CODE :

```
#include<stdio.h>
#include<stdlib.h>
#include<process.h>
struct node
{
        int info;
        struct node *llink;
        struct node *rlink;
        };
typedef struct node *NODE;
NODE getnode()
{
        NODE x;
        x=(NODE)malloc(sizeof(struct node));
        if(x==NULL)
        {
                printf("mem full\n");
                exit(0);
                }
```

```
        return x;
        }
void freenode(NODE x)
{
        free(x);
}
NODE dinsert_front(int item,NODE head)
{
NODE temp,cur;
temp=getnode();
temp->info=item;
cur=head->rlink;
head->rlink=temp;
temp->llink=head;
temp->rlink=cur;
cur->llink=temp;
return head;
}
NODE dinsert_rear(int item,NODE head)
{
```

```
NODE temp,cur;
temp=getnode();
temp->info=item;
cur=head->llink;
head->llink=temp;
temp->rlink=head;
temp->llink=cur;
cur->rlink=temp;
return head;
}
NODE ddelete_front(NODE head)
{
NODE cur,next;
if(head->rlink==head)
{
printf("dq empty\n");
return head;
}
cur=head->rlink;
next=cur->rlink;
```

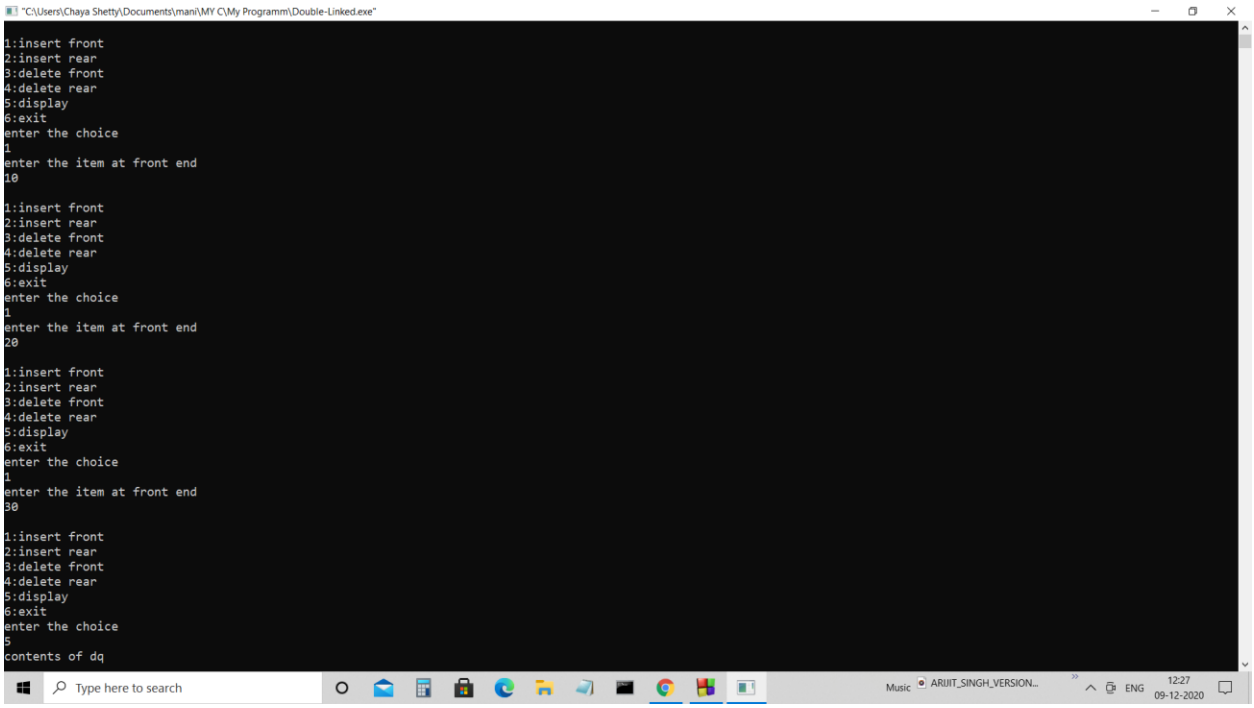
```
head->rlink=next;
next->llink=head;
printf("the node deleted is %d",cur->info);
freenode(cur);
return head;
}
NODE ddelete_rear(NODE head)
{
NODE cur,prev;
if(head->rlink==head)
{
printf("dq empty\n");
return head;
}
cur=head->llink;
prev=cur->llink;
head->llink=prev;
prev->rlink=head;
printf("the node deleted is %d",cur->info);
freenode(cur);
```

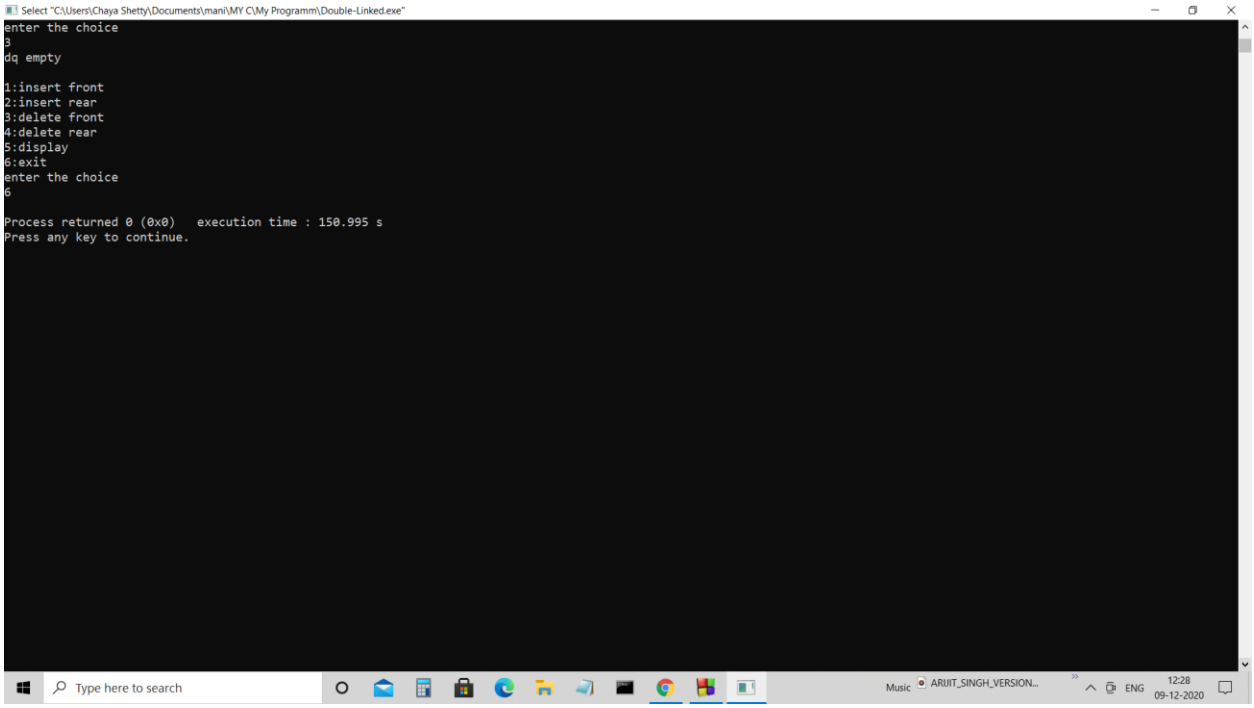
```
return head;
}
void display(NODE head)
{
NODE temp;
if(head->rlink==head)
{
printf("dq empty\n");
return;
}
printf("contents of dq\n");
temp=head->rlink;
while(temp!=head)
{
printf("%d\n",temp->info);
temp=temp->rlink;
}
printf("\n");
}
void main()
```

```
{
NODE head,last;
int item, choice;
head=getnode();
head->rlink=head;
head->llink=head;
for(;;)
{
        printf("\n1:insert front\n2:insert rear\n3:delete front\n4:delete
rear\n5:display\n6:exit\n");
        printf("enter the choice\n");
        scanf("%d",&choice);
        switch(choice)
        {
                case 1: printf("enter the item at front end\n");
                        scanf("%d",&item);
                        last=dinsert_front(item,head);
                        break;
                case 2: printf("enter the item at rear end\n");
                        scanf("%d",&item);
```

```
                last=dinsert_rear(item,head);
                break;
            case 3:last=ddelete_front(head);
                break;
            case 4: last=ddelete_rear(head);
                break;
            case 5: display(head);
                break;
            default:exit(0);
            }
        }
}
```

OUTPUT :


"C:\Users\Chaya Shetty\Documents\mani\MY C\My Programm\Double-Linked.exe"
1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:exit
enter the choice
1
enter the item at front end
10
1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:exit
enter the choice
1
enter the item at front end
20
1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:exit
enter the choice
1
enter the item at front end
30
1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:exit
enter the choice
5
contents of dq
Type here to search
Music
ARIJIT_SINGH_VERSION...
ENG
12:27
09-12-2020


Select "C:\Users\Chaya Shetty\Documents\mani\MY C\My Programm\Double-Linked.exe"
enter the choice
3
dq empty
1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:exit
enter the choice
6
Process returned 0 (0x0) execution time : 150.995 s
Press any key to continue.
Type here to search
Music
ARIJIT_SINGH_VERSION...
ENG
12:28
09-12-2020

CODE:

```
#include<stdio.h>
#include<stdlib.h>
#include<process.h>
struct node
 {
  int info;
  struct node *rlink;
  struct node *llink;
 };
typedef struct node *NODE;
NODE getnode()
{
NODE x;
x=(NODE)malloc(sizeof(struct node));
if(x==NULL)
 {
  printf("mem full\n");
  exit(0);
 }
```

```
 return x;
}
void freenode(NODE x)
{
free(x);
}
NODE insert_rear(NODE head,int item)
{
NODE temp,cur;
temp=getnode();
temp->rlink=NULL;
temp->llink=NULL;
temp->info=item;
cur=head->llink;
temp->llink=cur;
cur->rlink=temp;
head->llink=temp;
temp->rlink=head;
head->info=head->info+1;
return head;
```

```
}
NODE insert_leftpos(int item,NODE head)
{
NODE temp,cur,prev;
if(head->rlink==head)
{
printf("list empty\n");
return head;
}
cur=head->rlink;
while(cur!=head)
{
if(item==cur->info)break;
cur=cur->rlink;
}
if(cur==head)
{
 printf("key not found\n");
 return head;
 }
```

```
 prev=cur->llink;
 printf("enter towards left of %d=",item);
 temp=getnode();
 scanf("%d",&temp->info);
 prev->rlink=temp;
 temp->llink=prev;
 cur->llink=temp;
 temp->rlink=cur;
 return head;
}
NODE insert_righttpos(int item,NODE head)
{
NODE temp,cur,prev;
if(head->rlink==head)
{
printf("list empty\n");
return head;
}
cur=head->rlink;
while(cur!=head)
```

```
{
if(item==cur->info)break;
cur=cur->rlink;
}
if(cur==head)
{
 printf("key not found\n");
 return head;
 }
 prev=cur->rlink;
 printf("enter towards left of %d=",item);
 temp=getnode();
 scanf("%d",&temp->info);
 prev->llink=temp;
 temp->llink=cur;
 cur->rlink=temp;
 temp->rlink=prev;
 return head;
}
NODE delete_all_key(int item,NODE head)
```

```
{
NODE prev,cur,next;
int count;
   if(head->rlink==head)
    {
     printf("LE");
     return head;
     }
count=0;
cur=head->rlink;
while(cur!=head)
{
  if(item!=cur->info)
  cur=cur->rlink;
  else
 {
  count++;
  prev=cur->llink;
  next=cur->rlink;
  prev->rlink=next;
```

```
  next->llink=prev;
  freenode(cur);
  cur=next;
 }
}
if(count==0)
  printf("key not found");
else
  printf("key found at %d positions and are deleted\n", count);

return head;
}
void Search_info(int item,NODE head){
NODE cur;
if(head->rlink==head)
{
printf("list empty\n");
}
cur=head->rlink;
while(cur!=head)
```

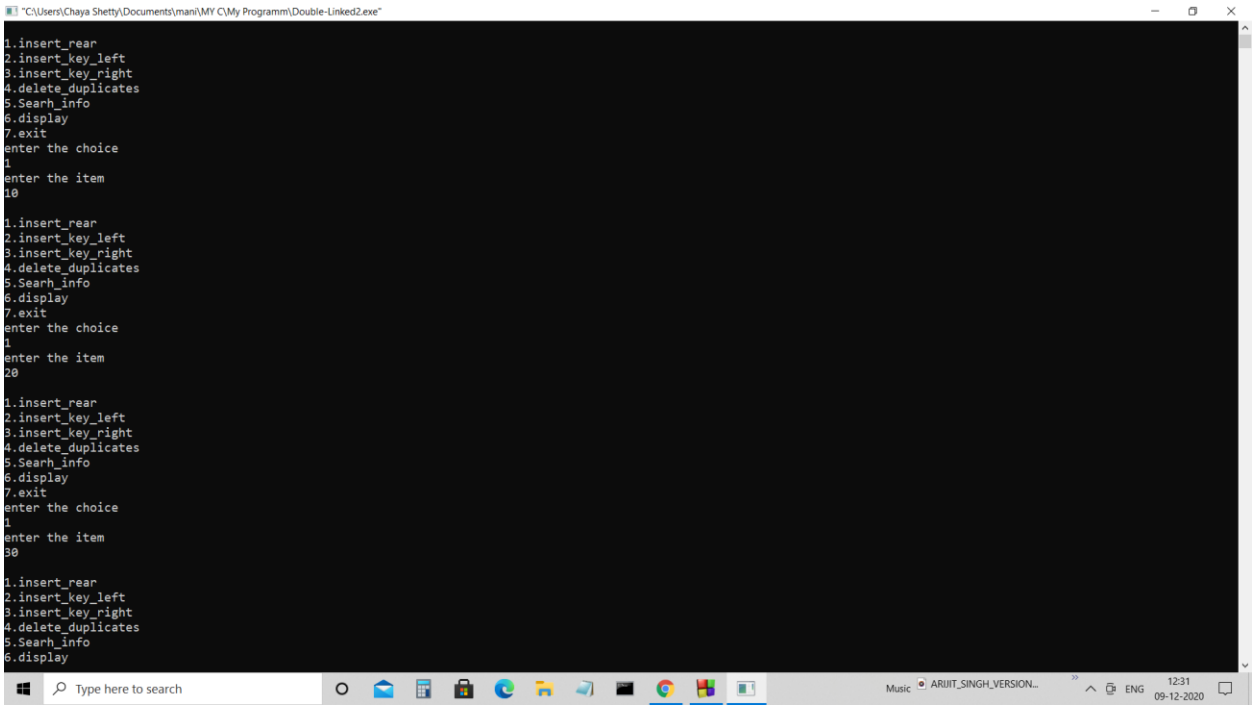
```
{
if(item==cur->info)
{
   printf("Search Successfull\n");
   break;
}
cur=cur->rlink;
}
if(cur==head)
{
 printf("Info not found\n");
 }
}
void display(NODE head)
{
NODE temp;
if(head->rlink==head)
{
printf("list empty\n");
return;
```

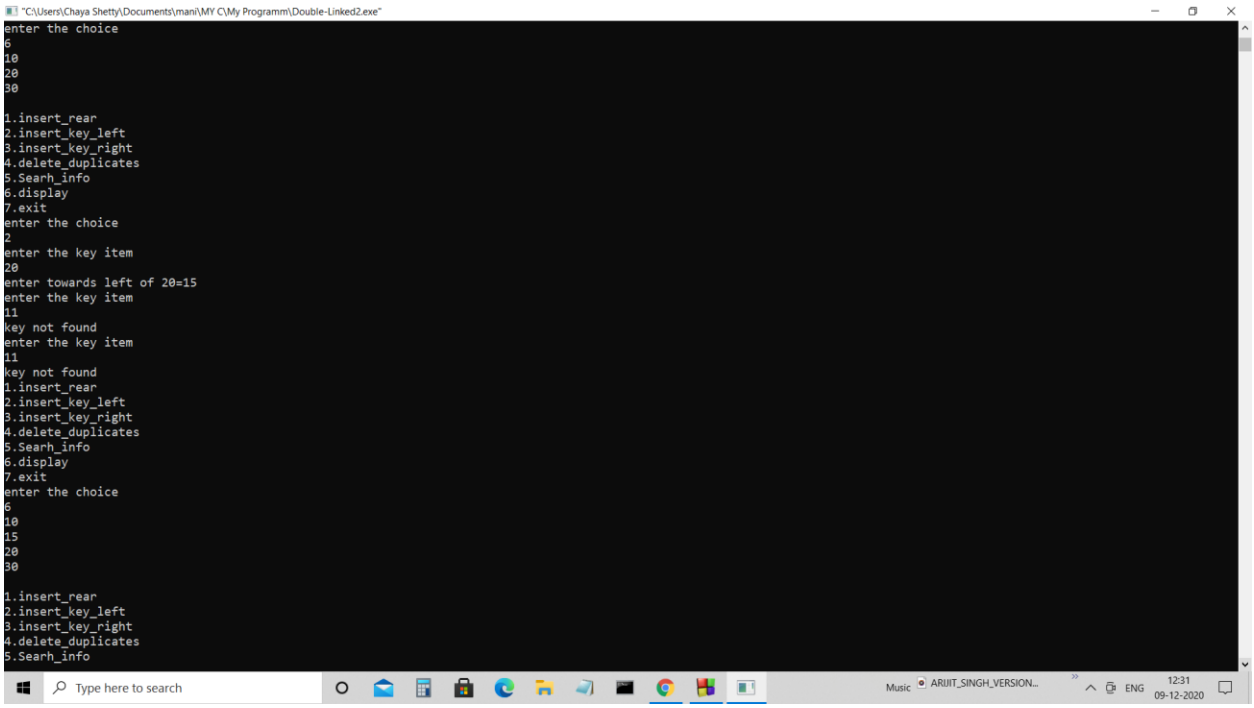
```
}
for(temp=head->rlink;temp!=head;temp=temp->rlink)
printf("%d\n",temp->info);
}
void main()
{
int item,choice,key;
NODE head;
head=getnode();
head->rlink=head;
head->llink=head;
for(;;)
{
printf("\n1.insert_rear\n2.insert_key_left\n3.insert_key_right\n4.delete_duplicates\n5.Searh_info\n6.display\n7.exit\n");
printf("enter the choice\n");
scanf("%d",&choice);
switch(choice)
 {
  case 1:printf("enter the item\n");
```

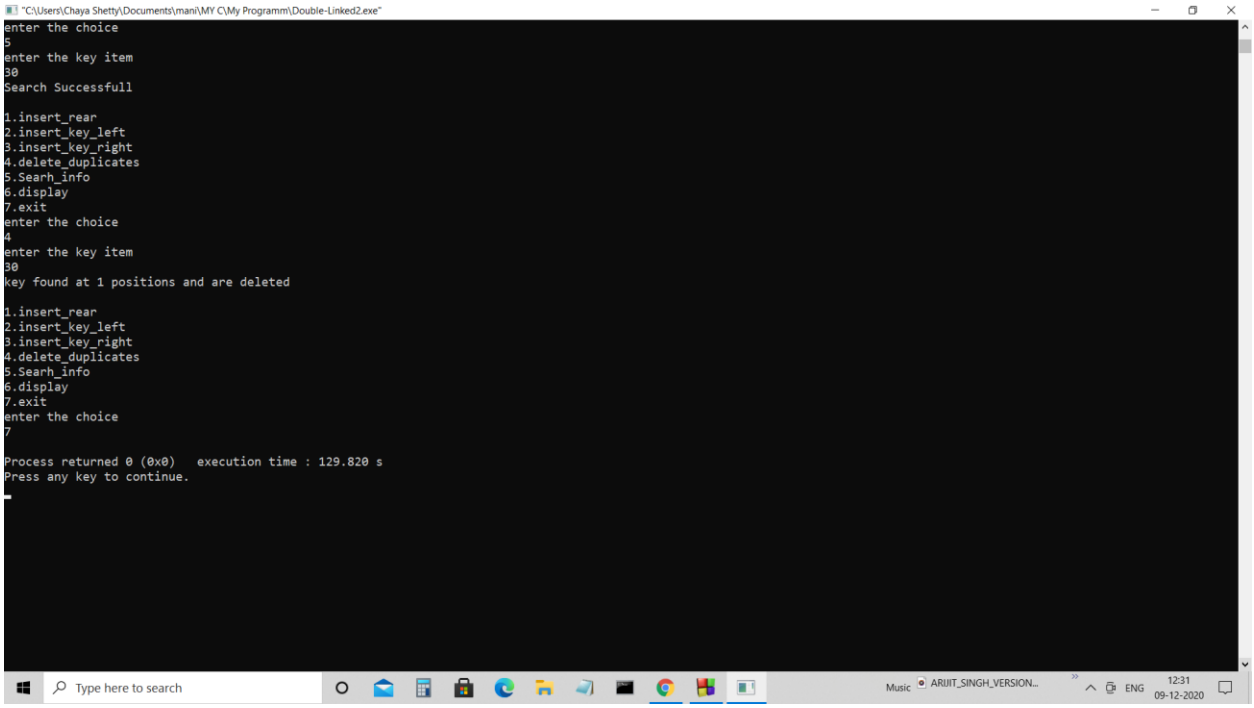
```
            scanf("%d",&item);
            head=insert_rear(head,item);
            break;
case 2:printf("enter the key item\n");
            scanf("%d",&item);
            head=insert_leftpos(item,head);
case 3:printf("enter the key item\n");
            scanf("%d",&item);
            head=insert_righttpos(item,head);
case 4:printf("enter the key item\n");
            scanf("%d",&item);
            head=delete_all_key(item,head);
            break;
case 5:printf("enter the key item\n");
            scanf("%d",&item);
            Search_info(item,head);
            break;
case 6:display(head);
            break;
default:exit(0);
```

```
                break;
  }
 }
}
```

OUTPUT:




"C:\Users\Chaya Shetty\Documents\mani\MY C\My Programm\Double-Linked2.exe"
enter the choice
6
10
20
30
1.insert_rear
2.insert_key_left
3.insert_key_right
4.delete_duplicates
5.Search_info
6.display
7.exit
enter the choice
2
enter the key item
20
enter towards left of 20=15
enter the key item
11
key not found
enter the key item
11
key not found
1.insert_rear
2.insert_key_left
3.insert_key_right
4.delete_duplicates
5.Search_info
6.display
7.exit
enter the choice
6
10
15
20
30
1.insert_rear
2.insert_key_left
3.insert_key_right
4.delete_duplicates
5.Search_info
Type here to search
Music
ARIJIT_SINGH_VERSION...
ENG
12:31
09-12-2020


"C:\Users\Chaya Shetty\Documents\mani\MY C\My Programm\Double-Linked2.exe"
enter the choice
5
enter the key item
30
Search Successfull
1.insert_rear
2.insert_key_left
3.insert_key_right
4.delete_duplicates
5.Search_info
6.display
7.exit
enter the choice
4
enter the key item
30
key found at 1 positions and are deleted
1.insert_rear
2.insert_key_left
3.insert_key_right
4.delete_duplicates
5.Search_info
6.display
7.exit
enter the choice
7
Process returned 0 (0x0)   execution time : 129.820 s
Press any key to continue.
Type here to search
Music
ARIJIT_SINGH_VERSION...
ENG
12:31
09-12-2020