22/10/24

WEEK → 04

Circular Queue

```
#include <stdio.h>
#define SIZE 5
int f = -1, rear = -1;
int q[SIZE];

enqueue (int item)
if (f == (r+1) % SIZE) queue is full
else r = (r+1) % size, and in r's place put the value
and if f == -1 increment f: f+1;

dequeue ()
if f == -1 print queue is empty;
else element deleted is %d, d[f];
if [f == r] f = -1 and r = -1
 else f = (f+1) % SIZE

display ()
if (f == -1) queue is empty.
else
print ("content of queue")
   for (i=f; i!=r; i=(i+1)% size);
      printf ("%d", q[i]);
      printf ("%d", q[r]);
```

Code:

```
#include <stdio.h>
#include <stdlib.h>
#define size 5

int f = -1;
int r = -1;
int q[size];

void enqueue(int item)
{
  if (f == (r+1) % size)
  printf("Queue is full");
  else {
     r = (r+1) % size;
     q[r] = item;
     if (f == -1)
        f = f+1;
  }
}

void dequeue()
{
  if (f == -1)
  printf("Queue is empty");
  else {
    printf("\n the deleted element is %d", q[f]);
```

```
if (f == r)
{
    f = -1;
    r = -1;
}
else {
    f = (f+1) % size;
}
}
}
void display() {
int i;
if (f == -1);
printf ("Queue is empty");
else {
    printf ("Content of queue");
    for( i = f ; i != r , i = (i+1) % size)
    printf ("%d", q[i]);
    printf ("%d", q[r]);
}
}
int main()
{
int ch, item;
for(;;)
{
```

```
printf("1. Insert");
printf("2. Delete");
printf("3. Display");
printf("4. Exit");
printf("Read Choice: ");
scanf("%d", &ch);

switch (ch)
{
  case 1: printf("Element to be inserted: ");
          scanf("%d", &item);
          enqueue(item);
          break;
  case 2: dequeue();
          break;
  case 3: display()
          break;
  default: exit(0);
  }
}
return 0;
}
```

Output

1. Insert
2. Delete
3. Display
4 exit

Read choice: 1
Element inserted : 10

read choice : 1
element inserted : 20

read choice : 1
element inserted : 30

read choice 1
element inserted : 40

read choice : 1
element inserted : 50

read choice : 1
element inserted : 60
Queue is full.

read choice : 2
element deleted is : 10

read choice : 3
elements present : 20 30 40 50

read choice : 1
Element inserted : 60

read choice : 3
element present : 20 30 40 50 60

22/10/24

Leet Code 2073


```
class solution {
public:
    int timeRequiredToBuy(vector<int>& ticket, int k){
        int c=0;
        while(true){
            if(tickets[k]==0)
                break;
            for(int i=0; i<tickets.size(); i++){
                if(tickets[k]==0)
                    break;
                if(tickets[i]>0){
                    tickets[i]--;
                }
            }
        }
        return c;
    }
};
```

Output:

Case 1: [2,3,2]

k = 2

Output = 6

Expected = 6

Case 2: [5,1,1,1]

k = 0

Output = 8

Expected = 8

22/10/24