }while(ch!=6);
getch();
}
```

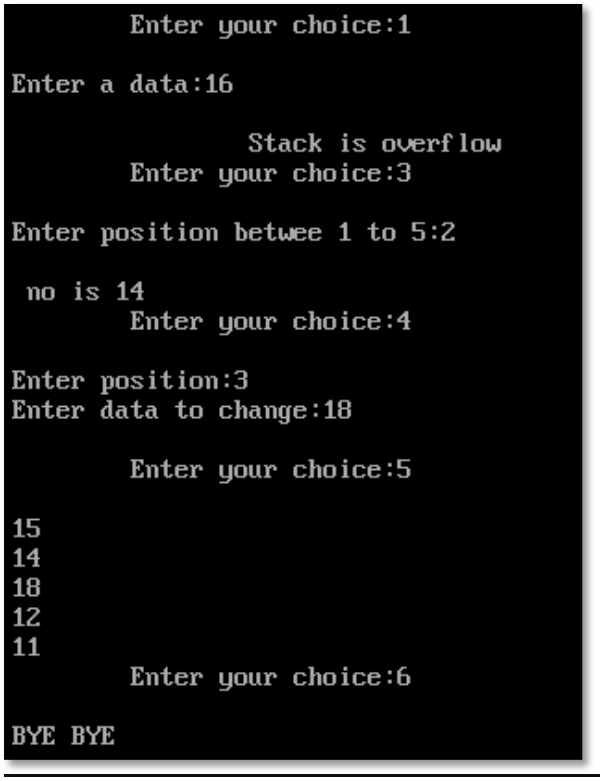
OUTPUT:-




Enter a data:50
Enter your choice:1
Enter a data:60
Stack is overflow
Enter your choice:5
50
40
30
20
10
Enter your choice:2
Deleted no is 50
Enter your choice:5
40
30
20
10
Enter your choice:


Enter your choice:2
Deleted no is 40
Enter your choice:2
Deleted no is 30
Enter your choice:2
Deleted no is 20
Enter your choice:5
10
Enter your choice:2
Deleted no is 10
Enter your choice:2
Stack is underflow
Enter your choice:1
Enter a data:11
Enter your choice:1
Enter a data:12


Enter a data:12
Enter your choice:1
Enter a data:13
Enter your choice:1
Enter a data:14
Enter your choice:1
Enter a data:15
Enter your choice:1
Enter a data:16
Stack is overflow
Enter your choice:3
Enter position betwee 1 to 5:2
no is 14
Enter your choice:4



Question (2)

//write a program to reverse the string using stack.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#define n 20
int top=-1;
char s[n];
void push(char data)
{
        if (top==(n-1))
        {
```

```
            printf("\n stack is overflow :" );
            return ;
        }
        top++;
        s[top]=data;
}
char pop()
{
        char data;
        if(top==-1)
        {
            printf("\n Stack is underflow :");
            return 0;
        }
        data=s[top];
        top--;
        return data;
}
void main()
{
        char r[n],str[n];
        int i;
        clrscr();
        printf("\n Enter string :-");
        gets(str);
```

```
        for(i=0;i<strlen(str);i++)
        push(str[i]);
        for(i=0;i<strlen(str);i++)
        r[i]=pop();
        r[i]='\0';
        puts(r);
        getch();
}
```

OUTPUT:-

Question (3)

//write a program to convert infix into postfix expression using stack.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#define n 20
int top=-1;
char s[n];
char infix[n],postfix[n];
```

```
void push(char data)
{
        if (top==(n-1))
        {
                printf("\n stack overflow");
                return ;
        }
        top++;
        s[top]=data;
}
char pop()
{
        char data;
        if(top==-1)
        {
                printf("\n stack underflow");
                return 0;
        }
        data=s[top];
        top--;
        return data;
}
void main()
{
        int i,j=-1;
```

```
clrscr();
printf("\nEnter infix expression : ");
scanf("%s",infix);
for(i=0;i<strlen(infix);i++)
{
        switch (infix[i])
        {
                case '(':
                        push(infix[i]);
                        break;
                case ')':
                        while (s[top]!='(')
                        postfix[++j]=pop();
                        pop();
                        break ;
                case '^' :
                        while(s[top]=='^')
                        postfix[++j]=pop();
                        push(infix[i]);
                        break ;
                case '*' :
                case '/' :
                        while((s[top]=='^')||(s[top]=='*')||(s[top]=='/'))
                        postfix[++j]=pop();
                        push(infix[i]);
```

```
                    break ;
                case '+' :
                case '-' :

        while((s[top]=='^')||(s[top]=='*')||(s[top]=='/')||(s[top]=='+')||(s[top]=='-'))
                    postfix[++j]=pop();
                    push(infix[i]);
                    break ;
                default :
                    postfix[++j]=infix[i];
                    break ;
            }

        }
        while(top!=-1)
        postfix[++j]=pop();
        postfix[++j]='\0';
        printf("postfix expression is %s",postfix);
        getch();
}
```

OUTPUT:-

Question (4)

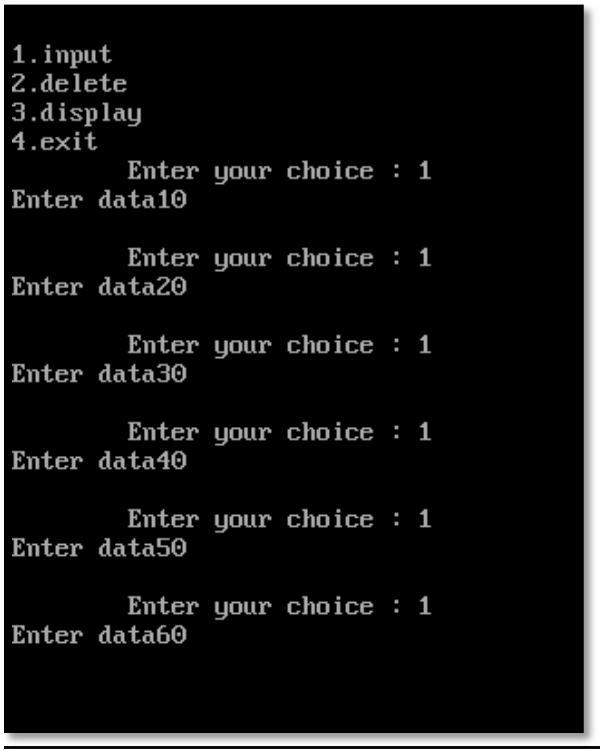
//write a menu driven program to implment simple queue with all opertion.

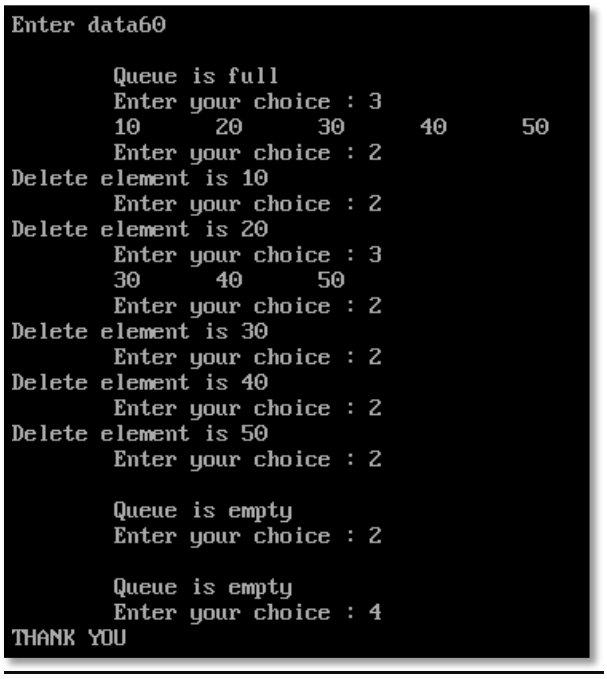
```
#include<stdio.h>
#include<conio.h>
#define n 5
int q[n];
int f=-1,r=-1;
void q_insert(int data)
{
	if(r==(n-1))
	{
		printf("\n\tQueue is full");
		return;
	}
	r++;
	q[r]=data;
	if(f==-1)
		f=0;
}
int q_delete()
{
	int data;
	if(f==-1)
	{
		printf("\n\tQueue is empty");
		return 0;
	}
```

```
        data=q[f];
        q[f]=0;
        if(f==r)
                f=r=-1;
        else
                f++;
        return data;
}
void display()
{
        int i;
        if(f==-1)
        {
                printf("\tQueue is empty");
                return;
        }
        for(i=f;i<=r;i++)
                printf("\t%d",q[i]);
}
void main()
{
        int ch,e;
        clrscr();
        printf("\n1.input");
        printf("\n2.delete");
        printf("\n3.display");
        printf("\n4.exit");
        do
```

```
{
        printf("\n\tEnter your choice : ");
        scanf("%d",&ch);
        switch(ch)
        {
                case 1:
                        printf("Enter data");
                        scanf("%d",&e);
                        q_insert(e);
                        break;
                case 2:
                        e=q_delete();
                        if(e!=0)
                                printf("Delete element is %d",e);
                        break;
                case 3:
                        display();
                        break;
                case 4:
                        printf("THANK YOU");
                        break;
                default:
                        printf("WRONG DATA");
                        break;
                }
        }while(ch!=4);
    getch();
}
```

OUTPUT:-


1.input
2.delete
3.display
4.exit
Enter your choice : 1
Enter data10
Enter your choice : 1
Enter data20
Enter your choice : 1
Enter data30
Enter your choice : 1
Enter data40
Enter your choice : 1
Enter data50
Enter your choice : 1
Enter data60


Enter data60
Queue is full
Enter your choice : 3
10 20 30 40 50
Enter your choice : 2
Delete element is 10
Enter your choice : 2
Delete element is 20
Enter your choice : 3
30 40 50
Enter your choice : 2
Delete element is 30
Enter your choice : 2
Delete element is 40
Enter your choice : 2
Delete element is 50
Enter your choice : 2
Queue is empty
Enter your choice : 2
Queue is empty
Enter your choice : 4
THANK YOU

Question (5)

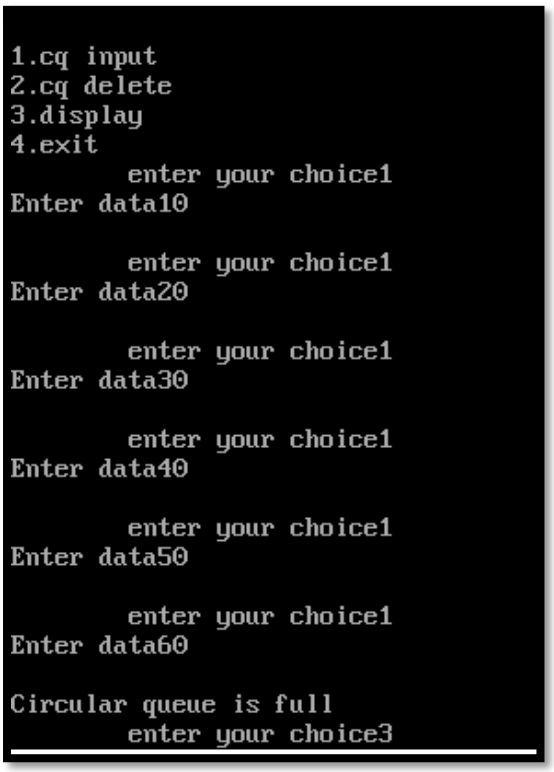
```
//write a menu driven program to implment circular queue with all opertion.
#include<stdio.h>
#include<conio.h>
#define n 5
int q[n];
int f=-1,r=-1;
void cq_insert(int data)
{
        if((r==(n-1)&&f==0)||(r+1)==f)
        {
                printf("\nCircular queue is full");
                return;
        }
        if(r==(n-1))
                r=0;
        else
                r++;
        q[r]=data;
        if(f==-1)
                f=0;
}
int cq_delete()
{
        int data;
        if(f==-1)
```

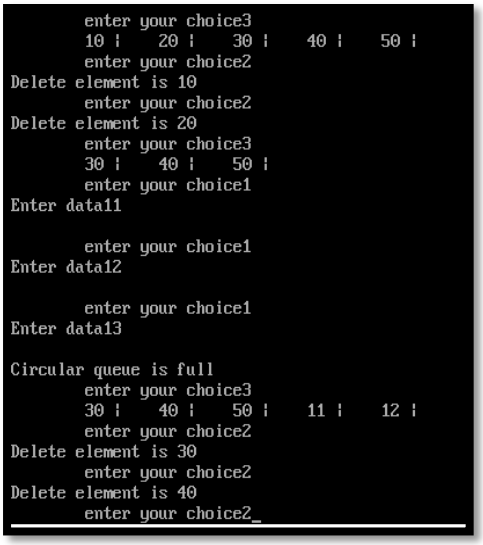
```
	{
		printf("Queue is empty");
		return 0;
	}
	data=q[f];
	q[f]=0;
	if(f==r)
		f=r=-1;
	else if(f==(n-1))
		f=0;
	else
		f++;
	return data;
}
void display()
{
	int i;
	if(r>f)
	{
		for(i=f;i<=r;i++)
			printf("\t%d |",q[i]);
	}
	else
	{
		for(i=f;i<=n-1;i++)
			printf("\t%d |",q[i]);
		for(i=0;i<=r;i++)
			printf("\t%d |",q[i]);
```

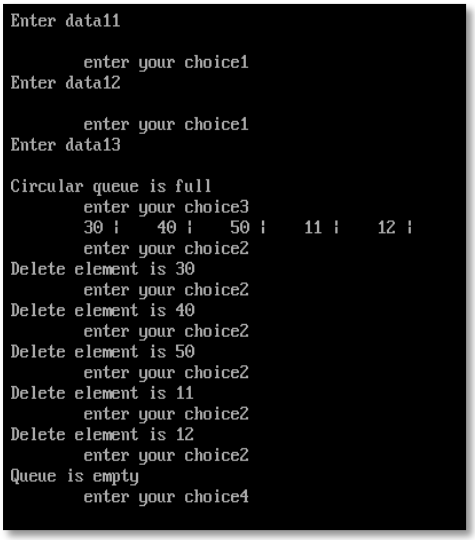
```
        }


}
void main()
{
        int ch,e;
        clrscr();
        printf("\n1.cq input");
        printf("\n2.cq delete");
        printf("\n3.display");
        printf("\n4.exit");
        do
        {
                printf("\n\tenter your choice");
                scanf("%d",&ch);
                switch(ch)
                {
                        case 1:
                                printf("Enter data");
                                scanf("%d",&e);
                                cq_insert(e);
                                break;
                        case 2:
                                e=cq_delete();
                                if(e!=0)
                                        printf("Delete element is %d",e);
                                break;
                        case 3:
```

```
                display();
                break;
            case 4:
                break;
            default:
                printf("TRY AGAIN");
                break;
            }
        }while(ch!=4);
    getch();
}
```

Output:-







Question (6)

//Write a menu driven program to implement output restricted D-queue with all operation.

```
#include<stdio.h>

#include<conio.h>

#define n 5

int q[n];

int f=-1,r=-1;

void dq_insert_right(int data)

{

        if(r==(n-1))

        {

                printf("\nD-queue is full from right");

                return;

        }

        r++;

        q[r]=data;

        if(f==-1)

                f=0;
```

```
}
void dq_insert_left(int data)
{
        if(f==0)
        {
                printf("\nD-queue is full from left");
                return;
        }
        if(f==-1)
        {
                f=n-1;
                r=n-1;
        }
        else
                f--;
        q[f]=data;

}
int dq_delete_left()
{
        int data;
        if(f==-1)
        {
                printf("Queue is empty");
                return 0;
        }
        data=q[f];
        q[f]=0;
```

```
        if(f==r)
                f=r=-1;
        else if(f==(n-1))
                f=0;
        else
                f++;
        return data;
}
int dq_delete_right()
{
        int data;
        if(r==-1)
        {
                printf("Queue is empty");
                return 0;
        }
        data=q[r];
        q[r]=0;
        r=r-1;
        if(f>r)
        {
                r=f=-1;
        }
        return data;
}
void display()
{
        int i;
```

```
        for(i=f;i<=n-1;i++)
        {
                if(q[i]!=0)
                {
                        printf("\t%d |",q[i]);
                }
        }
}
void main()
{
        int ch,e,ich;
        a:
        clrscr();
        printf("\n1.insert from right");
        printf("\n2.insert from left");
        printf("\n3.exit");
        printf("\nenter your choice:");
        scanf("%d",&ich);

        if(ich==3)
                goto b;
        do
        {
        switch(ich)
        {
                case 1:
                        printf("\n1.insert from right");
                        printf("\n2.delete from right");
```

```
            printf("\n3.display");
            printf("\n4.exit");
            printf("\nenter your choice:");
            scanf("%d",&ch);
            switch(ch)
            {
                  case 1:
                        printf("Enter element");
                        scanf("%d",&e);
                        dq_insert_right(e);
                        break;
                  case 2:
                        e=dq_delete_right();
                        printf("Deleted element is %d",e);
                        break;
                  case 3:
                        display();
                        break;
                  case 4:
                        ch=5;
                        break;
            }
            break;
      case 2:
            printf("\n1.insert from left");
            printf("\n2.delete from right");
            printf("\n3.display");
            printf("\n4.exit");
```

```
            printf("\nenter your choice:");
            scanf("%d",&ch);
            switch(ch)
            {
                  case 1:
                        printf("enter element");
                        scanf("%d",&e);
                        dq_insert_left(e);
                        break;
                  case 2:
                        e=dq_delete_right();
                        printf("deleted element is %d",e);
                        break;
                  case 3:
                        display();
                        break;
                  case 4:
                        ch=5;
                        break;
            }
            break;
case 3:
            break;
}
if((f==-1)&&(r==-1))
{
            printf("\nQueue is empty");
            goto a;
```

```
                }
        }while(ch!=5);
        b:
        getch();
}
```

Output:-

1. Insert from right
2. Insert from left
3. Exit

Enter your choice:1

1. Insert from right
2. Delete from right
3. Display
4. Exit

Enter your choice :1
Enter element : 10
Enter your choice :1
Enter element :20
Enter your choice :1
Enter element :30
Enter your choice :1
Enter element :40
Enter your choice :1
Enter element :50
Enter your choice :1
Enter element :60

D-queue is full from right

Enter your choice :3

10|20|30|40|50|

Enter your choice :2

Deleted element 50

10|20|30|40|

Enter your choice :2

Deleted element 40

Enter your choice :2

Deleted element 30

Enter your choice :2

Deleted element 20

Enter your choice :2

Deleted element 10

Enter your choice :2

1.Insert from right

2.Insert from left

3.Exit

Enter your choice: 2

1.Insert from left

2.Delete from left

3.Display

4.Exit

Enter your choice :1

Enter element : 10
Enter your choice :1
Enter element :20
Enter your choice :1
Enter element :30
Enter your choice :1
Enter element :40
Enter your choice :1
Enter element :50
Enter your choice :1
Enter element :60

D-queue is full from left

Enter your choice :3

10|20|30|40|50|

Enter your choice :2

Deleted element 10

20|30|40|50|

Enter your choice :2

Deleted element 20

Enter your choice :2

Deleted element 30

Enter your choice :2

Deleted element 40

Enter your choice :2

Deleted element 50

Enter your choice :2

Question (7)

//Write a menu driven program to implement circular singly link list with all operation.

```
#include<stdio.h>
 #include<conio.h>
 #include<alloc.h>

 struct node
 {
        int data;
        struct node *next;
}*head,*q,*newnode,*last,*temp;

 void create()
 {
        char ch='y';
        int e;
        head=NULL;
        do
        {
                printf("\nEnter data:");
                flushall();
                scanf("%d",&e);

                newnode=(struct node*)malloc(sizeof(struct node));

                newnode->data=e;
                if(head==NULL)
```

```
                head=newnode;
        else
                {
                        q->next=newnode;
                }
                q=newnode;
                printf("\nDo you want to continue?\nEnter your choice:");
                flushall();
                scanf("%c",&ch);
    }while(ch=='y');
    newnode->next=head;
    last=newnode;
}
void display()
{
    q=head;
    do
    {
        printf("\n\t|%d|%d|",q->data,q->next);
        q=q->next;
    }while(q!=head);
}
void add_begin()
{
    int e;
    printf("\nEnter data:");
    scanf("%d",&e);
    newnode=(struct node*)malloc(sizeof(struct node));
```

```
        newnode->data=e;
        last->next=newnode;
        newnode->next=head;
        head=newnode;
}
void add_end()
{
        int e;
        printf("\nEnter data:");
        scanf("%d",&e);
        newnode=(struct node*)malloc(sizeof(struct node));
        newnode->data=e;
        last->next=newnode;
        newnode->next=head;
        last=newnode;
}
void add_middle()
{
        int p,e,i;
        printf("\nEnter position:");
        scanf("%d",&p);
        printf("\nEnter data:");
        scanf("%d",&e);
        newnode=(struct node*)malloc(sizeof(struct node));
        newnode->data=e;
        q=head;
        for(i=1;i<=p-2;i++)
        {
```

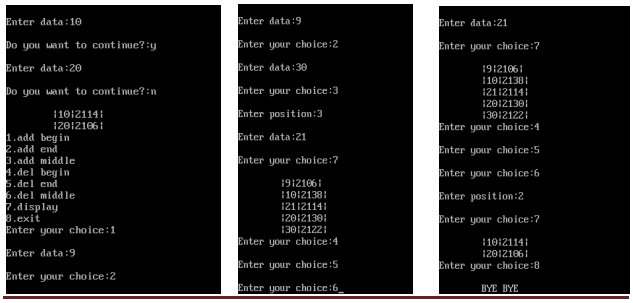
```
                q=q->next;
        }
        newnode->next=q->next;
        q->next=newnode;
}
void del_begin()
{
        temp=head;
        head=head->next;
        last->next=head;
        free(temp);
}
void del_end()
{
        q=head;
        do
        {
                q=q->next;
        }while(q->next->next!=head);
        temp=q->next;
        q->next=head;
        last=q;
        free(temp);
}
void del_middle()
{
        int i,p;
        printf("\nEnter position:");
```

```
        scanf("%d",&p);
        q=head;
        for(i=1;i<=p-2;i++)
        {
                q=q->next;
        }
        temp=q->next;
        q->next=q->next->next;
        free(temp);
}

void main()
{
        int ch;
        clrscr();
        create();
        display();
        do
        {
                printf("\n1.add begin");
                printf("\n2.add end");
                printf("\n3.add middle");
                printf("\n4.del begin");
                printf("\n5.del end");
                printf("\n6.del middle");
                printf("\n7.display");
                printf("\n8.exit");
                printf("\nEnter your choice:");
```

```
scanf("%d",&ch);
switch(ch)
{
        case 1:
                add_begin();
                break;
        case 2:
                add_end();
                break;
        case 3:
                add_middle();
                break;
        case 4:
                del_begin();
                break;
        case 5:
                del_end();
                break;
        case 6:
                del_middle();
                break;
        case 7:
                display();
                break;
        case 8:
                printf("\nBYE BYE");
                break;
}
```

```
    }while(ch!=8);
    getch();
}
```

Output:-



Question (8)

//Write a menu driven program to implement doubly link list with all operation.

```
#include<stdio.h>
#include<conio.h>
#include<alloc.h>

struct node
{
    int data;
    struct node *next;
    struct node *prev;
}*head,*q,*newnode,*last,*temp;

void create()
{
    char ch='y';
    int e;
```

```
        head=NULL;
        do
        {
                printf("\nEnter data:");
                flushall();
                scanf("%d",&e);

                newnode=(struct node*)malloc(sizeof(struct node));

                newnode->data=e;
                newnode->prev=NULL;
                if(head==NULL)
                        head=newnode;
                else
                        {
                                q->next=newnode;
                                newnode->prev=q;
                        }

                        q=newnode;
                        printf("\nDo you want to continue?:");
                        flushall();
                        scanf("%c",&ch);
        }while(ch=='y');
        newnode->next=NULL;
        last=newnode;
}
void display1()
```

```
{
        q=head;
        do
        {
                printf("\n\t|%d|%d|",q->data,q->next);
                q=q->next;
        }while(q!=NULL);
}
void display2()
{
        q=last;
        do
        {
                printf("\n\t|%d|%d|",q->data,q->next);
                q=q->prev;
        }while(q!=NULL);
}
void add_begin()
{
        int e;
        printf("\nEnter data:");
        scanf("%d",&e);
        newnode=(struct node*)malloc(sizeof(struct node));
        newnode->data=e;
        newnode->prev=NULL;
        newnode->next=head;
        head->prev=newnode;
        head=newnode;
```

```
}
void add_end()
{
        int e;
        printf("\nEnter data:");
        scanf("%d",&e);
        newnode=(struct node*)malloc(sizeof(struct node));
        newnode->data=e;
        newnode->next=NULL;
        last->next=newnode;
        newnode->prev=last;
        last=newnode;
}
void add_middle()
{
        int p,e,i;
        printf("\nEnter position:");
        scanf("%d",&p);
        printf("\nEnter data:");
        scanf("%d",&e);
        newnode=(struct node*)malloc(sizeof(struct node));
        newnode->data=e;
        q=head;
        for(i=1;i<=p-2;i++)
                q=q->next;
        newnode->next=q->next;
        newnode->prev=q;
        q->next->prev=newnode;
```

```
        q->next=newnode;
}
void del_begin()
{
        temp=head;
        head=head->next;
        head->prev=NULL;
        free(temp);
}
void del_end()
{
        temp=last;
        last=last->prev;
        last->next=NULL;
        free(temp);
}
void del_middle()
{
        int i,p;
        printf("\nEnter position:");
        scanf("%d",&p);
        q=head;
        for(i=1;i<=p-2;i++)
        {
                q=q->next;
        }
        temp=q->next;
        q->next=q->next->next;
```

```
        q->next->prev=q;
        free(temp);
}

void main()
{
        int ch;
        clrscr();
        create();
        display1();
        do
        {
                printf("\n1.add begin");
                printf("\n2.add end");
                printf("\n3.add middle");
                printf("\n4.del begin");
                printf("\n5.del end");
                printf("\n6.del middle");
                printf("\n7.display F");
                printf("\n8.display R");
                printf("\n9.exit");
                printf("\nEnter your choice:");
                scanf("%d",&ch);
                switch(ch)
                {
                        case 1:
                                add_begin();
                                break;
```

```
                case 2:
                        add_end();
                        break;
                case 3:
                        add_middle();
                        break;
                case 4:
                        del_begin();
                        break;
                case 5:
                        del_end();
                        break;
                case 6:
                        del_middle();
                        break;
                case 7:
                        display1();
                        break;
                case 8:
                        display2();
                        break;
                case 9:
                        printf("\nBYE BYE");
                        break;

        }
}while(ch!=9);
getch();
```

}

Output:-

Enter Data: 5

Do you want to continue?: y

Enter Data: 6

Do you want to continue?: y

Enter Data: 7

Do you want to continue?: n

|5 |2140|

|6|2150|

|7|0|

1.add begin

2.add end

3.add middle

4.del begin

5.del end

6.del middle

7.display F

8.display R

9.exit

Enter your choice:1

Enter Data: 0

Enter your choice:1

Enter Data: 9

Enter your choice:7

|9|2160|

|0|2130|

|5|2140|

|6|2150|

|7|0|

Enter your choice:2

Enter Data: 10

Enter your choice:8

|10|0|

|7|2180|

|6|2150|

|5|2140|

|0|2130|

|9|2160|

Enter your choice:3

Enter position:2

Enter Data: 15

Enter your choice:7

|9|2190|

|15|2160|

|0|2130|

|5|2140|

|6|2150|

|7|2180|

|10|0|

Enter your choice:4

Enter your choice:5

Enter your choice:6

Enter position :5

Enter your choice:7

|15|2160|

|0|2130|

|5|2140|

|6|0|

Enter your choice:1

BYE BYE

Question (9)

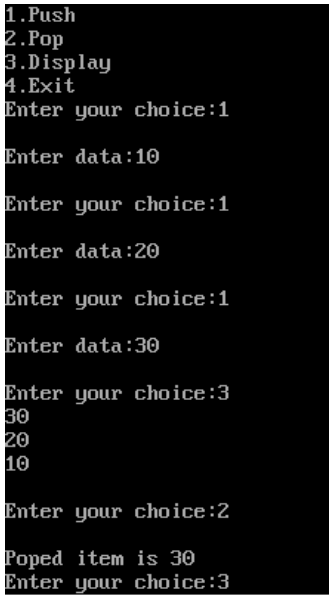
```
//write a program to implement dynamic stack.
#include<stdio.h>
 #include<conio.h>
 #include<alloc.h>

 struct node
{
        int data;
        struct node *next;
}*newnode,*head=NULL;
void push()
{
        int e;
        newnode=(struct node*)malloc(sizeof(struct node));
        printf("\nEnter data:");
        scanf("%d",&e);
        newnode->data=e;
        newnode->next=NULL;
        if(head==NULL)
        {
                head=newnode;
        }
        else
        {
                newnode->next=head;
```

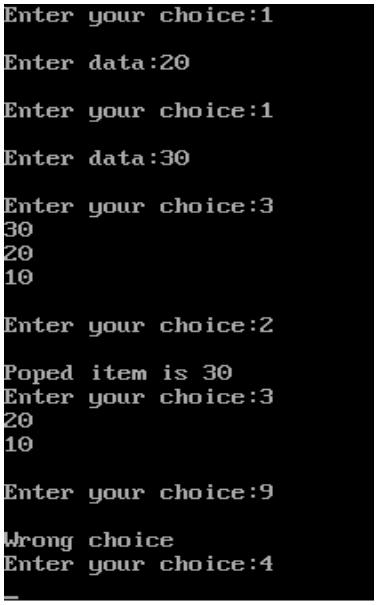
```
            head=newnode;
        }
}
void pop()
{
        struct node *tmp;
        if(head==NULL)
        {
            printf("\nStack is underflow");
        }
        else
        {
            tmp=head;
            printf("\nPoped item is %d",tmp->data);
            head=head->next;
            free(tmp);
        }
}
void display()
{
        struct node *tmp;
        tmp=head;
        if(head==NULL)
        {
            printf("\nStack is empty");
```

```
        }
        else
        {
                while(tmp!=NULL)
                {
                        printf("%d\n",tmp->data);
                        tmp=tmp->next;
                }
        }
}
void main()
{
        int ch;
        clrscr();
        do
        {
                printf("\n1.Push\n2.Pop\n3.Display\n4.Exit");
                printf("\nEnter your choice:");
                scanf("%d",&ch);
                switch(ch)
                {
                        case 1:
                                push();
                                break;
                        case 2:
```

```
                pop();
                break;
            case 3:
                display();
                break;
            case 4:
                break;
            default:
                printf("\nWrong choice");
        }
    }while(ch!=4);
    getch();
}
```

Output:-





Question (10)

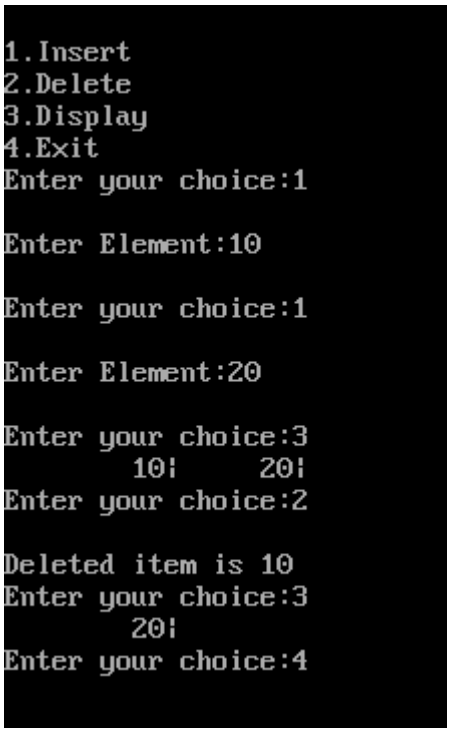
```
//write a program to implement dynamic queue
#include<stdio.h>
 #include<conio.h>
 #include<alloc.h>

struct node
{
        int data;
        struct node *next;
}*front=NULL,*rear=NULL,*newnode;
void q_insert()
{
        int e;
        newnode=(struct node*)malloc(sizeof(struct node));
        printf("\nEnter Element:");
        scanf("%d",&e);
        newnode->data=e;
        newnode->next=NULL;
        if(front==NULL)
                front=newnode;
        else
                rear->next=newnode;
        rear=newnode;
}
```

```
void q_delete()
{
        struct node *tmp;
        if(front==NULL)
                printf("\nQueue is empty");
        else
        {
                tmp=front;
                printf("\nDeleted item is %d",tmp->data);
                front=front->next;
                free(tmp);
        }
}
void display()
{
        struct node *tmp;
        tmp=front;
        if(front==NULL)
                printf("\nQueue is empty");
        else
        {
                while(tmp!=NULL)
                {
                        printf("\t%d|",tmp->data);
                        tmp=tmp->next;
```

```
            }
        }
}
void main()
{
        int ch;
        clrscr();
        do
        {
            printf("\n1.Insert\n2.Delete\n3.Display\n4.Exit");
            printf("\nEnter your choice:");
            scanf("%d",&ch);
            switch(ch)
            {
                case 1:
                    q_insert();
                    break;
                case 2:
                    q_delete();
                    break;
                case 3:
                    display();
                    break;
                case 4:
                    break;
```

```
                default:
                        printf("\nWrong choice");
            }
        }while(ch!=4);
getch();
}
```

Output:-

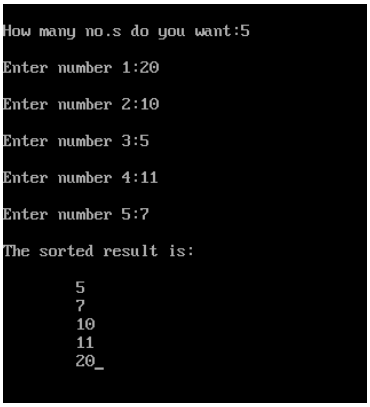


Question (11)

```
//write a program to implement insertion sort.
#include<stdio.h>
 #include<conio.h>
 #include<alloc.h>

 void insertion_sort(int a[],int n)
{
        int i,j,t;
        for(i=1;i<=n-1;i++)
        {
                t=a[i];
                j=i-1;
                while(t<a[j]&&j>=0)
                {
                        a[j+1]=a[j];
                        j--;
                }
                a[j+1]=t;
        }
        printf("\nThe sorted result is:\n");
        for(i=0;i<=n-1;i++)
        {
                printf("\n\t%d",a[i]);
        }
```

```
}

void main()
{
        int *a,n,i;
        clrscr();
        printf("\nHow many no.s do you want:");
        scanf("%d",&n);
        a=(int *)malloc(n * sizeof(int));
        for(i=0;i<=n-1;i++)
        {
                printf("\nEnter number %d:",i+1);
                scanf("%d",&a[i]);
        }
        insertion_sort(a,n);
        getch();
}
```

Output:-



Question (12)

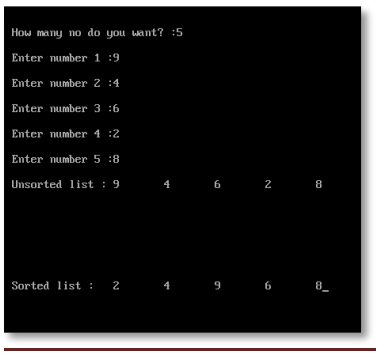
//write a program to implement quick sort.

```
#include<conio.h>
#include<stdio.h>
#include<alloc.h>
void quick_sort(int a[],int low,int up)
{
        int v,temp,l,r;
        a1:
                l=low;
                r=up-1;
                v=up;
                if(low>=up)
                {
                        return;
                }
                while(a[v]<=a[r])
                {
                        r=r-1;
                }
                while(a[v]>=a[l])
                {
                        l=l+1;
                }
                if(l<r)
```

```
                {
                        temp=a[l];
                        a[l]=a[r];
                        a[r]=temp;
                        goto a1;
                }
                else if(l>r)
                {
                        temp=a[l];
                        a[l]=a[v];
                        a[v]=temp;
                }
                printf("\n");
                quick_sort(a,low,l-1);
                quick_sort(a,l+1,up);
}
void display(int a[],int low,int up)
{
        int i;
        for(i=low;i<=up;i++)
        {
                printf("\t%d",a[i]);
        }
}
void main()
```

```
{
        int *a,n,i;
        clrscr();
        printf("\nHow many no do you want? :");
        scanf("%d",&n);
        a=(int *)malloc(n* sizeof(int));
        for(i=0;i<=n-1;i++)
        {
                printf("\nEnter number %d :",i+1);
                scanf("%d",&a[i]);
        }
        printf("\nUnsorted list :");
        display(a,0,n-1);
        quick_sort(a,0,n-1);
        printf("\n\n\nSorted list :");
        display(a,0,n-1);
        getch();
}
```

Output:-



Question (13)

//write a program to ceate singly link list and then perform sorting on node using selection sot.

```
#include<conio.h>
#include<stdio.h>
#include<alloc.h>
struct node
{
        int data;
        struct node *next;
}*head,*q,*newnode;
void create()
{
        int e;
        char ch;
        head=NULL;
        do
        {
                printf("\nEnter data :");
                scanf("%d",&e);
                newnode=(struct node*)malloc(sizeof(struct node));
                newnode->data=e;
                if(head==NULL)
                        head=newnode;
```

```
                else
                        q->next=newnode;
                q=newnode;
                printf("\ncontine :");
                flushall();
                scanf("%c",&ch);
        }while(ch=='y');
        q->next=NULL;
}
void disp()
{
        q=head;
        while(q!=NULL)
        {
                printf("\t%d",q->data);
                q=q->next;
        }
}

void selection_sort(struct node *h)
{
        struct node *t,*pos,*loc;
        int min,temp;
        t=h;
        while(h!=NULL)
```

```
        {
                min=h->data;
                t=h->next;
                pos=h;
                loc=h;
                while(t!=NULL)
                {
                        if(t->data<min)
                        {
                                min=t->data;
                                loc=t;
                        }
                        t=t->next;
                }
                temp=pos->data;
                pos->data=loc->data;
                loc->data=temp;
                h=h->next;
                }
                disp();
        //}
}
void main()
{
```

```
int ch;
clrscr();
printf("\n\t1.create\n\t2.display\n\t3.short\n\t4.exit");
do
{
        printf("\nEnter your choice :");
        scanf("%d",&ch);
        switch(ch)
        {
                case 1:
                        create();
                        break;
                case 2:
                        disp();
                        break;
                case 3:
                        selection_sort(head);
                        break;
                case 4:
                        printf("\n\n\t\tBYE BYE");
                        break;
                default:
                        printf("\n\tPlease Enter valid choice :");
        }
}while(ch!=4);
```

```
    getch();
}
```

Output:-

1. create

2. display

3. short

4. Exit

Enter your choice: 1

Enter data: 5

Continue: y

Enter your choice: 1

Enter data: 4

Continue: y

Enter your choice: 1

Enter data: 3

Continue: n

Enter your choice: 2

5 4 3

Enter your choice: 3

3 4 5

Enter your choice: 4

BYE BYE

Question (14)

//write a program to create two singly link list and then merge them and sort them.

```
#include<stdio.h>
 #include<conio.h>
 #include<alloc.h>

 struct node
{
        int data;
        struct node *next;
}*head,*head1,*head2,*q,*newnode;

 void create()
{
        char ch='y';
        int e;
        head=NULL;
        do
```

```
        {
                printf("\nEnter data:");
                scanf("%d",&e);

                newnode=(struct node*)malloc(sizeof(struct node));

                newnode->data=e;
                if(head==NULL)
                        head=newnode;
                else
                        q->next=newnode;
                q=newnode;
                printf("\ndo you want to continue?\nenter your choice:");
                flushall();
                scanf("%c",&ch);
        }while(ch=='y');
        newnode->next=NULL;
}
 void display()
 {
        q=head1;
        do
        {
                printf("\n\t|%d|%d|",q->data,q->next);
                q=q->next;
```

```
        }while(q!=NULL);
}
void merge()
{
        q=head1;
        while(q->next!=NULL)
        {
                q=q->next;
        }
        q->next=head2;
        printf("\nAfter merging\n");
        display();
}
void sort()
{
        int i,j,temp;
        struct node *t;
        q=head1;
        while(q!=NULL)
        {
                t=q->next;
                while(t!=NULL)
                {
                        if(t->data<q->data)
                        {
```

```
                        temp=q->data;
                        q->data=t->data;
                        t->data=temp;
                    }
                    t=t->next;
                }
                q=q->next;
        }
}
void main()
{
        int ch;
        clrscr();
        create();
        head1=head;
        display();
        create();
        head2=head;
        do
        {
                printf("\n1.merge");
                printf("\n2.sort");
                printf("\n3.display");
                printf("\n4.exit");
                printf("\nenter your choice:");
```

```
            scanf("%d",&ch);

            switch(ch)
            {
                case 1:
                    merge();
                    break;
                case 2:
                    sort();
                    break;
                case 3:
                    display();
                    break;
                case 4:
                    break;
            }
        }while(ch!=4);
        getch();
}
```

Output:-

```
Enter data:40

do you want to continue?
enter your choice:y

Enter data:30

do you want to continue?
enter your choice:n

        |40|1992|
        |30|0|
Enter data:20

do you want to continue?
enter your choice:y

Enter data:10

do you want to continue?
enter your choice:_
```

```
do you want to continue?
enter your choice:y

Enter data:10

do you want to continue?
enter your choice:n

1.merge
2.sort
3.display
4.exit
enter your choice:1

After merging

        |40|1992|
        |30|2000|
        |20|2008|
        |10|0|
1.merge
2.sort
3.display
4.exit
enter your choice:_
```

```
        |40|1992|
        |30|2000|
        |20|2008|
        |10|0|
1.merge
2.sort
3.display
4.exit
enter your choice:2

1.merge
2.sort
3.display
4.exit
enter your choice:3

        |10|1992|
        |20|2000|
        |30|2008|
        |40|0|
1.merge
2.sort
3.display
4.exit
enter your choice:
```

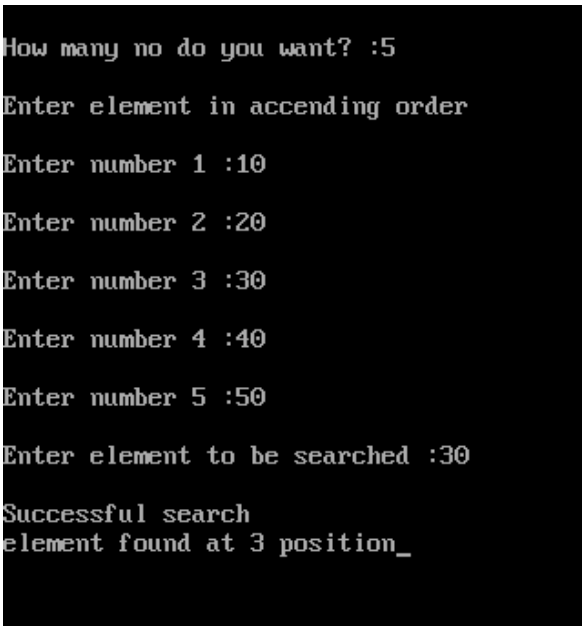
Question (15)

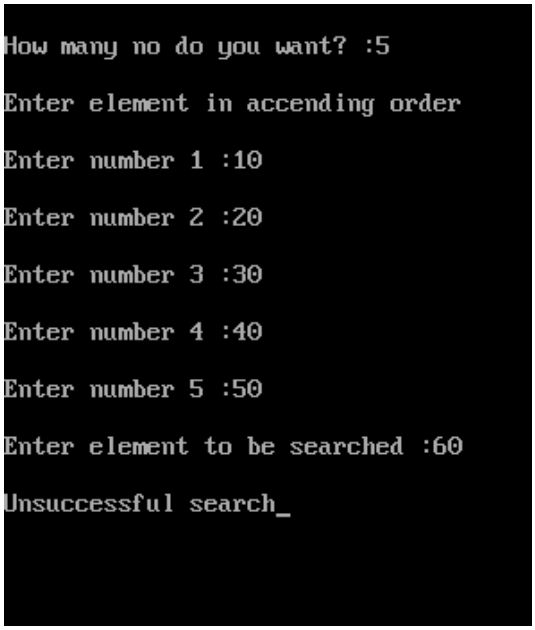
```
//write a program to implement binary search.
#include<conio.h>
#include<stdio.h>
#include<alloc.h>
void binary_search(int k[],int n,int x)
{
        int beg,mid,end;
        beg=0;
        end=n-1;
        mid=(beg+end)/2;
        while(beg<=end && x!=k[mid])
        {
                if(x>k[mid])
                {
                        beg=mid+1;
                }
                else
                {
                        end=mid-1;
                }
                mid=(beg+end)/2;
        }
        if(x==k[mid])
```

```
        {
                printf("\nSuccessful search \nelement found at %d position",mid+1);
        }
        else
        {
                printf("\nUnsuccessful search");
        }
}
void main()
{
        int n,i,*k,x;
        clrscr();
        printf("\nHow many no do you want? :");
        scanf("%d",&n);
        k=(int*)malloc((n)*sizeof(int));
        printf("\nEnter element in accending order\n");
        for(i=0;i<=n-1;i++)
        {
                printf("\nEnter number %d :",i+1);
                scanf("%d",&k[i]);
        }
        printf("\nEnter element to be searched :");
        scanf("%d",&x);
        binary_search(k,n,x);
        getch();
```

}

Output:-





Question (16)

//write a program to create two singly link list with following data empno, empname, dept and salary. Display the employee name and department whose salary is highest.

```
#include<stdio.h>
#include<conio.h>
#include<alloc.h>

struct node
{
        int empno,salary;
        char ename[30],dept[30];
        struct node *next;
```

```
}*head,*q,*newnode;
int count=0;

void create()
{
        char ch='y';
        int eno,sal;
        char name[30],edept[30];
        head=NULL;
        do
        {
                printf("\nEnter emp no:");
                scanf("%d",&eno);
                printf("\nEnter emp name:");
                scanf("%s",name);
                printf("\nEnter salary:");
                scanf("%d",&sal);
                printf("\nEnter department:");
                scanf("%s",edept);

                newnode=(struct node*)malloc(sizeof(struct node));
                newnode->empno=eno;
                strcpy(newnode->ename,name);
                newnode->salary=sal;
                strcpy(newnode->dept,edept);
```

```
            if(head==NULL)
            {
                head=newnode;
                count++;
            }
            else
            {
                q->next=newnode;
                count++;
            }
                q=newnode;
                printf("\nDo you want to continue?\nEnter your choice:");
                flushall();
                scanf("%c",&ch);
        }while(ch=='y');
            newnode->next=NULL;
}
void display()
{
        q=head;
        do
        {
            printf("\n|%d|%s|%d|%s",q->empno,q->ename,q->salary,q->dept);
            q=q->next;
```

```
        }while(q!=NULL);
}
void max()
{
        int i,max;
        char a[30];
        q=head;
        max=q->salary;
        for(i=1;i<=count;i++)
        {
                if(q->salary>max)
                {
                        max=q->salary;
                }
        q=q->next;
}
q=head;
do
{
        if(q->salary==max)
        {
                printf("\n\nName is %s",q->ename);
                printf("\nDepartment name is %s",q->dept);
        }
q=q->next;
```

```
}while(q!=NULL);
}
void main()
{
        clrscr();
        create();
        display();
        max();
        getch();
}
```

Output:-