Name :Dhanashri Rode

Roll No. 49

# include<stdio.h>

# define Size 3

int deque\_arr[Size];

int front = -1;

int rear = -1;

/\*Begin of main\*/

main()

{ int choice;

printf("1.Input restricted deque\n");

printf("2.Output restricted deque\n");

printf("Enter your choice : ");

scanf("%d",&choice);

switch(choice)

{

case 1 :

input\_que();

break;

case 2:

output\_que();

break;

default:

printf("Wrong choice\n");

}

}

/\*Begin of insert\_rear\*/

void insert\_rear()

{

int added\_item;

if((front == 0 && rear == Size-1) || (front == rear+1))

{ printf("Queue Overflow\n");

return;}

if (front == -1) /\* if queue is initially empty \*/

{ front = 0;

rear = 0;}

else

if(rear == Size-1) /\*rear is at last position of queue \*/

rear = 0;

else

rear = rear+1;

printf("Input the element for adding in queue : ");

scanf("%d", &added\_item);

deque\_arr[rear] = added\_item ;

}

/\*End of insert\_rear\*/

/\*Begin of insert\_front\*/

void insert\_front()

{ int added\_item;

if((front == 0 && rear == Size-1) || (front == rear+1))

{ printf("Queue Overflow \n");

return; }

if (front == -1)/\*If queue is initially empty\*/

{ front = 0;

rear = 0; }

else

if(front== 0)

front=Size-1;

else

front=front-1;

printf("Input the element for adding in queue : ");

scanf("%d", &added\_item);

deque\_arr[front] = added\_item ; }

/\*End of insert\_front\*/

/\*Begin of delete\_front\*/

void delete\_front()

{ if (front == -1)

{ printf("Queue Underflow\n");

return ;

}

printf("Element deleted from queue is : %d\n",deque\_arr[front]);

if(front == rear) /\*Queue has only one element \*/

{ front = -1;

rear=-1;

}

else

if(front == Size-1)

front = 0;

else

front = front+1;

}

/\*End of delete\_front\*/

/\*Begin of delete\_rear\*/

void delete\_rear()

{

if (front == -1)

{

printf("Queue Underflow\n");

return ;

}

printf("Element deleted from queue is : %d\n",deque\_arr[rear]);

if(front == rear) /\*queue has only one element\*/

{

front = -1;

rear=-1;

}

else

if(rear == 0)

rear=Size-1;

else

rear=rear-1; }

/\*End of delete\_rear\*/

/\*Begin of input\_que\*/

void display\_queue()

{

int front\_pos = front,rear\_pos = rear;

if(front == -1)

{ printf("Queue is empty\n");

return;

}

printf("Queue elements :\n");

if( front\_pos <= rear\_pos )

{

while(front\_pos <= rear\_pos)

{

printf("%d ",deque\_arr[front\_pos]);

front\_pos++;

}

}

else

{

while(front\_pos <= Size-1)

{ printf("%d ",deque\_arr[front\_pos]);

front\_pos++;

}

front\_pos = 0;

while(front\_pos <= rear\_pos)

{

printf("%d ",deque\_arr[front\_pos]);

front\_pos++;

}

}

printf("\n");

}

/\*End of display\_queue\*/

/\*Begin of input\_que\*/

void input\_que()

{ int choice;

do

{ printf("1.Insert at rear\n");

printf("2.Delete from front\n");

printf("3.Delete from rear\n");

printf("4.Display\n");

printf("5.Quit\n");

printf("Enter your choice : ");

scanf("%d",&choice);

switch(choice)

{ case 1:

insert\_rear();

break;

case 2:

delete\_front();

break;

case 3:

delete\_rear();

break;

case 4:

display\_queue();

break;

case 5:

break;

default:

printf("Wrong choice\n");

}

}while(choice!=5);

}

/\*End of input\_que\*/

/\*Begin of output\_que\*/

void output\_que()

{ int choice;

do

{ printf("1.Insert at rear\n");

printf("2.Insert at front\n");

printf("3.Delete from front\n");

printf("4.Display\n");

printf("5.Quit\n");

printf("Enter your choice : ");

scanf("%d",&choice);

switch(choice)

{

case 1:

insert\_rear();

break;

case 2:

insert\_front();

break;

case 3:

delete\_front();

break;

case 4:

display\_queue();

break;

case 5:

break;

default:

printf("Wrong choice\n");

}

}while(choice!=5);

}

/\*End of output\_que\*/

/\*Begin of main\*/

main()

{ int choice;

printf("1.Input restricted dequeue\n");

printf("2.Output restricted dequeue\n");

printf("Enter your choice : ");

scanf("%d",&choice);

switch(choice)

{

case 1 :

input\_que();

break;

case 2:

output\_que();

break;

default:

printf("Wrong choice\n");

}

}





