**C programs that implement Queue (its operations) using arrays:**

#include<stdio.h>

#define size 5

int front=-1,rear=-1,q[5];

void enqueue(int);

void dequeue();

void traverse();

void main()

{

int i,item,ch;

while(1)

{

printf("\n Enter your choice 1.Enqueue 2.Dequeue 3.traverse");

scanf("%d",&ch);

switch(ch)

{

case 1:printf("\n Enter the item ");

scanf("%d",&item);

enqueue(item);

break;

case 2:dequeue();

break;

case 3:traverse();

break;

default:exit(0);

}

}

}

void enqueue(int item)

{

if(rear==size-1)

{

printf("\n Queue is full ");

}

else

{

if(front==-1 && rear==-1)

{

front=0;

rear=rear+1;

q[rear]=item;

}

else

{

rear=rear+1;

q[rear]=item;

}

}

}

void dequeue()

{

if(front==-1 && rear==-1)

{

printf("\n Queue is empty");

}

else

{

if(front==rear)

{

printf("Dequeue item is %d",q[front]);

front=-1;

rear=-1;

}

else

{

printf("Dequeue item is %d",q[front]);

front=front+1;

}

}

}

void traverse()

{

int i;

if(front==-1 && rear==-1)

{

printf("\n Queue is empty");

}

else

{

printf("\n Items of Queue are ");

for(i=front;i<=rear;i++)

{

printf("%d\t",q[i]);

}

}

}

**OUTPUT:**

Enter your choice 1.Enqueue 2.Dequeue 3.traverse3

Queue is empty

Enter your choice 1.Enqueue 2.Dequeue 3.traverse1

Enter the item 10

Enter your choice 1.Enqueue 2.Dequeue 3.traverse1

Enter the item 20

Enter your choice 1.Enqueue 2.Dequeue 3.traverse1

Enter the item 30

Enter your choice 1.Enqueue 2.Dequeue 3.traverse1

Enter the item 40

Enter your choice 1.Enqueue 2.Dequeue 3.traverse1

Enter the item 50

Enter your choice 1.Enqueue 2.Dequeue 3.traverse1

Enter the item 60

Queue is full

Enter your choice 1.Enqueue 2.Dequeue 3.traverse3

Items of Queue are 10 20 30 40 50

Enter your choice 1.Enqueue 2.Dequeue 3.traverse2

Dequeue item is 10

Enter your choice 1.Enqueue 2.Dequeue 3.traverse2

Dequeue item is 20

Enter your choice 1.Enqueue 2.Dequeue 3.traverse2

Dequeue item is 30

Enter your choice 1.Enqueue 2.Dequeue 3.traverse2

Dequeue item is 40

Enter your choice 1.Enqueue 2.Dequeue 3.traverse2

Dequeue item is 50

Enter your choice 1.Enqueue 2.Dequeue 3.traverse2

Queue is empty

Enter your choice 1.Enqueue 2.Dequeue 3.traverse3

Queue is empty

Enter your choice 1.Enqueue 2.Dequeue 3.traverse

**C programs that implement Queue (its operations) using linked lists**

#include<stdio.h>

#include<stdlib.h>

void enqueue();

void dequeue();

void traverse();

struct node

{

int data;

struct node \*link;

}\*front,\*rear,\*ptr,\*ptr1,\*header,\*new1;

void main()

{

int i,ch;

header=(struct node\*)malloc(sizeof(struct node));

header->link=NULL;

front=NULL;

rear=NULL;

while(1)

{

printf("\n Enter your choice 1.Enqueue 2.Dequeue 3.traverse");

scanf("%d",&ch);

switch(ch)

{

case 1:enqueue();

break;

case 2:dequeue();

break;

case 3:traverse();

break;

default:exit(0);

}

}

}

void enqueue()

{

int item;

new1=(struct node\*)malloc(sizeof(struct node));

printf("\n enter item to enqueue");

scanf("%d",&item);

if(front==NULL && rear==NULL)

{

header->link=new1;

new1->link=NULL;

front=new1;

rear=new1;

new1->data=item;

}

else

{

rear->link=new1;

new1->data=item;

new1->link=NULL;

rear=rear->link;

}

}

void dequeue()

{

if(front==NULL && rear==NULL)

{

printf("\n Queue is empty");

}

else

{

if(front==rear) /\* Q has only one item \*/

{

header->link=front->link;

printf("Dequeue item is %d",front->data);

front=rear=NULL;

}

else

{

header->link=front->link;

printf("Dequeue item is %d",front->data);

free(front);

front=header->link;

}

}

}

void traverse()

{

ptr=header;

if(front==NULL && rear==NULL)

{

printf("\n Queue is empty");

}

else

{

printf("\n Items of Queue are ");

while(ptr->link!=NULL)

{

ptr=ptr->link;

printf("%d\t",ptr->data);

}

}

}

**OUTPUT:**

Enter your choice 1.Enqueue 2.Dequeue 3.traverse3

Queue is empty

Enter your choice 1.Enqueue 2.Dequeue 3.traverse1

enter item to enqueue10

Enter your choice 1.Enqueue 2.Dequeue 3.traverse1

enter item to enqueue20

Enter your choice 1.Enqueue 2.Dequeue 3.traverse1

enter item to enqueue30

Enter your choice 1.Enqueue 2.Dequeue 3.traverse3

Items of Queue are 10 20 30

Enter your choice 1.Enqueue 2.Dequeue 3.traverse2

Dequeue item is 10

Enter your choice 1.Enqueue 2.Dequeue 3.traverse2

Dequeue item is 20

Enter your choice 1.Enqueue 2.Dequeue 3.traverse3

Items of Queue are 30

Enter your choice 1.Enqueue 2.Dequeue 3.traverse2

Dequeue item is 30

Enter your choice 1.Enqueue 2.Dequeue 3.traverse3

Queue is empty

Enter your choice 1.Enqueue 2.Dequeue 3.traverse2

Queue is empty

Enter your choice 1.Enqueue 2.Dequeue 3.traverse