**1.C program to implement double ended queue.**

#include<stdio.h>

#include<conio.h>

#include<time.h>

void insert\_f(int,int[],int\*,int\*)

void\_delete\_r(int[],int\*,int\*);

void Qinsert(int[],int,int,int\*);

void Qdelete(int[],int\*,int\*);

void Qdisplay(int[],int,int);

void main()

{

int opt=0,item,f=0,r=-1,\*Q,size;

clock\_t,start,end;

double cpu\_time\_used;

clrscr();

start=clock();

printf(“double ended queue\n’);

printf(“\n enter size of queue”);

scanf(“%d”,&size);

Q=(int\*)malloc(size\*sizeof(int));

while(opt<6)

{

printf(“\n DQUEUE\n”);

printf(“1.insertfront\n 2.insert rear\n 3.delete front\n 4.delete rear\n 5.display\n 6.exit\n”);

printf(“enter option\n”);

scanf(“%d”,&opt);

swich(opt)

{

case 1:printf(enter item to be inserted\n”);

scanf(%d”,&item);

insert\_f(item,Q,&f,&r);

break;

case 2:printf(enter item to be inserted\n”);

scanf(%d”,&item);

Qinsert(Q,item,size,&r);

break;

case 3:Qdelete(Q,&f,&r);

break;

case 4:delete\_r(Q,&f,&r);

break;

case 5:Qdisplay(Q,f,r);

break;

}

end=clock();

cpu\_time\_used=((double)(end-start));

printf(“\n cpu\_time\_used%f”,cpu\_time\_used);

}  
}

void insert\_f(int item,int Q[],int \*f,int \*r)

{

if(\*f>\*r)

{

\*r=\*r+1;

printf(“Q[%d]=%d’,Q[\*r],item);

Q[\*r]=item;

return;

}

if(\*f!=0)

{

\*f=\*f-1;

Q[\*f]=item;

return;

}

printf(“front insertion not possible”);

getch();

}

void delete\_r(int Q[],int \*f,int \*r)

{

int item;

if(\*f>\*r)

{

printf(“empty queue\n”);

getch();

return;

}

item=Q[\*r];

\*r=\*r-1;

if(\*f>\*r)

{

\*f=0;

\*r=-1;

}

printf(“deleted element %d”,item);

getch();

}

void Qinsert(int Q[],int,item,int size,int \*r)

{

if(\*r==size-1)

{

printf(“queue full”);

getch();

}

else

{

\*r=\*r+1;

Q[\*r]=item;

}  
}

void Qdelete(int Q[],int\*f,int\*r)

{

int item;

if(\*f>\*r)

{

printf(“queue empty”);

getch();

return;

}

item =q[\*f];

\*f=\*f+1;

print(“delete element is %d”,item);

gerch();

if(\*f>\*r)

{

f=0;

\*r=-1;

}

}

void Qdisplay(int Q[],int f,intr)

{

int i;

if(f>r)

printf(“empty queue”);

else

{

for(i=f;i<=r;i++);

{

printf(“Q[%d]\n”),i,Q[i]);

printf(“\n”);

}

}

getch();

}