Write files: queue.c queue.h and testqueue.c

Submit a single file: MISID.tar.gz

Implement a queue of structures. The queue stores the following structure:

```
typedef struct data {
   char name[16];
   unsigned int age;
}data;
```

Write the following queue functions: qinit, enq, deq, qfull, qempty;

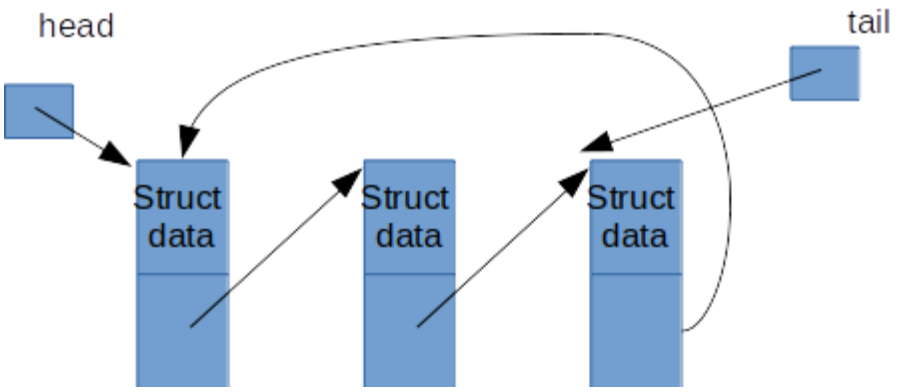
Define the proper prototypes in queue.h and write the code in queue.c

The queue implementation should be done using singly linked circularly connected structures (nodes), where each node will store one instance of 'data'.

You can use head and tail - both pointers

In qfull, you may simply assume that the queue is never full and write code accordingly. The real challenging part (for no extra marks) is to detect full if malloc fails, but still make sure that qfull() can detect it and enq() always works if qfull() said not-full.

Here is a diagram of how your queue will store the data: Note that *tail is optional* and you may or may not use it.



Do a proper typedef of the queue type in queue.h

Use this main() function in a file called testqueue.c to test your queue. Make sure that your code compiles with this code and you can test it.

```
#include <stdio.h>

#include "queue.h"
int main() {
   queue q;
   data d;

   qinit(&q);
   while(scanf("%s%u", d.name, &(d.age)) != -1)
      if(!qfull(&q))
         enq(&q, d);
   while(!qempty(&q)) {
      d = deq(&q);
      printf("%s %u\n", d.name, d.age);
   }
   return 0;
}
```