Pair Programming 4 Turn In

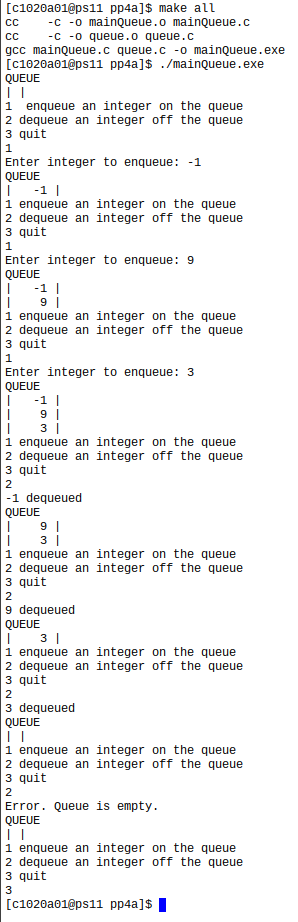
Name: \_\_\_\_\_Braden Bell\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Username: \_\_\_\_c1020a01\_\_\_\_\_\_

Partner name: \_\_\_\_\_\_Thomas Snyder\_\_\_\_\_\_\_\_\_\_\_\_ Partner username: \_\_\_\_c1020a10\_\_\_\_\_\_\_\_

\_X\_I certify that my partner worked with me on this assignment.

SCORE: \_\_\_\_\_\_\_\_\_\_\_\_ (to be filled in by instructor)

4a (5 points)



**[PASTE THE SCREEN SHOT OF THE TESTS WITH A WHITE BACKGROUND FOR PAIR PROGRAMMIG 4A HERE]**

/\* File: queue.c

\* Author: Braden Bell

\* Description: contains functions for queue manipulations

\*/

#include <stdio.h>

#include "queue.h"

#include "queueItem.h"

/\* initQueue: initializes the queue

\* Parameters:

\* q - pointer to the queue

\* Returns: nothing

\*/

void initQueue(Queue\* q) {

q -> count = 0;

q -> front = 0;

q -> rear = QUEUE\_SIZE - 1;

}

/\* enqueue: enqueues an item in the queue if the queue is not full

\* Parameters:

\* q - pointer to the stack

\* item - the item to push

\* Returns true if enqueue successful, false otherwise

\*/

int enqueue(Queue\* q, QueueItem item) {

if (q -> count < QUEUE\_SIZE) {

q -> count++;

//printf("Count: %d, Rear: %d\n", q -> count, q -> rear);

q -> rear = (q -> rear + 1) % QUEUE\_SIZE;

//printf("Rear: %d\n", q -> rear);

q -> array[ q -> rear] = item;

return 1;

}

return 0;

}

/\* dequeue: dequeues an item in the queue if the queue is not empty

\* Parameters:

\* q - pointer to the queue

\* item - the item to be returned, a pointer

\* Returns: true if dequeue was successful, false otherwise

\*/

int dequeue(Queue\* q, QueueItem\* item) {

if (q -> count > 0) {

q -> count--;

\*item = q -> array[q -> front];

q -> front = (q -> front + 1) % QUEUE\_SIZE;

return 1;

}

else

return 0;

}

/\* isEmpty: Returns true if queue is empty

\* Parameters:

\* q - the stack

\* Returns: true if stack is empty, false otherwise

\*/

int isEmpty(Queue q) {

if (q.count == 0)

return 1;

else

return 0;

}

/\* printStack: Prints all content in the queue

\* Parameter q, the stack

\* Returns nothing

\*/

void printQueue(Queue q) {

int i;

printf("QUEUE\n");

//printf("Count: %d, Rear: %d, Front: %d\n", q.count, q.rear, q.front);

if (q.count > 0) {

for (i = q.front; i != q.rear; i = (i + 1) % QUEUE\_SIZE) {

printf("| %4d |\n", q.array[i]);

}

}

if (q.count > 0) {

printf("| %4d |\n", q.array[q.rear]);

}

else

printf("| |\n");

}

/\* File: queue.h

\* Author: Braden Bell

\* Description: header file for a circular queue

\*/

#ifndef QUEUE\_H

#define QUEUE\_H

#include "queueItem.h"

#define QUEUE\_SIZE 10

typedef struct {

QueueItem array[QUEUE\_SIZE];

int count;

int front;

int rear;

} Queue;

void initQueue(Queue\*);

int enqueue(Queue\*, QueueItem);

int dequeue(Queue\*, QueueItem\*);

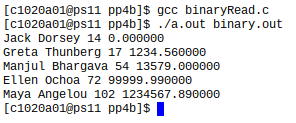
int isEmpty(Queue q);

void printQueue(Queue);

#endif

**[PASTE CODE FOR PAIR PROGRAMMING 4A HERE – NOT A SCREEN SHOT. DOWNLOAD queue.h, queue.c AND Makefile. OPEN THEM WITH A TEXT EDITOR LIKE NOTEPAD++ OR WORDPAD, COPY THE CODE, PASTE IT HERE. MAKE SURE IT IS SINGLE SPACED AND USE A COURIER NEW FONT]**

4b (5 points)



**[PASTE THE SCREEN SHOT OF THE TESTS WITH A WHITE BACKGROUND FOR PAIR PROGRAMMIG 4B HERE]**

/\* File: binaryRead.c

\* Author: Cindy

\* Description: example of reading binary data in a structure

\*/

#include <stdio.h>

#define MAX\_NAME\_LENGTH 30

typedef struct employee {

char firstName[MAX\_NAME\_LENGTH];

char lastName[MAX\_NAME\_LENGTH];

int age;

double salary;

} Employee;

int main(int argc, char \*argv[]) {

Employee e[5];

// ADD CODE HERE TO declare file pointer

FILE\* inFileStream;

int i = 0;

// ADD CODE HERE TO

if(argc == 2) {// if there is a filename argument

inFileStream = fopen(argv[1], "rb");

if (inFileStream != NULL) {//if file open is successful, open binary file whose name retreived from command line for reading

//ADD CODE HERE TO read 5 employees from the binary file

fread( &e[i], sizeof(Employee), 1, inFileStream);

while(!feof(inFileStream)) {

i++;

fread(&e[i], sizeof(Employee), 1, inFileStream);

}

}

//ADD CODE HERE TO close the input file

fclose(inFileStream);

// print employees read

for( i= 0; i < 5; i++ ) {

printf("%s %s %d %lf\n", e[i].firstName, e[i].lastName, e[i].age, e[i].salary );

}

}

return 0;

}

[PASTE **CODE FOR PAIR PROGRAMMING 4B HERE – NOT A SCREEN SHOT. DOWNLOAD binaryRead.c. OPEN IT WITH A TEXT EDITOR LIKE NOTEPAD++ OR WORDPAD, COPY THE CODE, PASTE IT HERE. MAKE SURE IT IS SINGLE SPACED AND USE A COURIER NEW FONT]**