**Shri. Vile Parle Kelavani Mandal's**

# Institute of Technology, Dhule

Survey No. 499, Plot No.2, Behind Gurudwara, Mumbai Agra Highway, Dhule (MS)

|  |  |
| --- | --- |
| Name :- Nikita Hire    Roll no :- 10 Date :-\_\_/\_\_/2020    Branch :- Computer    Batch :- s1 | Sign :- |
| Remark :- |



**Write a program to demonstrate Queue Operation**

#include<stdio.h>

int q[5];

int front=-1,rear=-1,i,ch;

void insert(int x)

{

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

{

front++;

rear++;

}

if(rear>5)

{

printf("Queue is Full\n");

}

else

{

q[rear]=x;

rear++;

}

}

void delet()

{

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

{

printf("Queue is Empty\n");

}

else

{

printf("Deleted Element : %d\n ",q[front]);

front++;

}

}

void display()

{

printf("Queue Elements : ");

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

{

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

}

}

int main()

{

int a;

printf("\*\*\*\*\*\*Queue Operation\*\*\*\*\*\*\n ");

printf("\n1.INSERT\n");

printf("2.DELETE\n");

printf("3.DISPLAY\n");

printf("4.EXIT\n");

do

{

printf("\nChoose the operation:(1/2/3/4) : ");

scanf("%d",&ch);

switch(ch)

{

case 1: printf("\nEnter Element : ");

scanf("%d",&a);

insert(a);

break;

case 2:delet();

break;

case 3:display();

break;

case 4:return 0;

}

}while(ch<4);

return 0;

}

**Write a Program to Demonstrate Circular Queue:**

# include<stdio.h>

# define MAX 5

int cqueue\_arr[MAX];

int front = -1;

int rear = -1;

void insert(int item)

{

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

{

printf("Queue Overflow \n");

return;

}

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

{

front = 0;

rear = 0;

}

else

{

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

rear = 0;

else

rear = rear+1;

}

cqueue\_arr[rear] = item ;

}

/\*End of insert\*/

/\*Begin of del\*/

void del()

{

if (front == -1)

{

printf("Queue Underflow\n");

return ;

}

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

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

{

front = -1;

rear=-1;

}

else

{

if(front == MAX-1)

front = 0;

else

front = front+1;

}

}

void display()

{

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 ",cqueue\_arr[front\_pos]);

front\_pos++;

}

else

{

while(front\_pos <= MAX-1)

{

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

front\_pos++;

}

front\_pos = 0;

while(front\_pos <= rear\_pos)

{

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

front\_pos++;

}

}

printf("\n");

}

/\*End of display\*/

/\*Begin of main\*/

int main()

{

int choice,item;

printf("1.Insert\n");

printf("2.Delete\n");

printf("3.Display\n");

printf("4.Quit\n");

do

{

printf("Enter your choice : ");

scanf("%d",&choice);

switch(choice)

{

case 1 :

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

scanf("%d", &item);

insert(item);

break;

case 2 :

del();

break;

case 3:

display();

break;

case 4:

break;

default:

printf("Wrong choice\n");

}

}while(choice!=4);

return 0;

}

**Write a Program to Demonstrate Singly Linked List:**

#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node\* next;

};

int main()

{

struct Node\* head = NULL;

struct Node\* a= NULL;

struct Node\* b = NULL;

struct Node\* last = NULL;

head = (struct Node\*)malloc(sizeof(struct Node));

a= (struct Node\*)malloc(sizeof(struct Node));

b= (struct Node\*)malloc(sizeof(struct Node));

head->data = 10;

head->next = a;

a->data = 20;

a->next = b;

b->data = 30;

b->next = NULL;

last=head;

while(last!=NULL)

{

printf("%d ",last->data);

last=last->next;

}

return 0;

}