PROGRAM 18

Algorithm

INSERT\_REAR (QUEUE, MAX, FRONT, REAR, ITEM)

Step 1:-[Check the overflow condition]

IF REAR= =MAX - 1 then

Print Overflow and end.

Step 2:-[Check 0 element condition]

IF FRONT = =-1 then

Set FRONT = 0 Set REAR = 0

ELSE

REAR = REAR +1

Step 3:-[Element insert at rear position]

Queue [REAR] = ITEM

Step 4:-End

DELETE \_REAR (QUEUE, MAX, FRONT, REAR, ITEM)

Step 1:-IF REAR = = -1 then

Print Underflow and exit

Step 2:-Set ITEM = QUEUE [FRONT]

Step 3:-IF FRONT= =REAR

Then set FRONT = = -1

Then set REAR = = -1

ELSE

REAR = REAR-1

Step 4:-End

INSERT \_FRONT(QUEUE, MAX, FRONT, REAR, ITEM)

Step 1:-[Check the overflow condition]

IF FRONT= =0 then

Print Overflow and end.

Step 2:-[Check 0 element condition]

IF FRONT = =-1 then

Set FRONT = 0 Set REAR = 0

ELSE

FRONT= FRONT-1

Step 3:-[Element insert at rear position]

Queue [FRONT] = ITEM

Step 4:-End

DELETE \_FRONT (QUEUE, MAX, FRONT, REAR, ITEM)

Step 1:-IF FRONT = = -1 then

Print Underflow and exit

Step 2:-Set ITEM = QUEUE [FRONT]

Step 3:-IF FRONT= =REAR

Then set FRONT = = -1

Then set REAR = = -1

ELSE

FRONT = FRONT -1

Step 4:-End

Source code

#include<stdio.h>

#include<stdlib.h>

#define MAX 3

int FRONT =-1;

int REAR =-1;

int QUEUE[MAX];

int inback(int []);

int infront(int []);

int delfront(int []);

int delback(int []);

int display(int []);

void main()

{

int a;

system("COLOR F0");

printf("\n1.Insert at Rear");

printf("\n2.insert at Front");

printf("\n3.Delete at Front");

printf("\n4.Delete at Rear");

printf("\n5.Display");

printf("\n6.End");

do

{

printf("\nEnter the Choice");

scanf("%d",&a);

switch(a)

{

case 1:inback(QUEUE);

break;

case 2:infront(QUEUE);

break;

case 3:delfront(QUEUE);

break;

case 4:delback(QUEUE);

break;

case 5:display(QUEUE);

break;

case 6:exit(0);

break;

default:printf("\nWrong Choice");

}

}while(1);

}

int inback(int QUEUE[MAX])

{

int item;

if(REAR==MAX-1)

{

printf("\nOverflow");

return;

}

if(FRONT==-1)

{

FRONT=0;

REAR=0;

}

else

{

REAR=REAR+1;

}

printf("\nEnter the item");

scanf("%d",&item);

QUEUE[REAR]=item;

return 0;

}

int infront(int QUEUE[MAX])

{

int item;

if(FRONT==0)

{

printf("\nOverflow");

return;

}

if(FRONT==-1)

{

FRONT=0;

REAR=0;

}

else

{

FRONT=FRONT-1;

}

printf("\nEnter the item");

scanf("%d",&item);

QUEUE[FRONT]=item;

return 0;

}

int delfront(int QUEUE[MAX])

{

int item;

if(FRONT==-1)

{

printf("\nunderflow");

}

item=QUEUE[FRONT];

if(FRONT==REAR)

{

FRONT=-1;

REAR=-1;

}

else

{

FRONT=FRONT-1;

}

return;

}

int delback(int QUEUE[MAX])

{

int item;

if(REAR==-1)

{

printf("\nunderflow");

}

item=QUEUE[REAR];

if(FRONT==REAR)

{

FRONT=-1;

REAR=-1;

} else

{ REAR=REAR-1; } }

int display(int QUEUE[MAX]) { int i;

for(i=FRONT;i<=REAR;i++)

{ printf("%d\n",QUEUE[i];

}

