EXTRA PROGRAMS:

```
DEQUEUE:
#include<stdio.h>
#include<stdlib.h>
#define qsize 3
int f=0,r=-1,ch;
int item,q[10];
int isfull()
  {
   return(r==qsize-1)?1:0;
  }
int isempty()
   return(f>r)?1:0;
  }
void insert_rear()
  {
   if(isfull())
      {
       printf("queue overflow\n");
        return;
      }
   r=r+1;
   q[r]=item;
  }
```

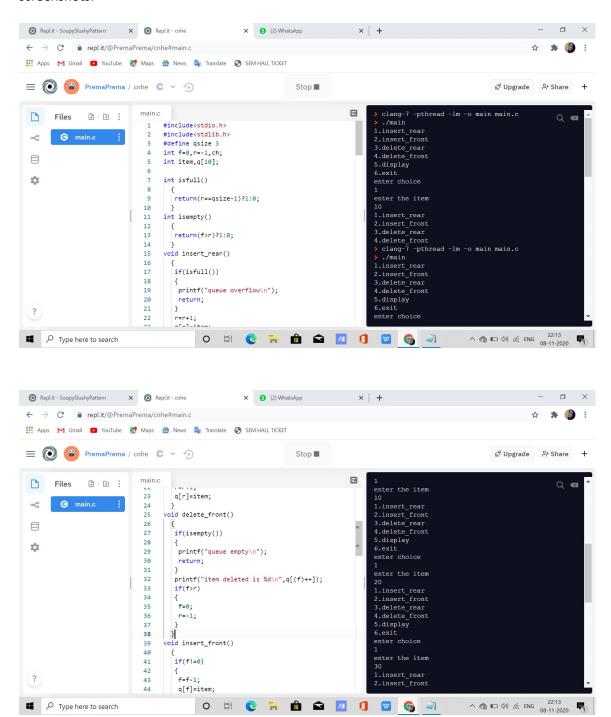
```
void delete_front()
  {
   if(isempty())
       printf("queue empty\n");
       return;
      }
   printf("item deleted is %d\n",q[(f)++]);\\
   if(f>r)
      {
       f=0;
       r=-1;
      }
  }
void insert_front()
  {
   if(f!=0)
      {
       f=f-1;
       q[f]=item;
       return;
     else if((f==0)&&(r==-1))
      {
       q[++(r)]=item;
       return;
      }
```

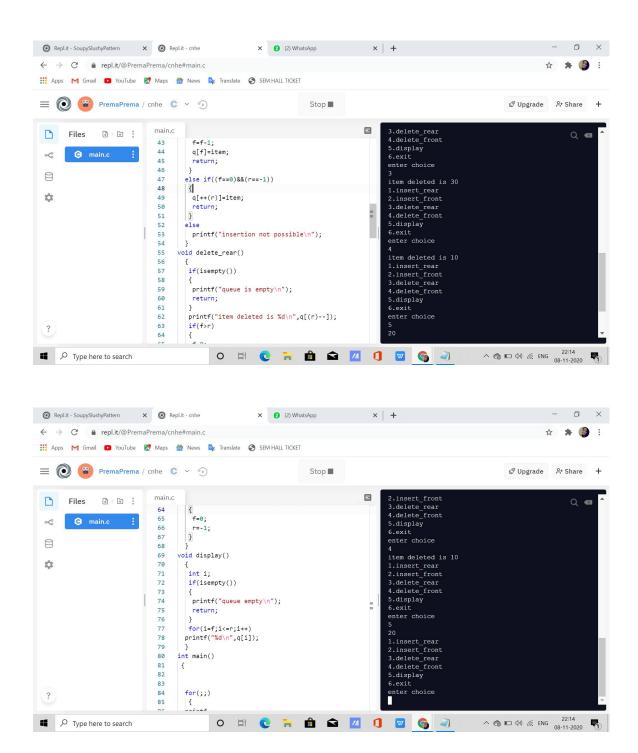
```
else
        printf("insertion not possible\n");
  }
void delete_rear()
  {
   if(isempty())
      {
        printf("queue is empty\n");
        return;
      }
   printf("item deleted is %d\n",q[(r)--]);
   if(f>r)
      {
        f=0;
        r=-1;
      }
  }
void display()
  {
   int i;
   if(isempty())
      {
        printf("queue empty\n");
        return;
      }
   for(i=f;i<=r;i++)
     printf("%d\n",q[i]);
```

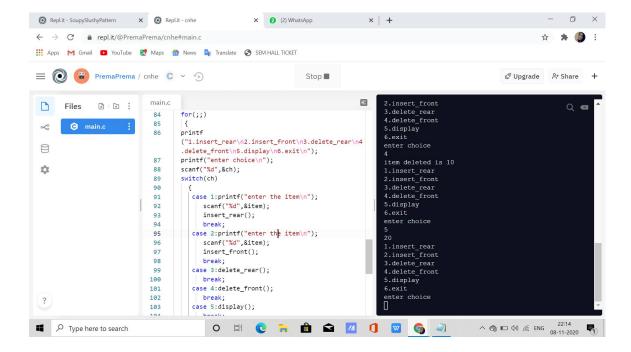
```
}
int main()
 {
  for(;;)
   {
     printf("1.insert\_rear\n2.insert\_front\n3.delete\_rear\n4.delete\_front\n5.display\n6.exi
t\n");
     printf("enter choice\n");
     scanf("%d",&ch);
     switch(ch)
        {
         case 1:printf("enter the item\n");
                   scanf("%d",&item);
                   insert_rear();
                   break;
         case 2:printf("enter the item\n");
                   scanf("%d",&item);
                   insert_front();
                   break;
         case 3:delete_rear();
                   break;
         case 4:delete_front();
                   break;
         case 5:display();
                   break;
         default:exit(0);
```

```
}
}
```

screenshots:







written:

```
PAGE NO.

DATE: / /
  Queus.
-> Dequeue. ("input rustricted and out rustricted)
  #include < stdio.h>
  # define quize 3
  int f=0, r=-1, ch;
  int item, g[10];
  int isfull)
  return ( == qsize-1) ?1:0;
  int is empty ()
  return (+ >>) ? 1:0;
  void invent_rear()
   i+ ((stul())
   $ pt (" guen overflow \n");
   return:
   2[x++]=item:
```

```
PAGE NO.
DATE: / /
   void dulte-front()
   if (isempty())
   f pt ("underflow")
   ) else
   $
    , isterms of ("item duted is v.d", g[++]);
    if (+>7)
    t t = 0;
    7=-1;
/* void insurt_front()
    9 94 (11=0)
    9+=4-1;
    2[+] = item;
    return,
    olse if ( (+==0) 88 (7==-1))
    q[++ ]= item;
    return;
    pt ("involtion not possible in");
    Il commend insert front to input sustrication
```

```
PAGE NO.
/* void delute-rear()
fit (Psempty ())
   1 pt (" queu to empty m");
   return, Cur winds water That
   P+ ("item duted is rd'in", q[r-]);
   1+ (+>7) Wage course with material 1 341 1 2 2000
   f=0;
   ~=-1;
   */ "womment delite man to sustrict output
   void display() where it with a should by a second
   int i;
   it (icempty())
   prints ( queu empty in");
   return;
   ter (i=+; i = 7; i++)
  p+ (" xdm", g[i7);
   void main!)
   ton(;;)
```

```
PAGE NO.
DATE: / /
pf ("1. insurt rear 12. insurt frond 173. delite may
m4 delete front ms display mc exit m");
pt 1" enter choice m");
st (" y.d", sun);
switch(ch) case 1: pf ("enter the item in");
st (" 7. d", sitem);
insuit rearly;
break; was ablue was in
case 2: pt ("enter the itemin");
s+ ("y.d", gitem);
insurt-front ();
break;
case 3: delete front rear ();
break; When y have a straight at the
case 4: delete-pront();
 break;
case 5: display():
break;
default: exit(0);
```