Linear Queue.

```
# Include <stdio.h>.
# Include <Stdlib.h>.
# define QNE_SIZE 5
 int item, front=0, rear=-1, q[10];

 void insert_rear()
{
   if (Rear == QNE SIZE -1){.
      Printf(" Queue overflow");
      Return;
}
   Rear = rear +1;
   q[Rear]= item;
}
 int delete front (){
 if ( front > rear ) {.
    front =0;
    Rear =-1;
   Return -1;
}
  return q[front++];
}
  void display Q() {.
     int i;
 if (front >rear) {.
    Print ("Queue is Empty\n");
      return;
```

```
}.
Printf ("contents of queue \n");
for(i = front; i<= rear; i++) {.
    Printf ("%d \n", q[i]);

}
}.
int main() {.
  int choice;
  for(;;) {.
 Printf ("1: insert rear  2. delete front  3. display
 4: exit \n");
Printf (" Enter the choice \n");
scanf ("%d", &choice);
 switch (choice) {.
  case 1: Printf ("enter the item to be inserted");
  scanf ("%d", &item);
  insert rear ();
  break;
  case 2: item = delete front ();
  if (item == -1)
   Printf (" Queue is empty\n");
  else
  Printf ("item deleted = %d \n", item);
   break;
case 3 : displayQ();
 break;
default : exit(0);
}
}
}.
```

output :

```
1. Insert rear   2: delete front   3: display   4: exit.

 Enter the choice   1

 Enter the item to be inserted   10.

1. Insertrear   2. deletefront   3. display   4. exit.

  Enter the choice  1

  Enter the item to be inserted  20.

1. Insertrear   2. deletefront   3. display   4. exit.

  Enter the choice  1.

   Enter the item to be inserted  30.

1. Insertrear  2. deletefront   3. display   4. exit
   Enter the choice  1.

    Enter the item to be inserted  40.

  queue overflow.

 1. Insertrear   2. delete front   3. display   4. exit.

  Enter the choice. 3
```

```
contents of queue

  10
  20
  30.

1. Insert Rear 2. delete front 3. display 4. exit.

Enter the choice ~~3~~ 2.

Item deleted = 10.

1. Insert Rear 2. delete front 3. display 4. exit.

Enter the choice 2.

Item deleted = 20.

1. Insert Rear 2. delete front 3. display 4. exit

Enter the choice 2.

item deleted = 30.

1. Insert Rear 2. delete front 3. display 4. exit

Enter the choice 2.

queue underflow.
```