DATA STRUCTURES
USING "C"

(Queue)

For

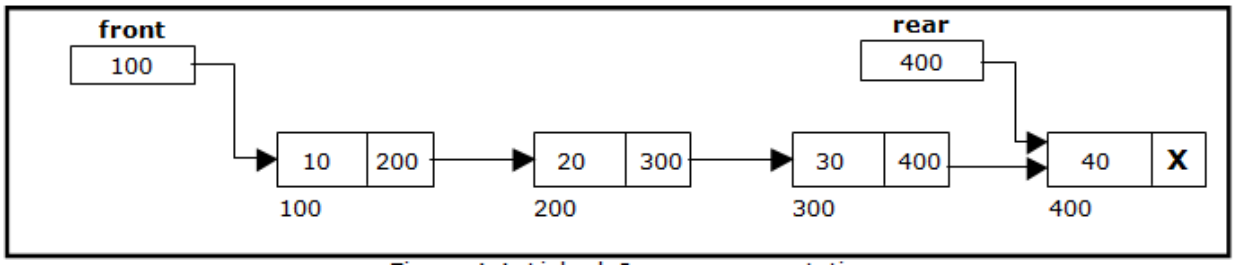
BCA Part-II (Session 2018-21) Students

BY

ANANT KUMAR
MCA, M. Phil, M. Tech.
Faculty Member
Department of Computer Science
J. D. Women's College, Patna

Linked List Implementation of Queue

We can represent a queue as a linked list. In a queue data is deleted from the front end and inserted at the rear end. We can perform similar operations on the two ends of a list. We use two pointers *front* and *rear* for our linked queue implementation.




```
# include <stdlib.h>
# include <conio.h>

typedef struct queue
{
int data;
struct queue *next;
}node;

node *front =NULL;
node *rear =NULL;

node* getnode();
void insertQ();
void deleteQ();
void displayQ();

void main()
{
        char ch;
        while(1)
        {
                clrscr();
                printf("\n \t..Queue operations using pointers.. ");
```

```
		printf("\n\t----------**********		\n");
		printf("\n 1. Insert ");
		printf("\n 2. Delete ");
		printf("\n 3. Display");
		printf("\n 4. Quit ");
		printf("\n Enter your choice: "); ch = getch();
		switch(ch)
		{
			case '1' :
				insertQ();
				break;
			case '2' :
				deleteQ();
				break;
			case '3' :
				displayQ();
				break;
			case '4':
				exit(0);
		}
		getch();
	}
}
node* getnode()
{
	node *temp;
	temp = (node *) malloc(sizeof(node)) ;
	printf("\n Enter data ");
	scanf("%d", &temp -> data);
	temp -> next = NULL;
	return temp;
}


void insertQ()
{
	node *newnode;
	newnode = getnode();
	if(newnode == NULL)
	{
```

```
            printf("\n Queue Full"); return;
        }
        if(front == NULL)
        {
            front = newnode;
             rear =newnode;
        }
        else
        {
            rear -> next = newnode;
            rear =newnode;

        }
}

void deleteQ()
{
        node *temp;
        if(front == NULL)
        {
            printf("\n\n\t Empty Queue..");
        }
        temp = front;
        front = front -> next;
        printf("\n\n\t Deleted element from queue is %d ", temp -> data);
        free(temp);
}

void displayQ()
{
        node *temp; if(front == NULL)
        {
            printf("\n\n\t\t Empty Queue ");
        }
        else
        {
            temp = front;
            printf("\n\n\n\t\t Elements in the Queue are: ");
            while(temp != NULL )
            {
```

```
                printf("%5d ", temp -> data);
                temp = temp -> next;
            }

        }
}
```