Document version: 1.1 (2015-11-15)

Curtin University – Department of Computing

Assignment Cover Sheet / Declaration of Originality

Complete this form if/as directed by your unit coordinator, lecturer or the assignment specification.

Last name:	Marcelo	Student ID:	163785
Other name(s):	Terence		
Unit name:	Operating Systems	Unit ID:	COMP2006
Lecturer / unit coordinator:	Sie Teng Soh	Tutor:	
Date of submission:	18 MAY 2020	Which assignment?	(Leave blank if the unit has only one assignment.)

I declare that:

- The above information is complete and accurate.
- The work I am submitting is *entirely my own*, except where clearly indicated otherwise and correctly referenced.
- I have taken (and will continue to take) all reasonable steps to ensure my work is *not accessible* to any other students who may gain unfair advantage from it.
- I have *not previously submitted* this work for any other unit, whether at Curtin University or elsewhere, or for prior attempts at this unit, except where clearly indicated otherwise.

I understand that:

- Plagiarism and collusion are dishonest, and unfair to all other students.
- Detection of plagiarism and collusion may be done manually or by using tools (such as Turnitin).
- If I plagiarise or collude, I risk failing the unit with a grade of ANN ("Result Annulled due to Academic Misconduct"), which will remain permanently on my academic record. I also risk termination from my course and other penalties.
- Even with correct referencing, my submission will only be marked according to what I have done myself, specifically for this assessment. I cannot re-use the work of others, or my own previously submitted work, in order to fulfil the assessment requirements.
- It is my responsibility to ensure that my submission is complete, correct and not corrupted.

Signature: Terence

Date of signature: 18 MAY 2020

(By submitting this form, you indicate that you agree with all the above text.)

Part A

The program has four threads, one thread for 'LIFT-R' (the producer) and three lift threads (consumers). I used a buffer shared between the four threads implemented using a first-in-first-out queue. A mutex lock was used once any of the four threads would write to the 'sim_out' file. A thread would acquire the mutex lock once it needs to write the to the 'sim-out' file and release it once it reaches the end of that iteration of the while loop. Therefore, only one thread can write to sim_out at a time.

For signal handling, whenever an item would be added to buffer by 'LIFT-R', a signal would be sent to any of the lift threads signalling that the buffer is not empty. When any of the lifts would remove an item from the buffer, they would signal to 'LIFT-R' that the buffer is not completely full.

Inside the function called by the 'LIFT-R' thread is a while loop that runs until the entire file has been read and stored to the buffer. Inside the function for the 3 lift threads is another while loop the runs while the buffer isn't empty or that the file hasn't been read to make sure that the thread doesn't end just because there's nothing in the buffer yet.

Lift_sim_A.c

```
#include <stdio.h>
#include <strings.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <time.h>
#include "lift_sim_A.h"
#include "Queue.h"

pthread_mutex_t bufferMutex = PTHREAD_MUTEX_INITIALIZER;
pthread_cond_t notEmpty = PTHREAD_COND_INITIALIZER;
pthread_cond_t notFull = PTHREAD_COND_INITIALIZER;

int main( int argsNo, char** args )
{
   if( argsNo < 3 )
   {
      printf( "\nPlease specify buffer size and time required.");
   }
   else
   {
      FILE* sim_input = fopen( "sim_input", "r" );
      FILE* sim_output = fopen( "sim_output", "a" );
      int totalSoFar = 0;
      /* totalSoFar: Incremented everytime something added to buffer */
      Queue* buffer = newQueue( atoi( args[1] ) );
      struct filesAndBuffer threadArgs;

      pthread_t Lift_R;
      pthread_t Lift_1;
      pthread_t Lift_2;
      pthread_t Lift_3;


      if( sim_input == NULL || sim_output == NULL )
      {
         printf( "\nsim_input or sim_output file is missing.\n");
         printf( "Please create those files first.\n\n\n");
      }
      else
      {
         /* Arguments to pass to each thread:   */
         /* - Pointers to both files, buffer, and total items to buffer so far */
         /* - Integer of how long a lift should take */
         /* - Integer for wether a line in the file is wrong */
         threadArgs.sim_input = sim_input;
         threadArgs.sim_output = sim_output;
         threadArgs.buffer = buffer;
         threadArgs.totalSoFar = &totalSoFar;
         threadArgs.liftTime = atoi( args[2] );
         threadArgs.fileErrors = 0;

         printf( "\n\nNow processing. Please Wait.\n\n" );

         pthread_create( &Lift_R, NULL, request, (void*)&threadArgs );
         pthread_create( &Lift_1, NULL, lift, (void*)&threadArgs );
         pthread_create( &Lift_2, NULL, lift, (void*)&threadArgs );
         pthread_create( &Lift_3, NULL, lift, (void*)&threadArgs );
         pthread_join( Lift_R, NULL );
         pthread_join( Lift_1, NULL );
         pthread_join( Lift_2, NULL );
         pthread_join( Lift_3, NULL );

      }
      free( buffer );
      fclose( sim_input );
      fclose( sim_output );
   }

   printf( "\n\n" );
   return 0;
}

void* request( void* inThreadArgs )
{
   int nRead, source, destination;
   struct filesAndBuffer* threadArgs = ( struct filesAndBuffer* )inThreadArgs;
   FILE* sim_input = threadArgs->sim_input;
   FILE* sim_output = threadArgs->sim_output;
   Queue* buffer = threadArgs->buffer;


   while( !feof( sim_input ) )
   {
      pthread_mutex_lock( &bufferMutex );
      while( buffer->total == buffer->sizeOfBuffer )
      {
         /* Wait first if buffer is still full */
         pthread_cond_wait( &notFull, &bufferMutex );
      }

      nRead = fscanf( sim_input, "%d %d\n", &source, &destination );
      if( nRead == 2 )
      {
         /* Adds to buffer, prints to file, increments total so far, and */
         /* signals that the buffer isn't empty.   */
         insert( buffer, source, destination );
         ( *threadArgs->totalSoFar )++;
         fprintf( sim_output, "\n\n_____________________________________" );
         fprintf( sim_output, "\nNew Lift Request from %d to %d",
                                                  source, destination);
         fprintf( sim_output, "\nRequest No:%d", *threadArgs->totalSoFar );
         fprintf( sim_output, "\n_____________________________________\n\n" );
         pthread_cond_signal( &notEmpty );
      }
      else if( nRead != 2 )
      {
         threadArgs->fileErrors = 1;
         /*Program doesn't continue if there's an invalid line in the file. */
         printf( "\n\nInvalid line detected." );
         printf( "\nCheck 'sim_input' and try again." );
      }
      pthread_mutex_unlock( &bufferMutex );
   }


   return NULL;
}

void* lift( void* inThreadArgs )
{
   QueueNode* nextToDo;
   struct filesAndBuffer* threadArgs = ( struct filesAndBuffer* )inThreadArgs;
   FILE* sim_input = threadArgs->sim_input;
   FILE* sim_output = threadArgs->sim_output;
   Queue* buffer = threadArgs->buffer;
   int currentPosition = 1, requestsReceived = 0;
   int totalMovement = 0, currentMovement = 0;
   /* Integers above are just for the numbers that need to be written to file */
   /* Elevators start at 1 so currentPosition is initialised to 1   */
   while( ( buffer->total != 0 || !feof( sim_input ) )
                                            && threadArgs->fileErrors != 1 )
   {
      /* When there are items in the buffer or there will be in the */
      /* future and that there hasn't been any bad lines in the file.*/
      pthread_mutex_lock( &bufferMutex );
      while( buffer->total == 0 && !feof( sim_input ) )
      {
         pthread_cond_wait( &notEmpty, &bufferMutex );
         /* Wait when there aren't any items in the buffer but */
         /* there will be in the future.  */
      }

      if( buffer->total != 0 )
      {
         nextToDo = deQueue( buffer );
         requestsReceived++;
         currentMovement = ( abs( currentPosition - nextToDo->source ) ) +
                        abs( ( nextToDo->destination - nextToDo->source ) );
         totalMovement = totalMovement + currentMovement;
         fprintf( sim_output, "\n\nLift Operation" );
         fprintf( sim_output, "\nPrevious Position:%d",currentPosition );
         fprintf( sim_output, "\nRequest:Floor %d to Floor %d",
                                    nextToDo->source, nextToDo->destination );
         fprintf( sim_output, "\nDetails:" );
         fprintf( sim_output, "\n   Go from floor %d to floor %d",
                                          currentPosition, nextToDo->source );
         fprintf( sim_output, "\n   Go from floor %d to floor %d",
                                    nextToDo->source, nextToDo->destination );
         fprintf( sim_output, "\n   #Movement for this request:%d",
                                                            currentMovement );
         fprintf( sim_output, "\n   #Request:%d", requestsReceived );
         fprintf( sim_output, "\n   Total #Movement:%d", totalMovement );
         currentPosition = nextToDo->destination;
         fprintf( sim_output, "\nCurrent Position:%d\n\n", currentPosition );

         free( nextToDo );
         sleep( threadArgs->liftTime );
         pthread_cond_signal( &notFull );
         /* Send signal that there's now one less item in buffer. */
      }
      pthread_mutex_unlock( &bufferMutex );
   }

   return NULL;
}
```

Lift_sim_A.h

```
#include "Queue.h"
struct filesAndBuffer
{
   FILE* sim_input;
   FILE* sim_output;
   Queue* buffer;
   int* totalSoFar;
   int liftTime;
   int fileErrors;
};
void* request( void* inThreadArgs );
void* lift( void* inThreadArgs );
```

Part A Sample Input

Sample Input located in the 'sim_input' file is shown in the right with 50 elevator requests.

When the program is ran with a buffer size of 6 and elevator duration (sleep function) of 4, the result of 'sim-output' is shown below.
e.g. ./lift_sim_A 6 4

```
19 18
13 3
19 7
17 16
1 4
6 20
11 12
11 18
13 8
10 18
10 14
20 18
11 16
14 2
5 17
3 20
15 5
3 4
15 17
1 5
4 11
15 8
7 11
16 11
17 11
5 6
5 14
7 15
20 4
2 13
10 2
5 10
17 18
5 4
12 13
4 9
10 6
19 7
2 12
17 14
15 9
5 12
14 19
20 14
17 20
3 12
5 12
3 4
17 12
20 3
```

```
_____________________________________
New Lift Request from 19 to 18
Request No:1
_____________________________________



_____________________________________
New Lift Request from 13 to 3
Request No:2
_____________________________________



_____________________________________
New Lift Request from 19 to 7
Request No:3
_____________________________________



_____________________________________
New Lift Request from 17 to 16
Request No:4
_____________________________________



_____________________________________
New Lift Request from 1 to 4
Request No:5
_____________________________________



_____________________________________
New Lift Request from 6 to 20
Request No:6
_____________________________________


Lift Operation
Previous Position:1
Request:Floor 19 to Floor 18
Details:
   Go from floor 1 to floor 19
   Go from floor 19 to floor 18
   #Movement for this request:19
   #Request:1
   Total #Movement:19
Current Position:18


Lift Operation
Previous Position:18
Request:Floor 13 to Floor 3
Details:
   Go from floor 18 to floor 13
   Go from floor 13 to floor 3
   #Movement for this request:15
   #Request:2
   Total #Movement:34
Current Position:3


Lift Operation
Previous Position:3
Request:Floor 19 to Floor 7
Details:
   Go from floor 3 to floor 19
   Go from floor 19 to floor 7
   #Movement for this request:28
   #Request:3
   Total #Movement:62
Current Position:7
```

```
Lift Operation
Previous Position:7
Request:Floor 17 to Floor 16
Details:
   Go from floor 7 to floor 17
   Go from floor 17 to floor 16
   #Movement for this request:11
   #Request:4
   Total #Movement:73
Current Position:16


Lift Operation
Previous Position:16
Request:Floor 1 to Floor 4
Details:
   Go from floor 16 to floor 1
   Go from floor 1 to floor 4
   #Movement for this request:18
   #Request:5
   Total #Movement:91
Current Position:4


Lift Operation
Previous Position:4
Request:Floor 6 to Floor 20
Details:
   Go from floor 4 to floor 6
   Go from floor 6 to floor 20
   #Movement for this request:16
   #Request:6
   Total #Movement:107
Current Position:20



_____________________________________
New Lift Request from 11 to 12
Request No:7
_____________________________________



_____________________________________
New Lift Request from 11 to 18
Request No:8
_____________________________________



_____________________________________
New Lift Request from 13 to 8
Request No:9
_____________________________________



_____________________________________
New Lift Request from 10 to 18
Request No:10
_____________________________________



_____________________________________
New Lift Request from 10 to 14
Request No:11
_____________________________________



_____________________________________
New Lift Request from 20 to 18
Request No:12
_____________________________________
```

```
Lift Operation
Previous Position:1
Request:Floor 11 to Floor 12
Details:
   Go from floor 1 to floor 11
   Go from floor 11 to floor 12
   #Movement for this request:11
   #Request:1
   Total #Movement:11
Current Position:12



Lift Operation
Previous Position:12
Request:Floor 11 to Floor 18
Details:
   Go from floor 12 to floor 11
   Go from floor 11 to floor 18
   #Movement for this request:8
   #Request:2
   Total #Movement:19
Current Position:18



Lift Operation
Previous Position:18
Request:Floor 13 to Floor 8
Details:
   Go from floor 18 to floor 13
   Go from floor 13 to floor 8
   #Movement for this request:10
   #Request:3
   Total #Movement:29
Current Position:8



Lift Operation
Previous Position:8
Request:Floor 10 to Floor 18
Details:
   Go from floor 8 to floor 10
   Go from floor 10 to floor 18
   #Movement for this request:10
   #Request:4
   Total #Movement:39
Current Position:18



Lift Operation
Previous Position:18
Request:Floor 10 to Floor 14
Details:
   Go from floor 18 to floor 10
   Go from floor 10 to floor 14
   #Movement for this request:12
   #Request:5
   Total #Movement:51
Current Position:14



Lift Operation
Previous Position:14
Request:Floor 20 to Floor 18
Details:
   Go from floor 14 to floor 20
   Go from floor 20 to floor 18
   #Movement for this request:8
   #Request:6
   Total #Movement:59
Current Position:18
```

```
New Lift Request from 11 to 16
Request No:13
_____________________________________



_____________________________________
New Lift Request from 14 to 2
Request No:14
_____________________________________



Lift Operation
Previous Position:18
Request:Floor 11 to Floor 16
Details:
   Go from floor 18 to floor 11
   Go from floor 11 to floor 16
   #Movement for this request:12
   #Request:7
   Total #Movement:71
Current Position:16



_____________________________________
New Lift Request from 5 to 17
Request No:15
_____________________________________



_____________________________________
New Lift Request from 3 to 20
Request No:16
_____________________________________



_____________________________________
New Lift Request from 15 to 5
Request No:17
_____________________________________



_____________________________________
New Lift Request from 3 to 4
Request No:18
_____________________________________



Lift Operation
Previous Position:1
Request:Floor 14 to Floor 2
Details:
   Go from floor 1 to floor 14
   Go from floor 14 to floor 2
   #Movement for this request:25
   #Request:1
   Total #Movement:25
Current Position:2



Lift Operation
Previous Position:2
Request:Floor 5 to Floor 17
Details:
   Go from floor 2 to floor 5
   Go from floor 5 to floor 17
   #Movement for this request:15
   #Request:2
   Total #Movement:40
Current Position:17
```

```
Lift Operation
Previous Position:17
Request:Floor 3 to Floor 20
Details:
   Go from floor 17 to floor 3
   Go from floor 3 to floor 20
   #Movement for this request:31
   #Request:3
   Total #Movement:71
Current Position:20



Lift Operation
Previous Position:20
Request:Floor 15 to Floor 5
Details:
   Go from floor 20 to floor 15
   Go from floor 15 to floor 5
   #Movement for this request:15
   #Request:4
   Total #Movement:86
Current Position:5



Lift Operation
Previous Position:5
Request:Floor 3 to Floor 4
Details:
   Go from floor 5 to floor 3
   Go from floor 3 to floor 4
   #Movement for this request:3
   #Request:5
   Total #Movement:89
Current Position:4



_____________________________________
New Lift Request from 15 to 17
Request No:19
_____________________________________



_____________________________________
New Lift Request from 1 to 5
Request No:20
_____________________________________



_____________________________________
New Lift Request from 4 to 11
Request No:21
_____________________________________



_____________________________________
New Lift Request from 15 to 8
Request No:22
_____________________________________



Lift Operation
Previous Position:4
Request:Floor 15 to Floor 17
Details:
   Go from floor 4 to floor 15
   Go from floor 15 to floor 17
   #Movement for this request:13
   #Request:6
   Total #Movement:102
Current Position:17
```

```
Lift Operation
Previous Position:17
Request:Floor 1 to Floor 5
Details:
   Go from floor 17 to floor 1
   Go from floor 1 to floor 5
   #Movement for this request:20
   #Request:7
   Total #Movement:122
Current Position:5



Lift Operation
Previous Position:5
Request:Floor 4 to Floor 11
Details:
   Go from floor 5 to floor 4
   Go from floor 4 to floor 11
   #Movement for this request:8
   #Request:8
   Total #Movement:130
Current Position:11



Lift Operation
Previous Position:11
Request:Floor 15 to Floor 8
Details:
   Go from floor 11 to floor 15
   Go from floor 15 to floor 8
   #Movement for this request:11
   #Request:9
   Total #Movement:141
Current Position:8



_____________________________________
New Lift Request from 7 to 11
Request No:23
_____________________________________



_____________________________________
New Lift Request from 16 to 11
Request No:24
_____________________________________



Lift Operation
Previous Position:20
Request:Floor 7 to Floor 11
Details:
   Go from floor 20 to floor 7
   Go from floor 7 to floor 11
   #Movement for this request:17
   #Request:7
   Total #Movement:124
Current Position:11



Lift Operation
Previous Position:11
Request:Floor 16 to Floor 11
Details:
   Go from floor 11 to floor 16
   Go from floor 16 to floor 11
   #Movement for this request:10
   #Request:8
   Total #Movement:134
Current Position:11
```

```
_____________________________________
New Lift Request from 17 to 11
Request No:25
_____________________________________



_____________________________________
New Lift Request from 5 to 6
Request No:26
_____________________________________



_____________________________________
New Lift Request from 5 to 14
Request No:27
_____________________________________



_____________________________________
New Lift Request from 7 to 15
Request No:28
_____________________________________



_____________________________________
New Lift Request from 20 to 4
Request No:29
_____________________________________



_____________________________________
New Lift Request from 2 to 13
Request No:30
_____________________________________


Lift Operation
Previous Position:8
Request:Floor 17 to Floor 11
Details:
   Go from floor 8 to floor 17
   Go from floor 17 to floor 11
   #Movement for this request:15
   #Request:10
   Total #Movement:156
Current Position:11


Lift Operation
Previous Position:11
Request:Floor 5 to Floor 6
Details:
   Go from floor 11 to floor 5
   Go from floor 5 to floor 6
   #Movement for this request:7
   #Request:11
   Total #Movement:163
Current Position:6


Lift Operation
Previous Position:6
Request:Floor 5 to Floor 14
Details:
   Go from floor 6 to floor 5
   Go from floor 5 to floor 14
   #Movement for this request:10
   #Request:12
   Total #Movement:173
Current Position:14
```

```
Lift Operation
Previous Position:14
Request:Floor 7 to Floor 15
Details:
   Go from floor 14 to floor 7
   Go from floor 7 to floor 15
   #Movement for this request:15
   #Request:13
   Total #Movement:188
Current Position:15


Lift Operation
Previous Position:15
Request:Floor 20 to Floor 4
Details:
   Go from floor 15 to floor 20
   Go from floor 20 to floor 4
   #Movement for this request:21
   #Request:14
   Total #Movement:209
Current Position:4


Lift Operation
Previous Position:4
Request:Floor 2 to Floor 13
Details:
   Go from floor 4 to floor 2
   Go from floor 2 to floor 13
   #Movement for this request:13
   #Request:15
   Total #Movement:222
Current Position:13


_____________________________________
New Lift Request from 10 to 2
Request No:31
_____________________________________



_____________________________________
New Lift Request from 5 to 10
Request No:32
_____________________________________



_____________________________________
New Lift Request from 17 to 18
Request No:33
_____________________________________



_____________________________________
New Lift Request from 5 to 4
Request No:34
_____________________________________



_____________________________________
New Lift Request from 12 to 13
Request No:35
_____________________________________
```

```
Lift Operation
Previous Position:13
Request:Floor 10 to Floor 2
Details:
   Go from floor 13 to floor 10
   Go from floor 10 to floor 2
   #Movement for this request:11
   #Request:16
   Total #Movement:233
Current Position:2



Lift Operation
Previous Position:2
Request:Floor 5 to Floor 10
Details:
   Go from floor 2 to floor 5
   Go from floor 5 to floor 10
   #Movement for this request:8
   #Request:17
   Total #Movement:241
Current Position:10



Lift Operation
Previous Position:10
Request:Floor 17 to Floor 18
Details:
   Go from floor 10 to floor 17
   Go from floor 17 to floor 18
   #Movement for this request:8
   #Request:18
   Total #Movement:249
Current Position:18



Lift Operation
Previous Position:18
Request:Floor 5 to Floor 4
Details:
   Go from floor 18 to floor 5
   Go from floor 5 to floor 4
   #Movement for this request:14
   #Request:19
   Total #Movement:263
Current Position:4



Lift Operation
Previous Position:4
Request:Floor 12 to Floor 13
Details:
   Go from floor 4 to floor 12
   Go from floor 12 to floor 13
   #Movement for this request:9
   #Request:20
   Total #Movement:272
Current Position:13



_____________________________________
New Lift Request from 4 to 9
Request No:36
_____________________________________



_____________________________________
New Lift Request from 10 to 6
Request No:37
_____________________________________
```

```
Lift Operation
Previous Position:11
Request:Floor 4 to Floor 9
Details:
   Go from floor 11 to floor 4
   Go from floor 4 to floor 9
   #Movement for this request:12
   #Request:9
   Total #Movement:146
Current Position:9



Lift Operation
Previous Position:9
Request:Floor 10 to Floor 6
Details:
   Go from floor 9 to floor 10
   Go from floor 10 to floor 6
   #Movement for this request:5
   #Request:10
   Total #Movement:151
Current Position:6



_____________________________________
New Lift Request from 19 to 7
Request No:38
_____________________________________



_____________________________________
New Lift Request from 2 to 12
Request No:39
_____________________________________



_____________________________________
New Lift Request from 17 to 14
Request No:40
_____________________________________



_____________________________________
New Lift Request from 15 to 9
Request No:41
_____________________________________



_____________________________________
New Lift Request from 5 to 12
Request No:42
_____________________________________



_____________________________________
New Lift Request from 14 to 19
Request No:43
_____________________________________



Lift Operation
Previous Position:16
Request:Floor 19 to Floor 7
Details:
   Go from floor 16 to floor 19
   Go from floor 19 to floor 7
   #Movement for this request:15
   #Request:8
   Total #Movement:86
Current Position:7
```

```
Lift Operation
Previous Position:7
Request:Floor 2 to Floor 12
Details:
   Go from floor 7 to floor 2
   Go from floor 2 to floor 12
   #Movement for this request:15
   #Request:9
   Total #Movement:101
Current Position:12


Lift Operation
Previous Position:12
Request:Floor 17 to Floor 14
Details:
   Go from floor 12 to floor 17
   Go from floor 17 to floor 14
   #Movement for this request:8
   #Request:10
   Total #Movement:109
Current Position:14


Lift Operation
Previous Position:14
Request:Floor 15 to Floor 9
Details:
   Go from floor 14 to floor 15
   Go from floor 15 to floor 9
   #Movement for this request:7
   #Request:11
   Total #Movement:116
Current Position:9


Lift Operation
Previous Position:9
Request:Floor 5 to Floor 12
Details:
   Go from floor 9 to floor 5
   Go from floor 5 to floor 12
   #Movement for this request:11
   #Request:12
   Total #Movement:127
Current Position:12


Lift Operation
Previous Position:12
Request:Floor 14 to Floor 19
Details:
   Go from floor 12 to floor 14
   Go from floor 14 to floor 19
   #Movement for this request:7
   #Request:13
   Total #Movement:134
Current Position:19


_____________________________________
New Lift Request from 20 to 14
Request No:44
_____________________________________


_____________________________________
New Lift Request from 17 to 20
Request No:45
_____________________________________
```

```
Lift Operation
Previous Position:19
Request:Floor 20 to Floor 14
Details:
   Go from floor 19 to floor 20
   Go from floor 20 to floor 14
   #Movement for this request:7
   #Request:14
   Total #Movement:141
Current Position:14


Lift Operation
Previous Position:14
Request:Floor 17 to Floor 20
Details:
   Go from floor 14 to floor 17
   Go from floor 17 to floor 20
   #Movement for this request:6
   #Request:15
   Total #Movement:147
Current Position:20


_____________________________________
New Lift Request from 3 to 12
Request No:46
_____________________________________


_____________________________________
New Lift Request from 5 to 12
Request No:47
_____________________________________


_____________________________________
New Lift Request from 3 to 4
Request No:48
_____________________________________


_____________________________________
New Lift Request from 17 to 12
Request No:49
_____________________________________


_____________________________________
New Lift Request from 20 to 3
Request No:50
_____________________________________


Lift Operation
Previous Position:20
Request:Floor 3 to Floor 12
Details:
   Go from floor 20 to floor 3
   Go from floor 3 to floor 12
   #Movement for this request:26
   #Request:16
   Total #Movement:173
Current Position:12
```

```
Lift Operation
Previous Position:12
Request:Floor 5 to Floor 12
Details:
   Go from floor 12 to floor 5
   Go from floor 5 to floor 12
   #Movement for this request:14
   #Request:17
   Total #Movement:187
Current Position:12



Lift Operation
Previous Position:12
Request:Floor 3 to Floor 4
Details:
   Go from floor 12 to floor 3
   Go from floor 3 to floor 4
   #Movement for this request:10
   #Request:18
   Total #Movement:197
Current Position:4



Lift Operation
Previous Position:4
Request:Floor 17 to Floor 12
Details:
   Go from floor 4 to floor 17
   Go from floor 17 to floor 12
   #Movement for this request:18
   #Request:19
   Total #Movement:215
Current Position:12



Lift Operation
Previous Position:12
Request:Floor 20 to Floor 3
Details:
   Go from floor 12 to floor 20
   Go from floor 20 to floor 3
   #Movement for this request:25
   #Request:20
   Total #Movement:240
Current Position:3
```

Known Issues With Part A

For some combinations of buffer size and elevator duration, the program would get stuck after entering the 47th or 48th request in the buffer. This can be replicated by entering 4 as the buffer size and 1 as the duration e.g. **./lift_sim_A 4 1**

Due to time constraints error/file checking was not implemented. The program will only work if both ‘sim_input’ and ‘sim_output’ files are present and if ‘sim_input’ has no invalid lines.

Part B

For this program I used fork() to create new processes. The parent process calls fork and then runs the request function which acts as the producer adding items to the buffer from the file. The first child created calls fork() again and it’s child also calls fork() ending up with the 3 processes (excluding the initial process) that acts as the 3 lifts.

I used two pairs of semaphores, one to indicate whether the buffer is at capacity or empty and another to indicate when at item has been added or removed from the buffer.

The functions used by the processes is similar to the ones in Part A.

Lift_sim_B.c

```
#include <stdio.h>
#include <strings.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <time.h>
#include <semaphore.h>
#include "lift_sim_B.h"
#include "Queue.h"

sem_t notEmpty;
sem_t notFull;
sem_t pmut;
sem_t cmut;

int main( int argsNo, char** args )
{
   if( argsNo < 3 )
   {
      printf( "\nPlease specify buffer size and time required.");
   }
   else
   {
      FILE* sim_input = fopen( "sim_input", "r" );
      FILE* sim_output = fopen( "sim_output", "a" );
      int totalSoFar = 0;
      /* totalSoFar: Incremented everytime something added to buffer */
      Queue* buffer = newQueue( atoi( args[1] ) );
      struct filesAndBuffer threadArgs;
      int shm_id;
      pid_t pid;

      if( sim_input == NULL || sim_output == NULL )
      {
         printf( "\nsim_input or sim_output file is missing.\n");
         printf( "Please create those files first.\n\n\n");
      }
      else
      {
         /* Arguments to pass to each thread:   */
         /* - Pointers to both files, buffer, and total items to buffer so far */
         /* - Integer of how long a lift should take */
         threadArgs.sim_input = sim_input;
         threadArgs.sim_output = sim_output;
         threadArgs.buffer = buffer;
         threadArgs.totalSoFar = &totalSoFar;
         threadArgs.liftTime = atoi( args[2] );

         shm_id = shmget(2009, 2048, 0);
         sem_init( &notEmpty, 4, 1 );
         sem_init( &notFull, 4, 1 );
         sem_init( &cmut, 4, 1 );
         sem_init( &pmut, 4, 1 );

         printf( "\n\nNow processing. Please Wait.\n\n" );

         pid = fork();

         if( pid != 0 )
         {
            request( threadArgs );
         }
         else
         {
            pid = fork();
            if( pid == 0 )
            {
               fork();
            }
            printf( "\nHELLO\n" );
            lift( threadArgs );
         }
         sem_destroy( &notEmpty );
         sem_destroy( &notFull );
      }

      free( buffer );
      fclose( sim_input );
      fclose( sim_output );
   }

   printf( "\n\n" );
   return 0;
}

void* request( struct filesAndBuffer threadArgs )
{
   int nRead, source, destination;
   FILE* sim_input = threadArgs.sim_input;
   FILE* sim_output = threadArgs.sim_output;
   Queue* buffer = threadArgs.buffer;

   while( !feof( sim_input ) )
   {
      sem_wait( &pmut );
      while( buffer->total == buffer->sizeOfBuffer )
      {
         /* Wait first if buffer is still full */
         sem_post( &notFull );
      }

      nRead = fscanf( sim_input, "%d %d\n", &source, &destination );
      if( nRead == 2 )
      {
         /* Adds to buffer, prints to file, increments total so far, and */
         /* signals that the buffer isn't empty.   */
         insert( buffer, source, destination );
         ( *threadArgs.totalSoFar )++;
         fprintf( sim_output, "\n\n_____________________________________" );
         fprintf( sim_output, "\nNew Lift Request from %d to %d",
                                                   source, destination);
         fprintf( sim_output, "\nRequest No:%d", *threadArgs.totalSoFar );
         fprintf( sim_output, "\n_____________________________________\n\n" );
      }
      else if( nRead != 2 )
      {
         /* Program continues even if there's an invalid line in the file. */
         printf( "Invalid line detected. Line has been ignored." );
      }
      sem_post( &notEmpty );
      sem_post( &pmut );
   }

   printf( "\n\nAlmost Done.\n\n" );

   return NULL;
}

void* lift( struct filesAndBuffer threadArgs )
{
   QueueNode* nextToDo;
   FILE* sim_input = threadArgs.sim_input;
   FILE* sim_output = threadArgs.sim_output;
   Queue* buffer = threadArgs.buffer;
   int currentPosition = 1, requestsReceived = 0;
   int totalMovement = 0, currentMovement = 0;
   /* Integers above are just for the numbers that need to be written to file */
   /* Elevators start at 1 so currentPosition is initialised to 1   */

   while( buffer->total != 0 || !feof( sim_input ) )
   {
      sem_wait( &cmut );
      /* When there are items in the buffer or there will be in the future */
      while( buffer->total == 0 && !feof( sim_input ) )
      {
         /* Wait when there aren't any items in the buffer but */
         /* there will be in the future.  */
         sem_post( &notEmpty );
      }
      if( buffer->total != 0 )
      {
         nextToDo = deQueue( buffer );
         requestsReceived++;
         currentMovement = ( abs( currentPosition - nextToDo->source ) ) +
                         abs( ( nextToDo->destination - nextToDo->source ) );
         totalMovement = totalMovement + currentMovement;
         fprintf( sim_output, "\n\nLift Operation" );
         fprintf( sim_output, "\nPrevious Position:%d",currentPosition );
         fprintf( sim_output, "\nRequest:Floor %d to Floor %d",
                                     nextToDo->source, nextToDo->destination );
         fprintf( sim_output, "\nDetails:" );
         fprintf( sim_output, "\n   Go from floor %d to floor %d",
                                           currentPosition, nextToDo->source );
         fprintf( sim_output, "\n   Go from floor %d to floor %d",
                                     nextToDo->source, nextToDo->destination );
         fprintf( sim_output, "\n   #Movement for this request:%d",
                                                         currentMovement );
         fprintf( sim_output, "\n   #Request:%d", requestsReceived );
         fprintf( sim_output, "\n   Total #Movement:%d", totalMovement );
         currentPosition = nextToDo->destination;
         fprintf( sim_output, "\nCurrent Position:%d\n\n", currentPosition );

         sleep( threadArgs.liftTime );
         /* Send signal that there's now one less item in buffer. */
      }
      sem_post( &cmut );
      sem_post( &notFull );
   }

   return NULL;
}
```

Lift_sim_B.h

```
#include "Queue.h"
struct filesAndBuffer
{
   FILE* sim_input;
   FILE* sim_output;
   Queue* buffer;
   int* totalSoFar;
   int liftTime;
};
void* request( struct filesAndBuffer threadArgs );
void* lift( struct filesAndBuffer threadArgs );
```

Part B Issues

I was not able to correctly implement the shared memory requirement of Part B therefore it could not print anything to the sim_out file.

Source Codes Shared Between Parts A and B

Queue.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <pthread.h>
#include "Queue.h"

Queue* newQueue( int sizeOfBuffer )
{
   /*Creates a queue*/
   Queue* newQueue;
   newQueue = (Queue*)malloc(sizeof(Queue));
   (*newQueue).head = NULL;
   (*newQueue).tail = NULL;
   (*newQueue).sizeOfBuffer = sizeOfBuffer;
   (*newQueue).total = 0;
   return newQueue;
}

void insert( struct Queue* inQueue, int inSource, int inDestination )
{
   /*Creates a struct with the value*/
   QueueNode* newNode;

   newNode = (QueueNode*)malloc(sizeof(QueueNode));
   newNode->source = inSource;
   newNode->destination = inDestination;
   newNode->next = NULL;

   if( inQueue->total < inQueue->sizeOfBuffer )
   {
      if( inQueue->head == NULL )
      {
      inQueue->head = newNode;
      inQueue->tail = newNode;
      }
      else
      {
         inQueue->tail->next = newNode;
         inQueue->tail = newNode;
      }
      inQueue->total = inQueue->total + 1;
   }
}

QueueNode* deQueue( struct Queue* inQueue )
{
   QueueNode* head = inQueue->head;
   QueueNode* temp;

   temp = head;

   if( inQueue->head->next != NULL )
   {
      inQueue->head = inQueue->head->next;
   }
   else
   {
      inQueue->head = NULL;
   }
   inQueue->total = inQueue->total - 1;

   return temp;
}
```

Queue.h

```
#ifndef NODE_H
#define NODE_H
typedef struct QueueNode
{
   int source;
   int destination;
   struct QueueNode* next;
} QueueNode;
#endif

#ifndef HEAD_H
#define HEAD_H
typedef struct Queue
{
   QueueNode* head;
   QueueNode* tail;
   int sizeOfBuffer;
   int total;
} Queue;
#endif

#ifndef QUEUE_H
#define QUEUE_H

/* Returns a Linked List that it created. */
Queue* newQueue( int sizeOfBuffer );

/* Puts nValue into a struct and adds it to the last of
 * the list. */
void insert( struct Queue* queue, int source, int destination );
QueueNode* deQueue( struct Queue* inQueue );
#endif
```

makefile

```
CC = gcc
CFLAGS = -Wall -pedantic -ansi -g -pthread
OBJ = lift_sim_A.o Queue.o
all: lift_sim_A lift_sim_B
lift_sim_A : $(OBJ)
	$(CC) $(OBJ) -lm -pthread -o lift_sim_A
lift_sim_A.o : lift_sim_A.c Queue.h
	$(CC) -c lift_sim_A.c $(CFLAGS)
lift_sim_B : lift_sim_B.o Queue.o
	$(CC) lift_sim_B.o Queue.o -lm -pthread -o lift_sim_B
lift_sim_B.o : lift_sim_B.c Queue.h
	$(CC) -c lift_sim_B.c $(CFLAGS)
Queue.o : Queue.c Queue.h
	$(CC) -c Queue.c $(CFLAGS)
clean:
	rm -f lift_sim_A lift_sim_B lift_sim_A.o lift_sim_B.o Queue.o
```