```
Circular Queul
# welude Latdio. h>
                                              ( ) many book
# include < process h >
int item, front = 0, real = -1, q, [QUE_SIZE], wunt = 0;
# define QUE-SIZE 3
roid insertrear ()
    if (wunt == QUE_SIZE)
     prints ("Queue Overflow m");
     return;
 rear = (rear +1) / BUE_SIZE;
 g[rear] = item;
  count++;
 int delete front ()
   4 (court == 0) return -1;
    item = g[front];
    front = (front+1)/. QUE_SIZE;
    return item;
  roid display (1)
     int i,f;
     if ( rount = = 0)
    2 print (" Queue is empty \n");
      return;
   f=front;
  printy (" Contents of queue In");
   for (out i=1; i <= count; i++)
      printf (" 1.d \n', 9[5]);
      f = (+1) /. QUE-SIZE;
```

of while & ethis he roid main () of include express his int choice; # define que size 3 run (Fazis 300) printf ("In 1: insertreas In2: deletefront In 3: display In 4: exit (n"); print ("Enter the choice (n"); or privily Columns Ever scarf (" 1/2 d", & choice); switch (choice) case 1: printf ("Enter the item to be inserted In"); scanf ("/d", & item); met trusty courters insertreas (); break; ist delete front () case 2° item = delete front (); if (court == 0) return if (iten == -1) printf () Juene is empty (n'); print ("Item deleted = %d \n", item); break ; case 3: display(); wid displayar break; it is ti case 4: exit (0); if (court = = 0) break; default: prints ("Invalid choice \n"); virty ("Contents of queue In"); for (=1) ic= count; i++) print ("1.d. (a. 9. [5]); f = (1+1) t. QUE-SIZE;