Due: Thu November 3, 2022 PA04 Zachary Montoya
Submitted: October 29th, 2022

Programming Assignment #04

Experience the race conditions and context switching.

```
/*===========================================================*/
/* race.c --- for playing with ECE437 */
/*===========================================================*/
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

struct {int balance[2];} Bank={{100,100}}; //global variable defined
void* MakeTransactions() { //routine for thread execution

  int i, j, tmp1, tmp2, rint; double dummy;
  for (i=0; i < 100; i++) {
    rint = (rand()%30)-15;

    if (((tmp1=Bank.balance[0])+rint)>=0 && ((tmp2=Bank.balance[1])-rint)>=0) {
      Bank.balance[0] = tmp1 + rint; // Adding rint to index 0
      for (j=0; j < rint*1000; j++) {
        dummy=2.345*8.765/1.234;
      } // spend time on purpose
      Bank.balance[1] = tmp2 - rint; // Removing rint from index 1
    }
  }
  return NULL;
}

int main(int argc, char **argv) {
  int i; void* voidptr=NULL; pthread_t tid[2];

  srand(getpid());
  printf("Init balances A:%d + B:%d ==> %d!\n",
  Bank.balance[0],Bank.balance[1],Bank.balance[0]+Bank.balance[1]);

  for (i=0; i<2; i++) if (pthread_create(&tid[i],NULL,MakeTransactions, NULL)) {
    perror("Error in thread creating\n");
    return(1); }
  for (i=0; i<2; i++) if (pthread_join(tid[i], (void*)&voidptr)) {
    perror("Error in thread joining\n");
    return(1);}

  printf("Let's check the balances A:%d + B:%d ==> %d ?= 200\n",
  Bank.balance[0],Bank.balance[1],Bank.balance[0]+Bank.balance[1]);
  return 0;
}
```

Q1) Compile then run the above code for 20-40 times. Write a paragraph to explain the results.

Q2) Use thread library calls (mutex lock and unlock) to modify the code in Q1) to remove any potential race conditions. Show your modification of the code and explain the outcome with your modification

Q3) Rewrite your code in Q1) replacing threads by processes.

- Instead of creating two threads to call "MakeTransactions", you will use fork() to create a child process. Both parent and child processes will call procedure "MakeTransactions".
- Since two processes will not share a common address space, you will need to rewrite code to allocate "Bank" as a shared variable (by applying shared memory IPC, see Slide M02c)
- Other parts (i.e., set up initial values, print the initial values and balance, and print the ending values and balance) stay the same.
- Show your implementation code in the written report, compile then run your new process-based code for 20-40 times. Write a paragraph to explain if the race condition still exists.

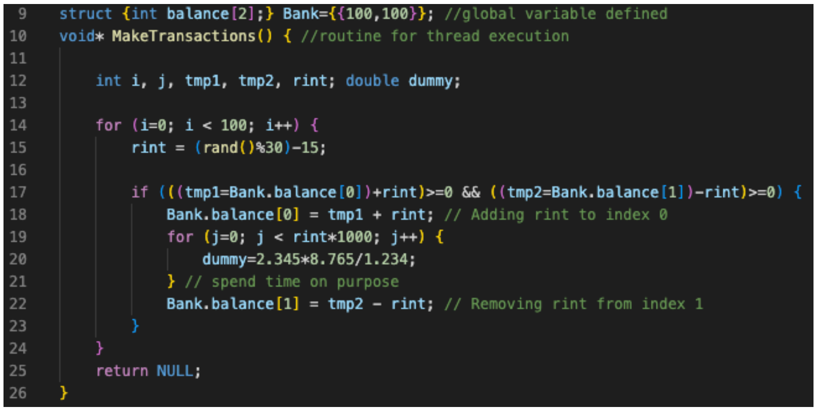
Q4) Use semaphore calls to modify your code in Q3 in order to remove any potential race conditions. Show your modification of the code and explain the outcome with your modification. (You can use either named or unnamed semaphore)

Note: Please submit your source code as well as a written report, which should contain and highlight the modified code segments (you may highlight/using a different font color) to achieve the required functions.

Table of Contents

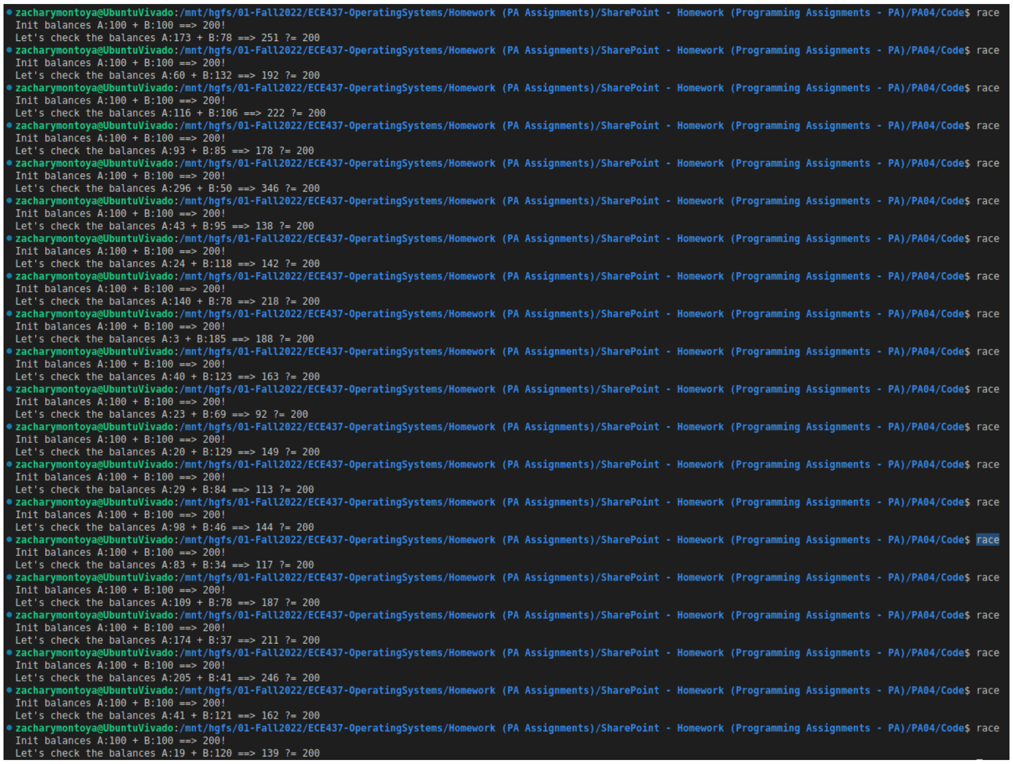
Q1) Compile then run the above code for 20-40 times. Write a paragraph to explain the results.

The outputs of the balances after the transactions executed within the function MakeTransactions() should sum to 200 because theoretically all that is being done is a rint , a random number for all intents and purposes, amount of money is being transferred from Account B to Account A. This is as shown on line 18 and line 22 of the source code screen shot below, see Figure 1. Account A is index 0 of Bank.balance and Account B is index 1 of Bank.balance. However, this is not what occurs, there is erroneous behavior and the values of account A and account B do not summate to 200. They are over or under the amount by an unpredictable value. This is caused by a race condition between threads executing the for-loop, beginning on line 14 of the image below, see Figure 1. The race condition occurs because both threads are attempting to access the balance[0] and balance[1] variables at the same time. This is a problem, especially in this implementation where this could be a software accounting for a person's money. In Figure 2 and Figure 3 the outputs of 40 separate iterations of race.c being ran was captured, these illustrate the erroneous behavior of the source code.



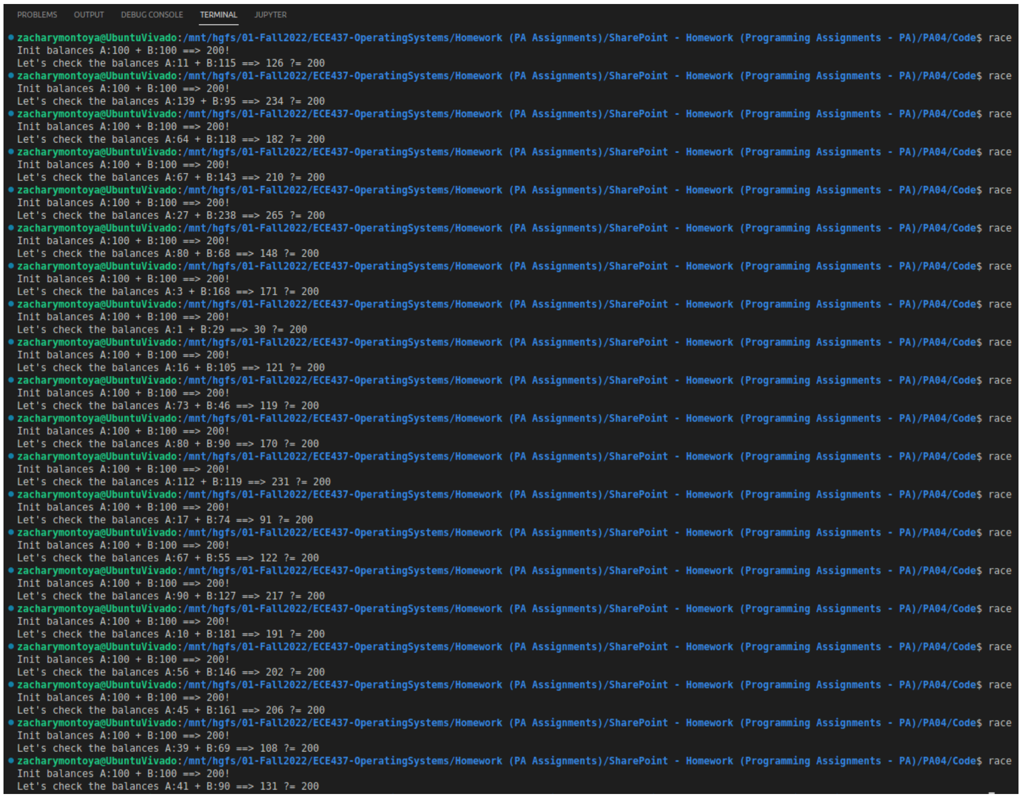
```
 9  struct {int balance[2];} Bank={{100,100}}; //global variable defined
10  void* MakeTransactions() { //routine for thread execution
11
12      int i, j, tmp1, tmp2, rint; double dummy;
13
14      for (i=0; i < 100; i++) {
15          rint = (rand()%30)-15;
16
17          if (((tmp1=Bank.balance[0])+rint)>=0 && ((tmp2=Bank.balance[1])-rint)>=0) {
18              Bank.balance[0] = tmp1 + rint; // Adding rint to index 0
19              for (j=0; j < rint*1000; j++) {
20                  dummy=2.345*8.765/1.234;
21              } // spend time on purpose
22              Bank.balance[1] = tmp2 - rint; // Removing rint from index 1
23          }
24      }
25      return NULL;
26  }
```

Figure 1 - Screenshot of the MakeTransactions() function within the race.c source code



```
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:173 + B:78 ==> 251 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:60 + B:132 ==> 192 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:116 + B:106 ==> 222 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:93 + B:85 ==> 178 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:296 + B:50 ==> 346 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:43 + B:95 ==> 138 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:24 + B:118 ==> 142 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:140 + B:78 ==> 218 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:3 + B:185 ==> 188 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:40 + B:123 ==> 163 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:23 + B:69 ==> 92 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:20 + B:129 ==> 149 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:29 + B:84 ==> 113 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:98 + B:46 ==> 144 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:83 + B:34 ==> 117 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:109 + B:78 ==> 187 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:174 + B:37 ==> 211 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:205 + B:41 ==> 246 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:41 + B:121 ==> 162 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:19 + B:120 ==> 139 ?= 200
```

Figure 2 - First 20 outputs of the original race.c source being ran.

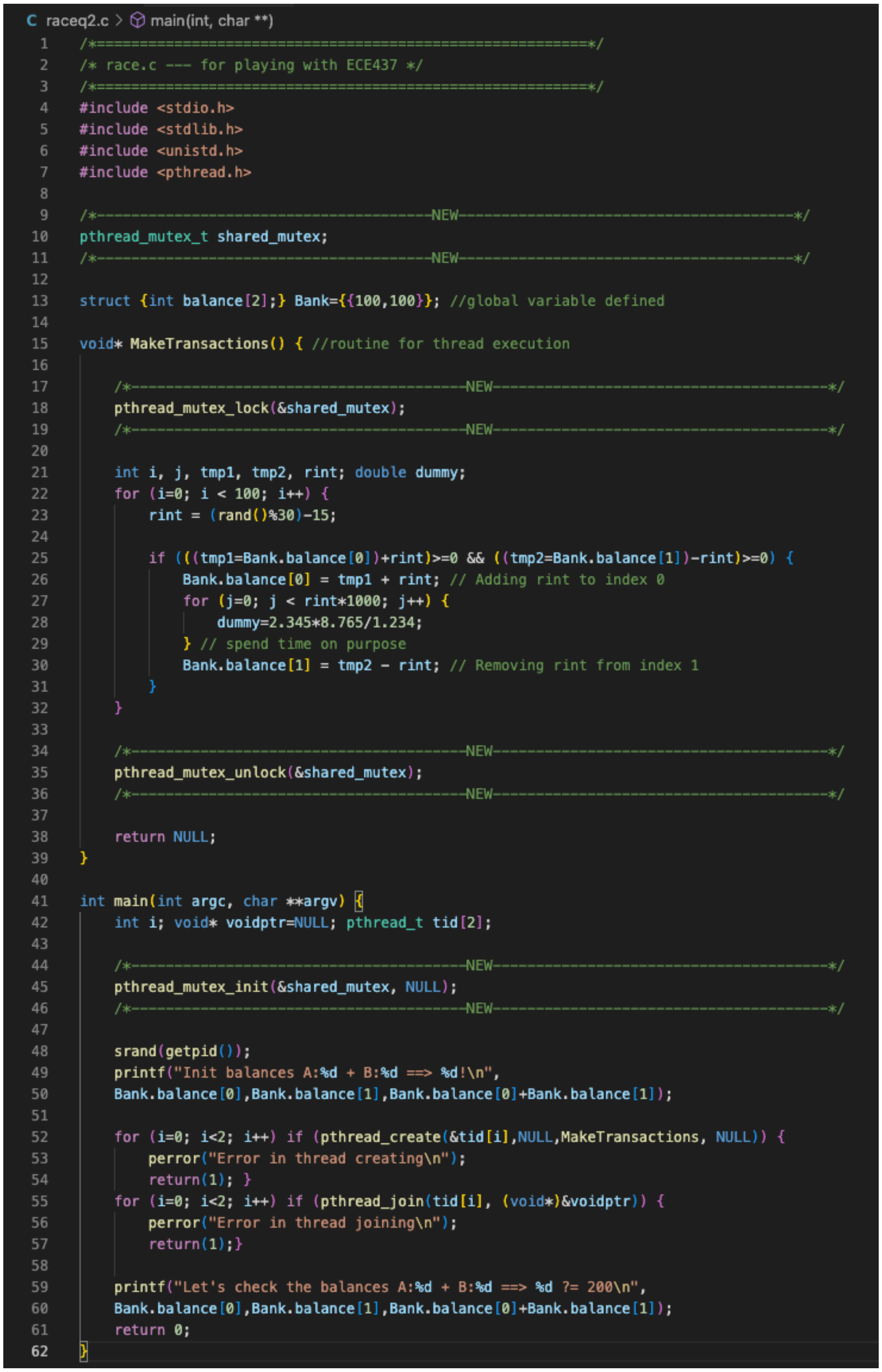


```
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:11 + B:115 ==> 126 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:139 + B:95 ==> 234 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:64 + B:118 ==> 182 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:67 + B:143 ==> 210 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:27 + B:238 ==> 265 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:80 + B:68 ==> 148 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:3 + B:168 ==> 171 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:1 + B:29 ==> 30 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:16 + B:105 ==> 121 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:73 + B:46 ==> 119 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:80 + B:90 ==> 170 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:112 + B:119 ==> 231 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:17 + B:74 ==> 91 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:67 + B:55 ==> 122 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:90 + B:127 ==> 217 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:10 + B:181 ==> 191 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:56 + B:146 ==> 202 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:45 + B:161 ==> 206 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:39 + B:69 ==> 108 ?= 200
zacharymontoya@UbuntuVivado:/mnt/hgfs/01-Fall2022/ECE437-OperatingSystems/Homework (PA Assignments)/SharePoint - Homework (Programming Assignments - PA)/PA04/Code$ race
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:41 + B:90 ==> 131 ?= 200
```

Figure 3 - Second 20 outputs of the original race.c source being ran.

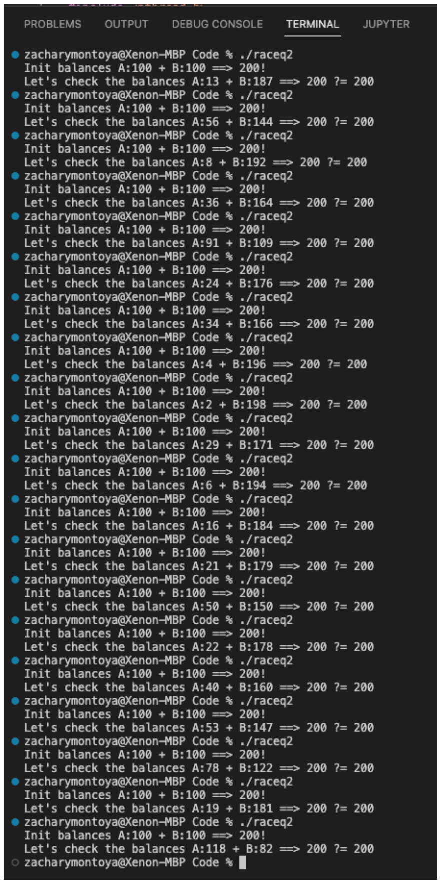
Q2) Use thread library calls (mutex lock and unlock) to modify the code in Q1) to remove any potential race conditions. Show your modification of the code and explain the outcome with your modification

The source code file race.c was modified and resaved as raceq2.c, the changes within this file were to implement a mutex lock using the Thread API mutex lock library calls. The scope of the changes are on code lines 10, 18, 35, and 45 as is shown below, see Figure 4 below. The changes are also bracket with NEW around them for convenience. The entire function MakeTransactions() was treated as a critical section for the same for the sake of simplicity, therefore right at the beginning of this function pthread_mutex_lock referencing the shared_mutex was added. In line with this logic at the end of the MakeTransactions() function a pthread_mutex_unlock referencing the same shared_mutex added. This ensures mutual exclusion for the threads executing the MakeTransactions() function. This approach yielded successful results, the modified source code named raceq2.c was ran 20 times to test repeatability and it passed, see Figure 5 below for image evidence.,



```c
/*=========================================================*/
/* race.c --- for playing with ECE437 */
/*=========================================================*/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>

/*----------------------------------------NEW--------------------------------------*/
pthread_mutex_t shared_mutex;
/*----------------------------------------NEW--------------------------------------*/

struct {int balance[2];} Bank={{100,100}}; //global variable defined

void* MakeTransactions() { //routine for thread execution

    /*----------------------------------------NEW--------------------------------------*/
    pthread_mutex_lock(&shared_mutex);
    /*----------------------------------------NEW--------------------------------------*/

    int i, j, tmp1, tmp2, rint; double dummy;
    for (i=0; i < 100; i++) {
        rint = (rand()%30)-15;

        if (((tmp1=Bank.balance[0])+rint)>=0 && ((tmp2=Bank.balance[1])-rint)>=0) {
            Bank.balance[0] = tmp1 + rint; // Adding rint to index 0
            for (j=0; j < rint*1000; j++) {
                dummy=2.345*8.765/1.234;
            } // spend time on purpose
            Bank.balance[1] = tmp2 - rint; // Removing rint from index 1
        }
    }

    /*----------------------------------------NEW--------------------------------------*/
    pthread_mutex_unlock(&shared_mutex);
    /*----------------------------------------NEW--------------------------------------*/

    return NULL;
}

int main(int argc, char **argv) {
    int i; void* voidptr=NULL; pthread_t tid[2];

    /*----------------------------------------NEW--------------------------------------*/
    pthread_mutex_init(&shared_mutex, NULL);
    /*----------------------------------------NEW--------------------------------------*/

    srand(getpid());
    printf("Init balances A:%d + B:%d ==> %d!\n",
    Bank.balance[0],Bank.balance[1],Bank.balance[0]+Bank.balance[1]);

    for (i=0; i<2; i++) if (pthread_create(&tid[i],NULL,MakeTransactions, NULL)) {
        perror("Error in thread creating\n");
        return(1); }
    for (i=0; i<2; i++) if (pthread_join(tid[i], (void*)&voidptr)) {
        perror("Error in thread joining\n");
        return(1);}

    printf("Let's check the balances A:%d + B:%d ==> %d ?= 200\n",
    Bank.balance[0],Bank.balance[1],Bank.balance[0]+Bank.balance[1]);
    return 0;
}
```

Figure 4 - Revised race.c file for Q2



```
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:13 + B:187 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:56 + B:144 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:8 + B:192 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:36 + B:164 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:91 + B:109 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:24 + B:176 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:34 + B:166 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:4 + B:196 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:2 + B:198 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:29 + B:171 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:6 + B:194 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:16 + B:184 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:21 + B:179 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:50 + B:150 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:22 + B:178 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:40 + B:160 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:53 + B:147 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:78 + B:122 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:19 + B:181 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq2
Init balances A:100 + B:100 ==> 200!
Let's check the balances A:118 + B:82 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % 
```

Figure 5 – 20 Outputs of the code modified for question 2

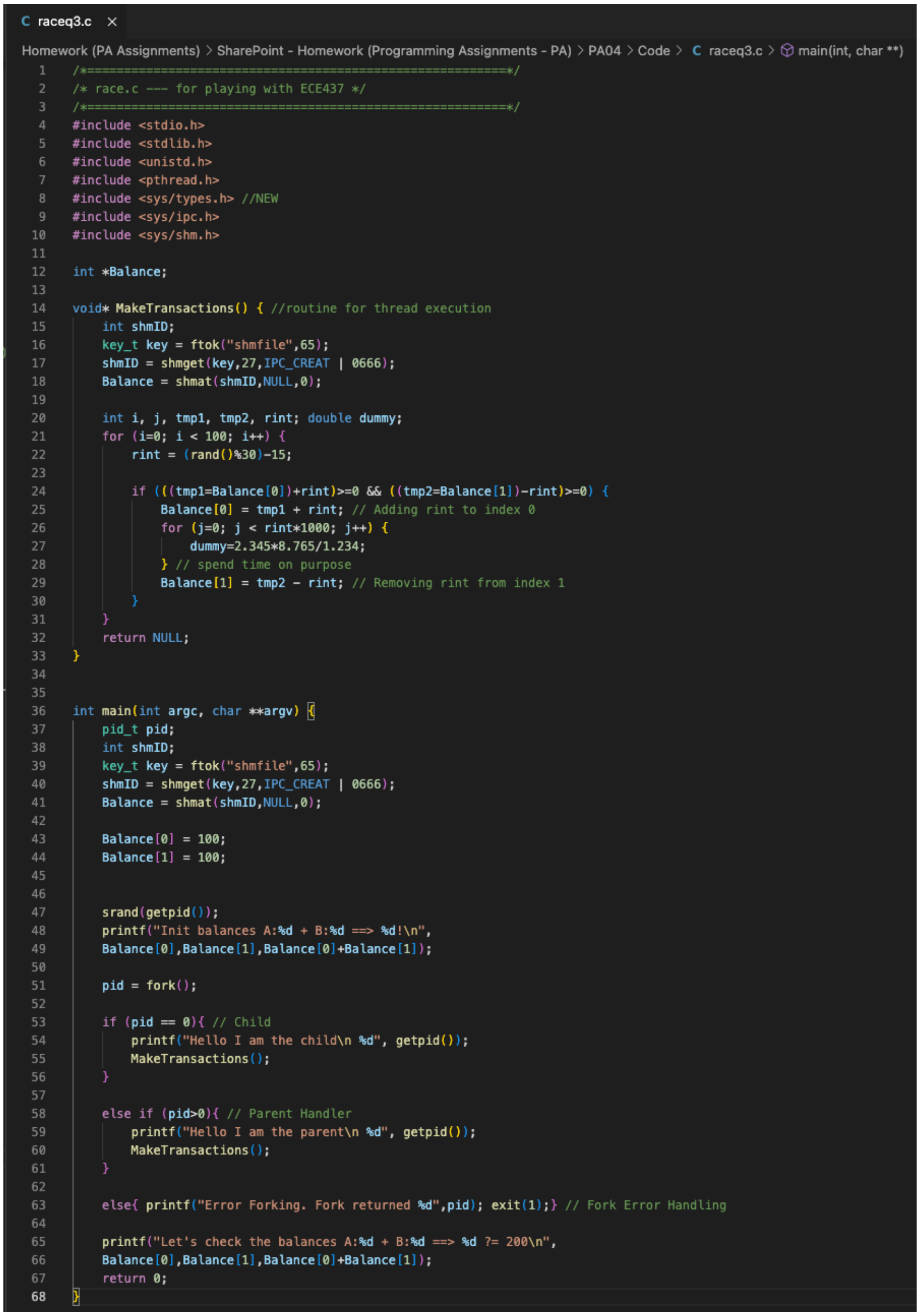
Q3) Rewrite your code in Q1) replacing threads by processes.

- Instead of creating two threads to call “MakeTransactions”, you will use fork() to create a child process. Both parent and child processes will call procedure “MakeTransactions”.
- Since two processes will not share a common address space, you will need to rewrite code to allocate “Bank” as a shared variable (by applying shared memory IPC, see Slide M02c)
- Other parts (i.e., set up initial values, print the initial values and balance, and print the ending values and balance) stay the same.
- Show your implementation code in the written report, compile then run your new process-based code for 20-40 times. Write a paragraph to explain if the race condition still exists.

The code was modified per the criterion described above with the exception: (1) of instead of making the struct a shared variable, a shared variable array named *Balance* was used, and (2) I was interested in delineating the child and parent output, so I put *printf* functions within the conditional branches for the parent and childs. Functionally this achieves the same user requirements as defined above. The code is as shown in the Figure 6 below.

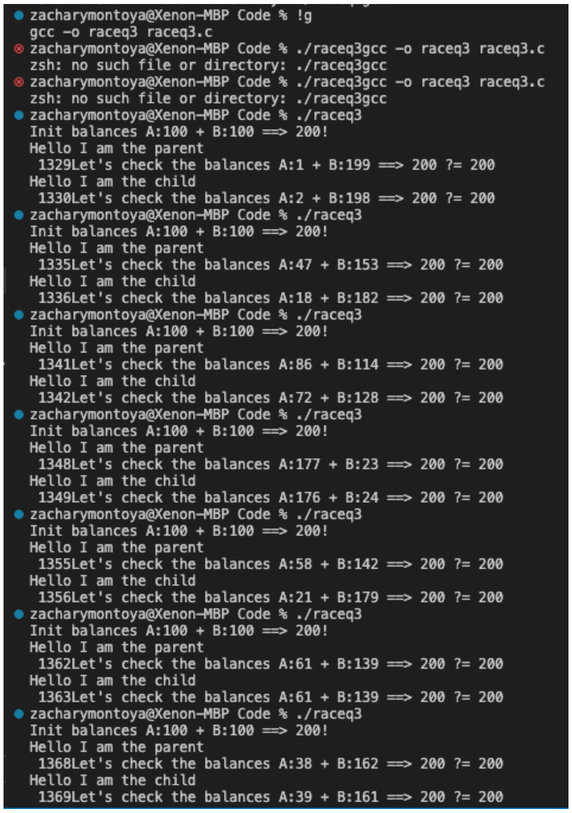
When conducting a repeatability test with a sample of n=20, the code behaved without expediting eradict behavior due to race conditions. Albeit when the repeatability testing was expanded to n=37, the CPU scheduler executed a context switch while the child and parent functions were executing the *MakeTransactions.* This is as shown in, Figure 9. It is evident this is the case because the parent and child both signaled their branch start then the balances were reported, and they were reported incorrectly. The typical output when there isn’t a context switch is the parent will branch, execute MakeTransactions, report balances, child will branch, execute MakeTransactions, and then report balances. The typical output doesn’t yield an error because they are not altering the balances at the same time as they did the 37th time the code was ran.

The terminal was cleared and then reran. Almost immediately the source code was producing race conditions. See Figure 10.


raceq3.c
Homework (PA Assignments) > SharePoint - Homework (Programming Assignments - PA) > PA04 > Code > raceq3.c > main(int, char **)

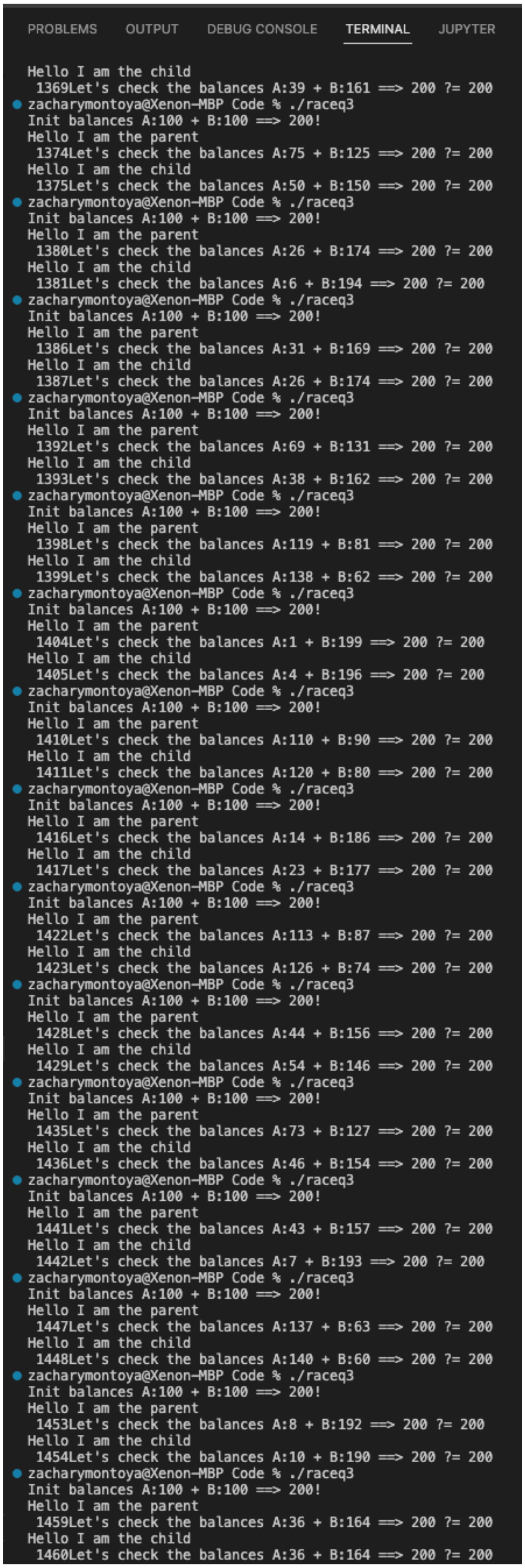
```c
/*=========================================================*/
/* race.c --- for playing with ECE437 */
/*=========================================================*/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <sys/types.h> //NEW
#include <sys/ipc.h>
#include <sys/shm.h>

int *Balance;

void* MakeTransactions() { //routine for thread execution
    int shmID;
    key_t key = ftok("shmfile",65);
    shmID = shmget(key,27,IPC_CREAT | 0666);
    Balance = shmat(shmID,NULL,0);

    int i, j, tmp1, tmp2, rint; double dummy;
    for (i=0; i < 100; i++) {
        rint = (rand()%30)-15;

        if (((tmp1=Balance[0])+rint)>=0 && ((tmp2=Balance[1])-rint)>=0) {
            Balance[0] = tmp1 + rint; // Adding rint to index 0
            for (j=0; j < rint*1000; j++) {
                dummy=2.345*8.765/1.234;
            } // spend time on purpose
            Balance[1] = tmp2 - rint; // Removing rint from index 1
        }
    }
    return NULL;
}


int main(int argc, char **argv) {
    pid_t pid;
    int shmID;
    key_t key = ftok("shmfile",65);
    shmID = shmget(key,27,IPC_CREAT | 0666);
    Balance = shmat(shmID,NULL,0);

    Balance[0] = 100;
    Balance[1] = 100;


    srand(getpid());
    printf("Init balances A:%d + B:%d ==> %d!\n",
    Balance[0],Balance[1],Balance[0]+Balance[1]);

    pid = fork();

    if (pid == 0){ // Child
        printf("Hello I am the child\n %d", getpid());
        MakeTransactions();
    }

    else if (pid>0){ // Parent Handler
        printf("Hello I am the parent\n %d", getpid());
        MakeTransactions();
    }

    else{ printf("Error Forking. Fork returned %d",pid); exit(1);} // Fork Error Handling

    printf("Let's check the balances A:%d + B:%d ==> %d ?= 200\n",
    Balance[0],Balance[1],Balance[0]+Balance[1]);
    return 0;
}
```

Figure 6 - Revised race.c file for Q3



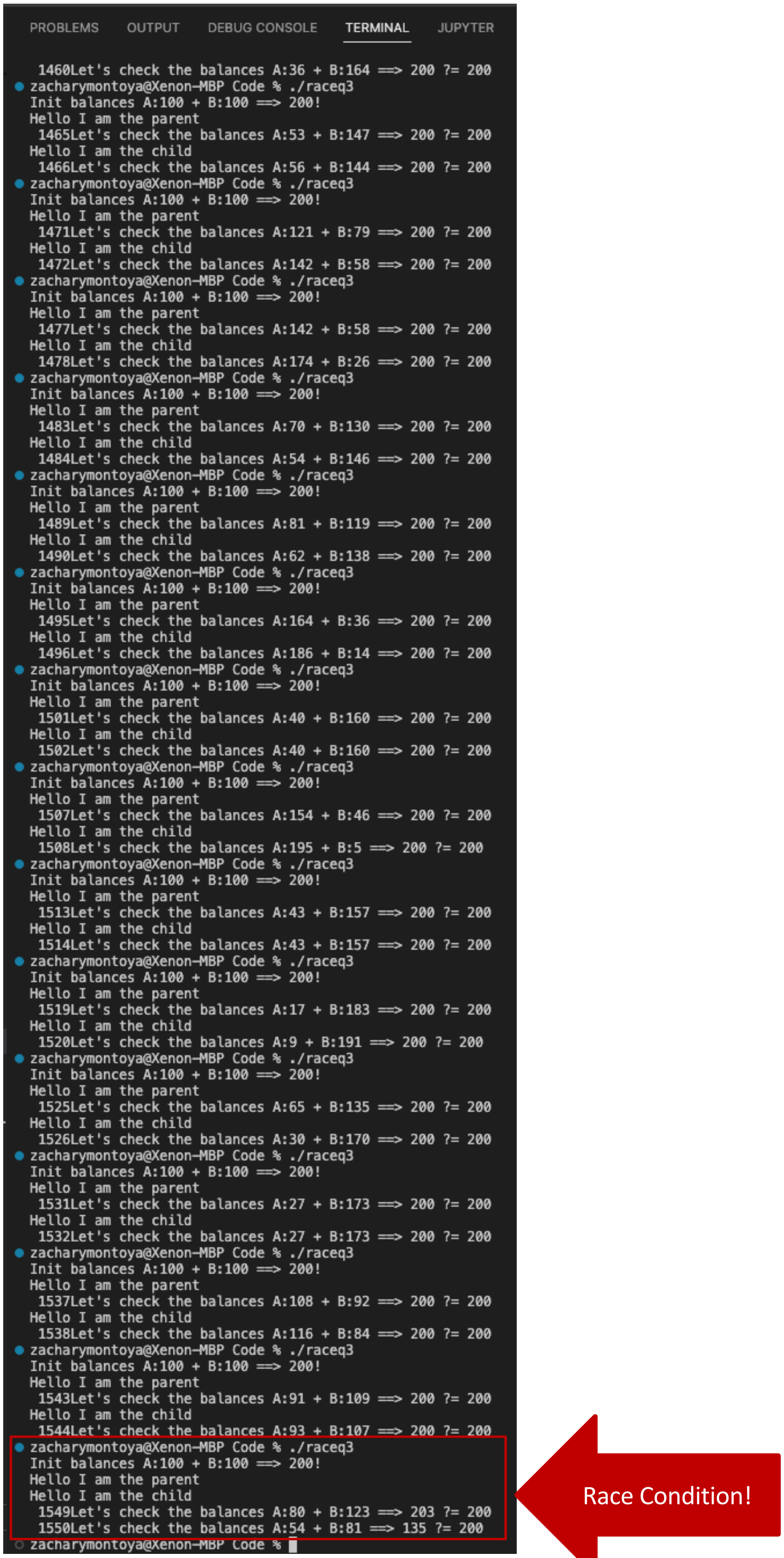
```
  ● zacharymontoya@Xenon-MBP Code % !g
    gcc -o raceq3 raceq3.c
  ⊗ zacharymontoya@Xenon-MBP Code % ./raceq3gcc -o raceq3 raceq3.c
    zsh: no such file or directory: ./raceq3gcc
  ⊗ zacharymontoya@Xenon-MBP Code % ./raceq3gcc -o raceq3 raceq3.c
    zsh: no such file or directory: ./raceq3gcc
1 ● zacharymontoya@Xenon-MBP Code % ./raceq3
    Init balances A:100 + B:100 ==> 200!
    Hello I am the parent
     1329Let's check the balances A:1 + B:199 ==> 200 ?= 200
    Hello I am the child
     1330Let's check the balances A:2 + B:198 ==> 200 ?= 200
2 ● zacharymontoya@Xenon-MBP Code % ./raceq3
    Init balances A:100 + B:100 ==> 200!
    Hello I am the parent
     1335Let's check the balances A:47 + B:153 ==> 200 ?= 200
    Hello I am the child
     1336Let's check the balances A:18 + B:182 ==> 200 ?= 200
3 ● zacharymontoya@Xenon-MBP Code % ./raceq3
    Init balances A:100 + B:100 ==> 200!
    Hello I am the parent
     1341Let's check the balances A:86 + B:114 ==> 200 ?= 200
    Hello I am the child
     1342Let's check the balances A:72 + B:128 ==> 200 ?= 200
4 ● zacharymontoya@Xenon-MBP Code % ./raceq3
    Init balances A:100 + B:100 ==> 200!
    Hello I am the parent
     1348Let's check the balances A:177 + B:23 ==> 200 ?= 200
    Hello I am the child
     1349Let's check the balances A:176 + B:24 ==> 200 ?= 200
5 ● zacharymontoya@Xenon-MBP Code % ./raceq3
    Init balances A:100 + B:100 ==> 200!
    Hello I am the parent
     1355Let's check the balances A:58 + B:142 ==> 200 ?= 200
    Hello I am the child
     1356Let's check the balances A:21 + B:179 ==> 200 ?= 200
6 ● zacharymontoya@Xenon-MBP Code % ./raceq3
    Init balances A:100 + B:100 ==> 200!
    Hello I am the parent
     1362Let's check the balances A:61 + B:139 ==> 200 ?= 200
    Hello I am the child
     1363Let's check the balances A:61 + B:139 ==> 200 ?= 200
7 ● zacharymontoya@Xenon-MBP Code % ./raceq3
    Init balances A:100 + B:100 ==> 200!
    Hello I am the parent
     1368Let's check the balances A:38 + B:162 ==> 200 ?= 200
    Hello I am the child
     1369Let's check the balances A:39 + B:161 ==> 200 ?= 200
```

Figure 7 – First 7 Outputs of the code modified for question 3



```
   Hello I am the child
    1369Let's check the balances A:39 + B:161 ==> 200 ?= 200
8  ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1374Let's check the balances A:75 + B:125 ==> 200 ?= 200
   Hello I am the child
    1375Let's check the balances A:50 + B:150 ==> 200 ?= 200
9  ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1380Let's check the balances A:26 + B:174 ==> 200 ?= 200
   Hello I am the child
    1381Let's check the balances A:6 + B:194 ==> 200 ?= 200
10 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1386Let's check the balances A:31 + B:169 ==> 200 ?= 200
   Hello I am the child
    1387Let's check the balances A:26 + B:174 ==> 200 ?= 200
11 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1392Let's check the balances A:69 + B:131 ==> 200 ?= 200
   Hello I am the child
    1393Let's check the balances A:38 + B:162 ==> 200 ?= 200
12 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1398Let's check the balances A:119 + B:81 ==> 200 ?= 200
   Hello I am the child
    1399Let's check the balances A:138 + B:62 ==> 200 ?= 200
13 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1404Let's check the balances A:1 + B:199 ==> 200 ?= 200
   Hello I am the child
    1405Let's check the balances A:4 + B:196 ==> 200 ?= 200
14 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1410Let's check the balances A:110 + B:90 ==> 200 ?= 200
   Hello I am the child
    1411Let's check the balances A:120 + B:80 ==> 200 ?= 200
15 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1416Let's check the balances A:14 + B:186 ==> 200 ?= 200
   Hello I am the child
    1417Let's check the balances A:23 + B:177 ==> 200 ?= 200
16 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1422Let's check the balances A:113 + B:87 ==> 200 ?= 200
   Hello I am the child
    1423Let's check the balances A:126 + B:74 ==> 200 ?= 200
17 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1428Let's check the balances A:44 + B:156 ==> 200 ?= 200
   Hello I am the child
    1429Let's check the balances A:54 + B:146 ==> 200 ?= 200
18 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1435Let's check the balances A:73 + B:127 ==> 200 ?= 200
   Hello I am the child
    1436Let's check the balances A:46 + B:154 ==> 200 ?= 200
19 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1441Let's check the balances A:43 + B:157 ==> 200 ?= 200
   Hello I am the child
    1442Let's check the balances A:7 + B:193 ==> 200 ?= 200
20 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1447Let's check the balances A:137 + B:63 ==> 200 ?= 200
   Hello I am the child
    1448Let's check the balances A:140 + B:60 ==> 200 ?= 200
21 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1453Let's check the balances A:8 + B:192 ==> 200 ?= 200
   Hello I am the child
    1454Let's check the balances A:10 + B:190 ==> 200 ?= 200
22 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1459Let's check the balances A:36 + B:164 ==> 200 ?= 200
   Hello I am the child
    1460Let's check the balances A:36 + B:164 ==> 200 ?= 200
```

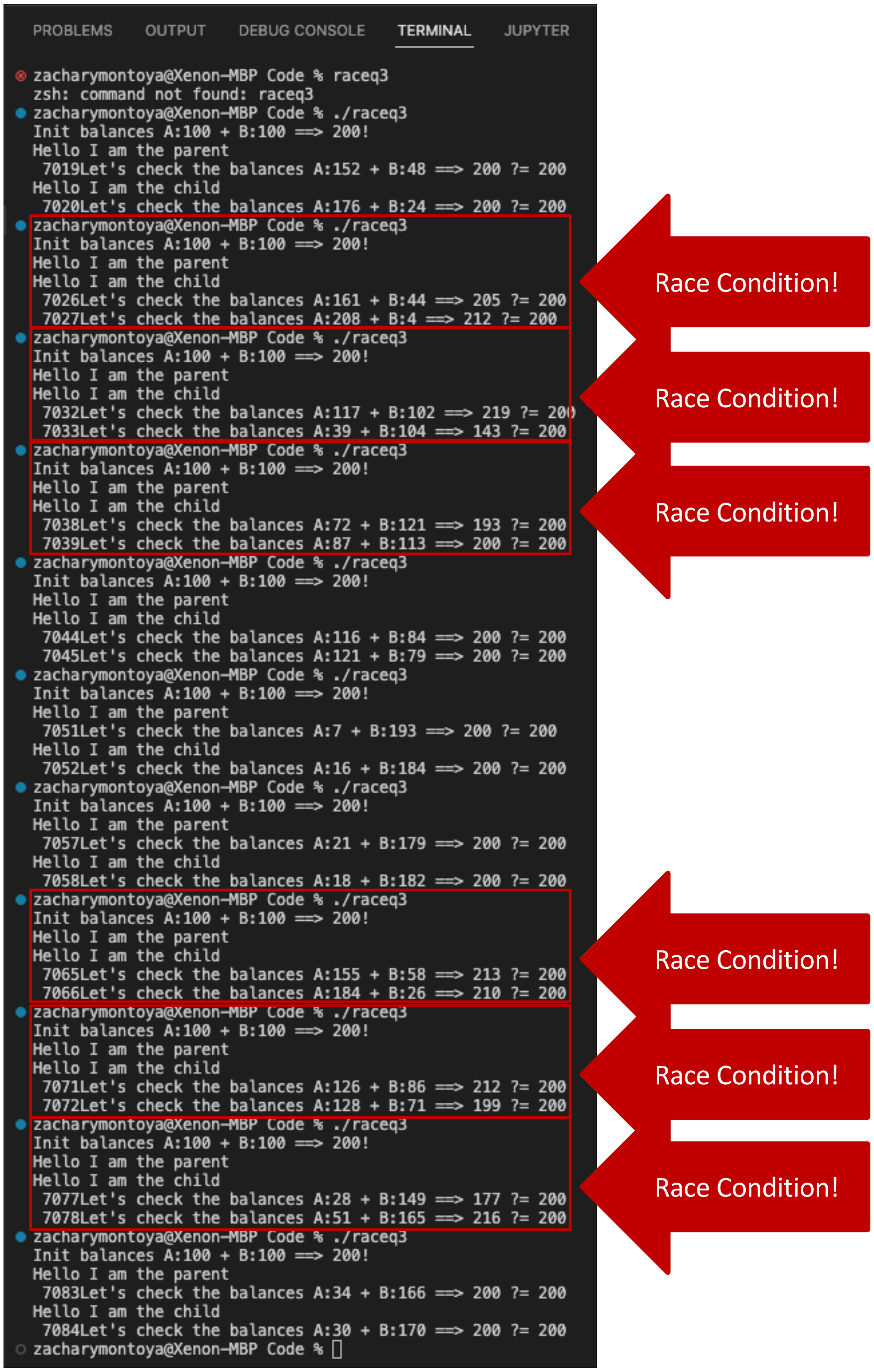
Figure 8 – Second 15 Outputs of the code modified for question 3



```
    1460Let's check the balances A:36 + B:164 ==> 200 ?= 200
23 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1465Let's check the balances A:53 + B:147 ==> 200 ?= 200
   Hello I am the child
    1466Let's check the balances A:56 + B:144 ==> 200 ?= 200
24 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1471Let's check the balances A:121 + B:79 ==> 200 ?= 200
   Hello I am the child
    1472Let's check the balances A:142 + B:58 ==> 200 ?= 200
25 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1477Let's check the balances A:142 + B:58 ==> 200 ?= 200
   Hello I am the child
    1478Let's check the balances A:174 + B:26 ==> 200 ?= 200
26 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1483Let's check the balances A:70 + B:130 ==> 200 ?= 200
   Hello I am the child
    1484Let's check the balances A:54 + B:146 ==> 200 ?= 200
27 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1489Let's check the balances A:81 + B:119 ==> 200 ?= 200
   Hello I am the child
    1490Let's check the balances A:62 + B:138 ==> 200 ?= 200
28 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1495Let's check the balances A:164 + B:36 ==> 200 ?= 200
   Hello I am the child
    1496Let's check the balances A:186 + B:14 ==> 200 ?= 200
29 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1501Let's check the balances A:40 + B:160 ==> 200 ?= 200
   Hello I am the child
    1502Let's check the balances A:40 + B:160 ==> 200 ?= 200
30 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1507Let's check the balances A:154 + B:46 ==> 200 ?= 200
   Hello I am the child
    1508Let's check the balances A:195 + B:5 ==> 200 ?= 200
31 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1513Let's check the balances A:43 + B:157 ==> 200 ?= 200
   Hello I am the child
    1514Let's check the balances A:43 + B:157 ==> 200 ?= 200
32 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1519Let's check the balances A:17 + B:183 ==> 200 ?= 200
   Hello I am the child
    1520Let's check the balances A:9 + B:191 ==> 200 ?= 200
33 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1525Let's check the balances A:65 + B:135 ==> 200 ?= 200
   Hello I am the child
    1526Let's check the balances A:30 + B:170 ==> 200 ?= 200
34 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1531Let's check the balances A:27 + B:173 ==> 200 ?= 200
   Hello I am the child
    1532Let's check the balances A:27 + B:173 ==> 200 ?= 200
35 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1537Let's check the balances A:108 + B:92 ==> 200 ?= 200
   Hello I am the child
    1538Let's check the balances A:116 + B:84 ==> 200 ?= 200
36 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
    1543Let's check the balances A:91 + B:109 ==> 200 ?= 200
   Hello I am the child
    1544Let's check the balances A:93 + B:107 ==> 200 ?= 200
37 ● zacharymontoya@Xenon-MBP Code % ./raceq3
   Init balances A:100 + B:100 ==> 200!
   Hello I am the parent
   Hello I am the child
    1549Let's check the balances A:80 + B:123 ==> 203 ?= 200
    1550Let's check the balances A:54 + B:81 ==> 135 ?= 200
   ○ zacharymontoya@Xenon-MBP Code % ▌
```

Race Condition!

Figure 9 – Last 15 Outputs of the code modified for question 3

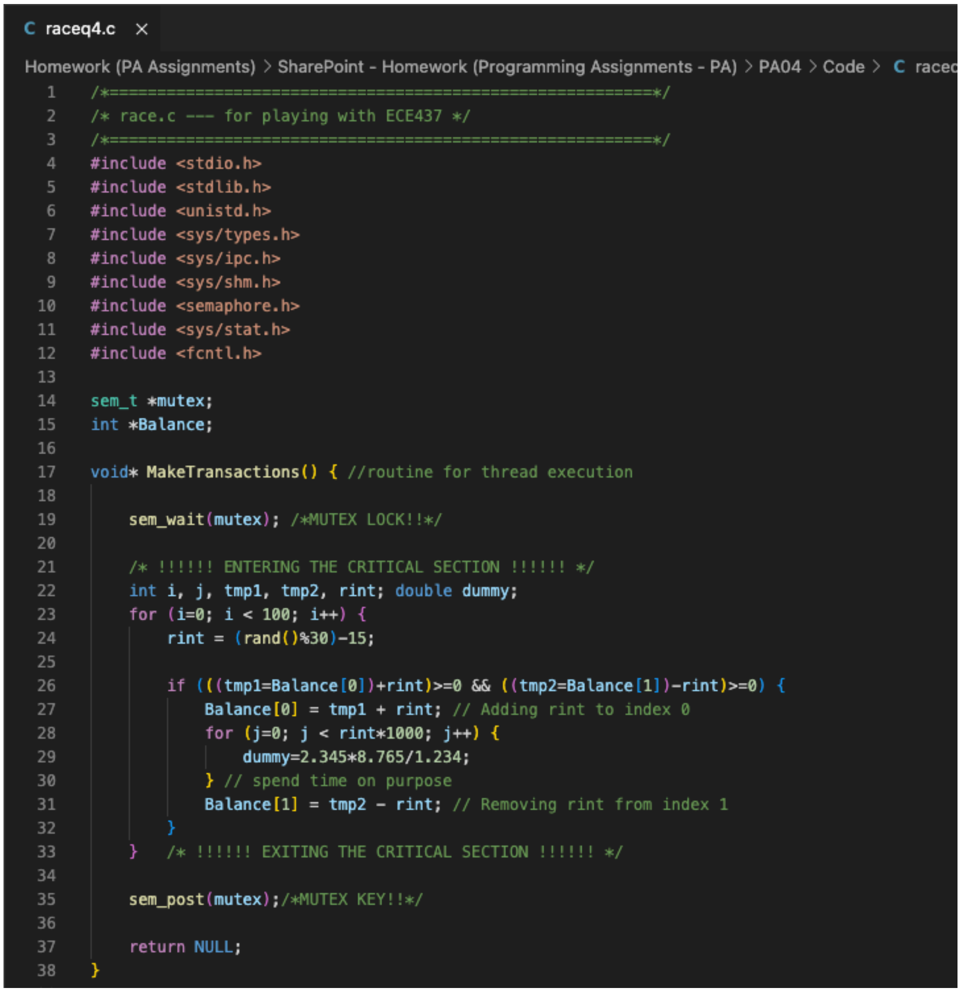


```
zacharymontoya@Xenon-MBP Code % raceq3
zsh: command not found: raceq3
zacharymontoya@Xenon-MBP Code % ./raceq3
Init balances A:100 + B:100 ==> 200!
Hello I am the parent
 7019Let's check the balances A:152 + B:48 ==> 200 ?= 200
Hello I am the child
 7020Let's check the balances A:176 + B:24 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq3
Init balances A:100 + B:100 ==> 200!
Hello I am the parent
Hello I am the child
 7026Let's check the balances A:161 + B:44 ==> 205 ?= 200
 7027Let's check the balances A:208 + B:4 ==> 212 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq3
Init balances A:100 + B:100 ==> 200!
Hello I am the parent
Hello I am the child
 7032Let's check the balances A:117 + B:102 ==> 219 ?= 200
 7033Let's check the balances A:39 + B:104 ==> 143 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq3
Init balances A:100 + B:100 ==> 200!
Hello I am the parent
Hello I am the child
 7038Let's check the balances A:72 + B:121 ==> 193 ?= 200
 7039Let's check the balances A:87 + B:113 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq3
Init balances A:100 + B:100 ==> 200!
Hello I am the parent
Hello I am the child
 7044Let's check the balances A:116 + B:84 ==> 200 ?= 200
 7045Let's check the balances A:121 + B:79 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq3
Init balances A:100 + B:100 ==> 200!
Hello I am the parent
 7051Let's check the balances A:7 + B:193 ==> 200 ?= 200
Hello I am the child
 7052Let's check the balances A:16 + B:184 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq3
Init balances A:100 + B:100 ==> 200!
Hello I am the parent
 7057Let's check the balances A:21 + B:179 ==> 200 ?= 200
Hello I am the child
 7058Let's check the balances A:18 + B:182 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq3
Init balances A:100 + B:100 ==> 200!
Hello I am the parent
Hello I am the child
 7065Let's check the balances A:155 + B:58 ==> 213 ?= 200
 7066Let's check the balances A:184 + B:26 ==> 210 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq3
Init balances A:100 + B:100 ==> 200!
Hello I am the parent
Hello I am the child
 7071Let's check the balances A:126 + B:86 ==> 212 ?= 200
 7072Let's check the balances A:128 + B:71 ==> 199 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq3
Init balances A:100 + B:100 ==> 200!
Hello I am the parent
Hello I am the child
 7077Let's check the balances A:28 + B:149 ==> 177 ?= 200
 7078Let's check the balances A:51 + B:165 ==> 216 ?= 200
zacharymontoya@Xenon-MBP Code % ./raceq3
Init balances A:100 + B:100 ==> 200!
Hello I am the parent
 7083Let's check the balances A:34 + B:166 ==> 200 ?= 200
Hello I am the child
 7084Let's check the balances A:30 + B:170 ==> 200 ?= 200
zacharymontoya@Xenon-MBP Code % 
```

Figure 10 – Rerunning Q3

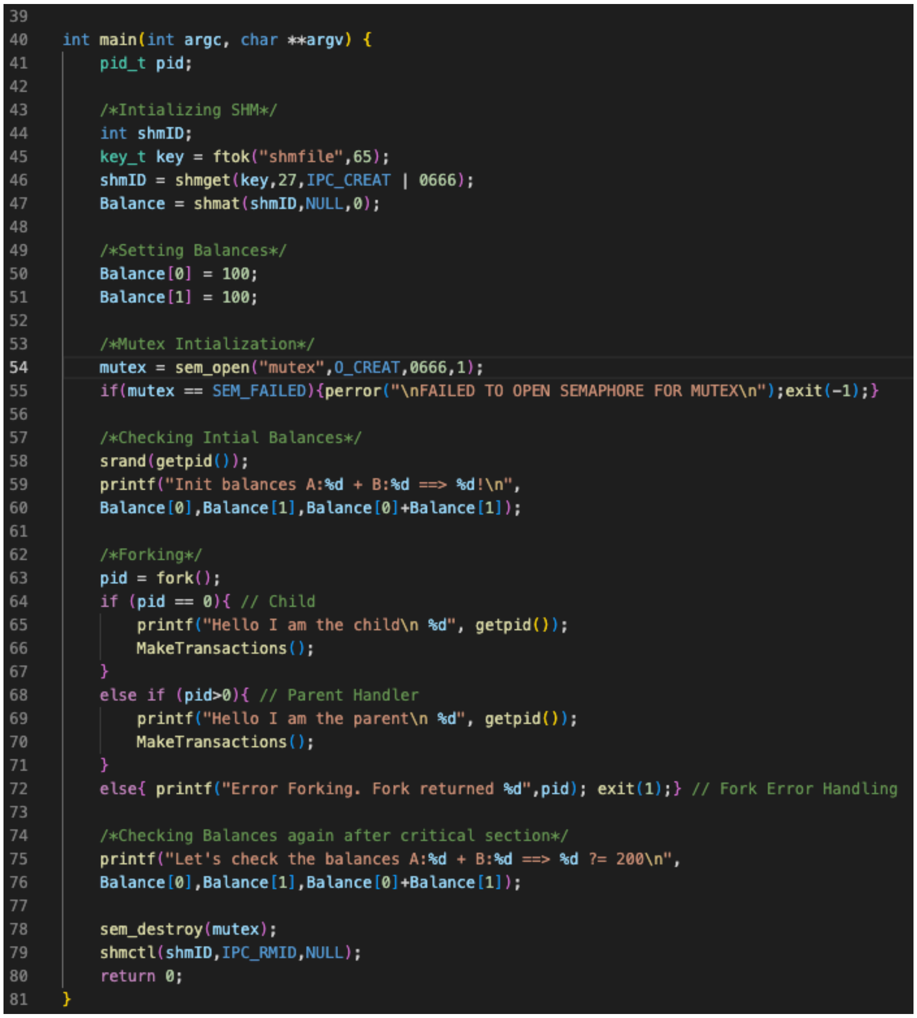
Q4) Use semaphore calls to modify your code in Q3 in order to remove any potential race conditions. Show your modification of the code and explain the outcome with your modification. (You can use either named or unnamed semaphore)

A named binary semaphore was used to achieved mutual exclusion for the critical section, *MakeTransactions*. As Dr. Sun mentioned in lecture it is very important to use a named semaphore here, this code was initially attempted to be implemented used an unnamed semaphore and it did not function as intended. The implementation of the semaphore is as shown in the following figures, see Figure 11 and Figure 12. This source code was tested for reliability with n=40 samples, all 40 sample passed the test and yielded the correct results regardless of any context switching between parent and child processes. This is as shown below, see Figure 13, Figure 14, Figure 15, and Figure 16.


C raceq4.c ×
Homework (PA Assignments) > SharePoint - Homework (Programming Assignments - PA) > PA04 > Code > C racec

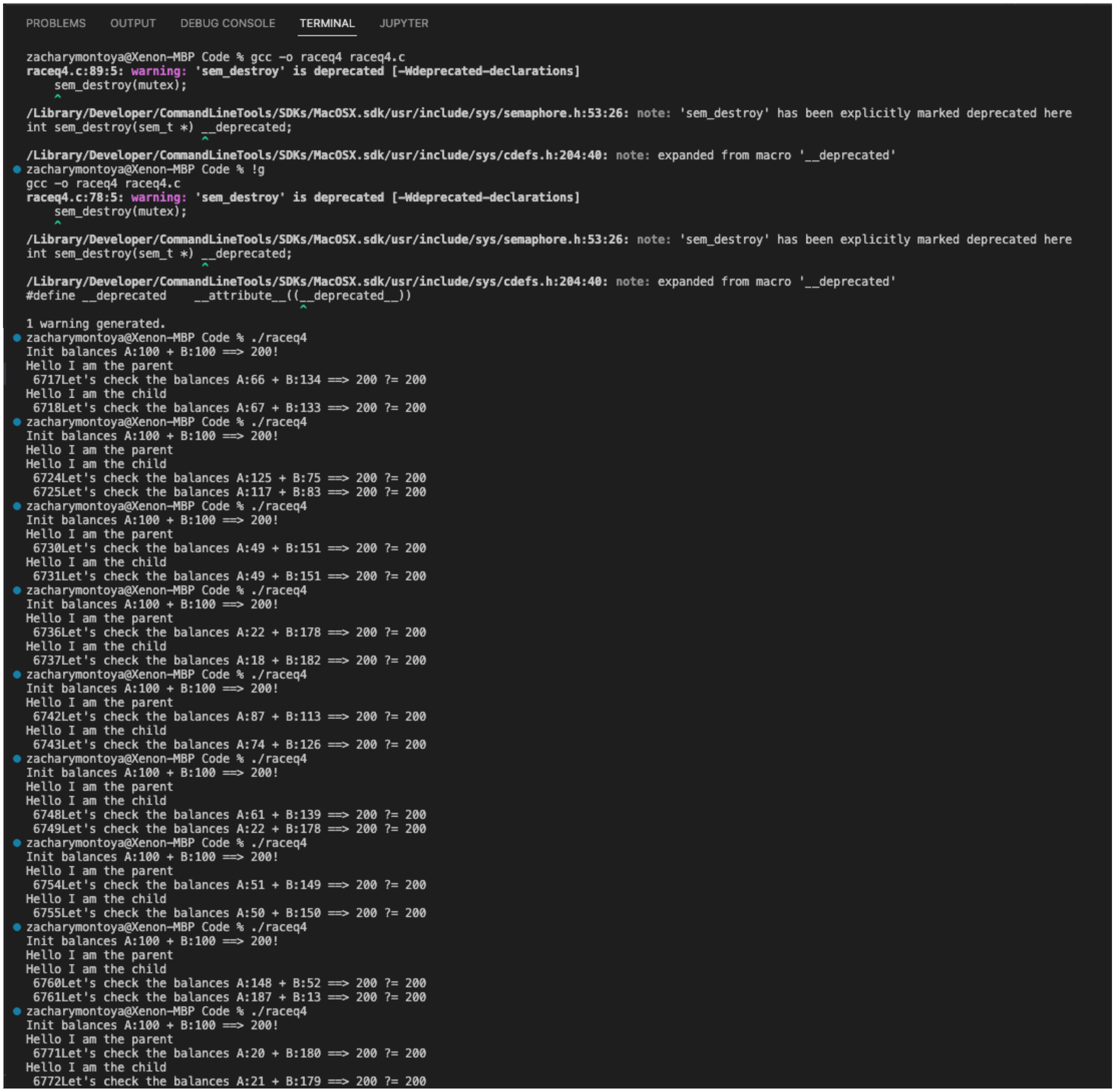
```c
/*=========================================================*/
/* race.c --- for playing with ECE437 */
/*=========================================================*/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <semaphore.h>
#include <sys/stat.h>
#include <fcntl.h>

sem_t *mutex;
int *Balance;

void* MakeTransactions() { //routine for thread execution

    sem_wait(mutex); /*MUTEX LOCK!!*/

    /* !!!!!! ENTERING THE CRITICAL SECTION !!!!!! */
    int i, j, tmp1, tmp2, rint; double dummy;
    for (i=0; i < 100; i++) {
        rint = (rand()%30)-15;

        if (((tmp1=Balance[0])+rint)>=0 && ((tmp2=Balance[1])-rint)>=0) {
            Balance[0] = tmp1 + rint; // Adding rint to index 0
            for (j=0; j < rint*1000; j++) {
                dummy=2.345*8.765/1.234;
            } // spend time on purpose
            Balance[1] = tmp2 - rint; // Removing rint from index 1
        }
    }   /* !!!!!! EXITING THE CRITICAL SECTION !!!!!! */

    sem_post(mutex);/*MUTEX KEY!!*/

    return NULL;
}
```

Figure 11 - Revised race.c file for Q4 1 of 2



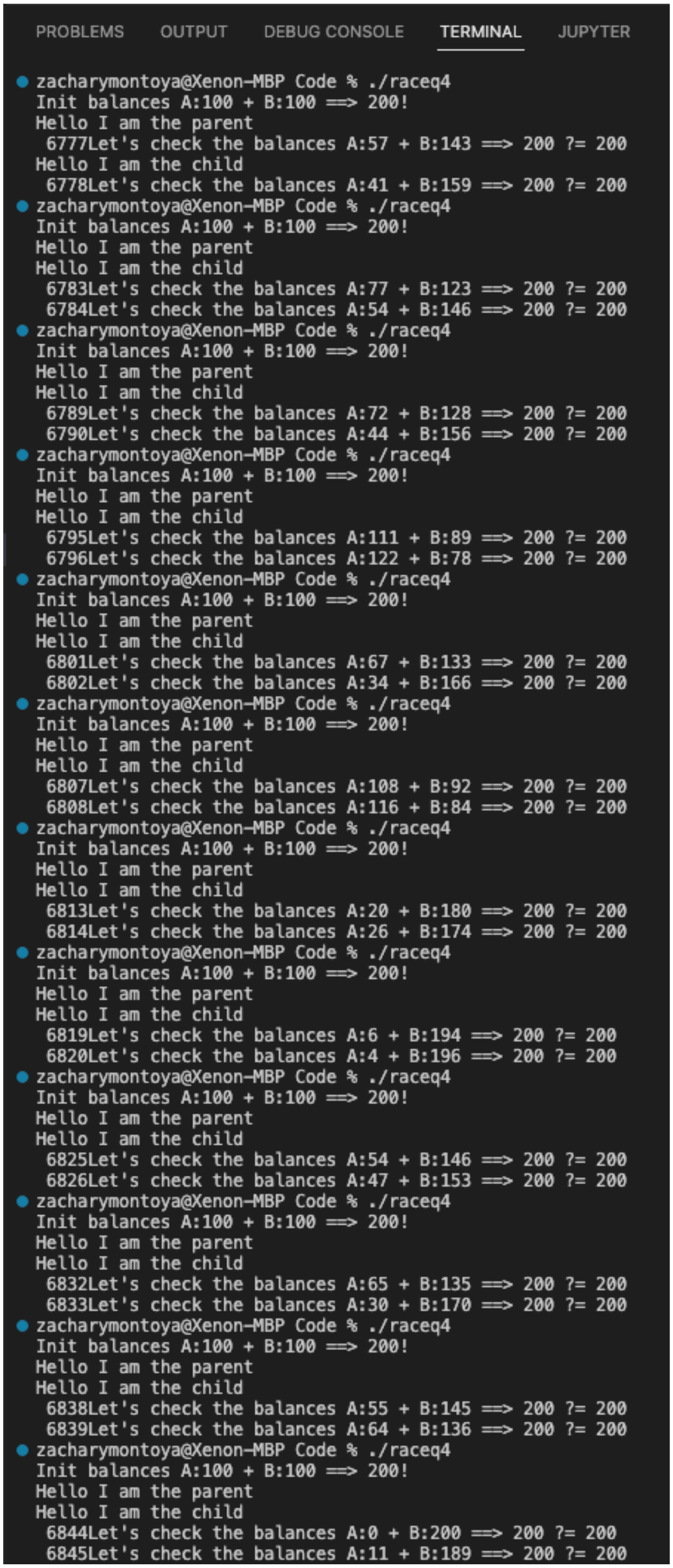
```c

int main(int argc, char **argv) {
    pid_t pid;

    /*Intializing SHM*/
    int shmID;
    key_t key = ftok("shmfile",65);
    shmID = shmget(key,27,IPC_CREAT | 0666);
    Balance = shmat(shmID,NULL,0);

    /*Setting Balances*/
    Balance[0] = 100;
    Balance[1] = 100;

    /*Mutex Intialization*/
    mutex = sem_open("mutex",O_CREAT,0666,1);
    if(mutex == SEM_FAILED){perror("\nFAILED TO OPEN SEMAPHORE FOR MUTEX\n");exit(-1);}

    /*Checking Intial Balances*/
    srand(getpid());
    printf("Init balances A:%d + B:%d ==> %d!\n",
    Balance[0],Balance[1],Balance[0]+Balance[1]);

    /*Forking*/
    pid = fork();
    if (pid == 0){ // Child
        printf("Hello I am the child\n %d", getpid());
        MakeTransactions();
    }
    else if (pid>0){ // Parent Handler
        printf("Hello I am the parent\n %d", getpid());
        MakeTransactions();
    }
    else{ printf("Error Forking. Fork returned %d",pid); exit(1);} // Fork Error Handling

    /*Checking Balances again after critical section*/
    printf("Let's check the balances A:%d + B:%d ==> %d ?= 200\n",
    Balance[0],Balance[1],Balance[0]+Balance[1]);

    sem_destroy(mutex);
    shmctl(shmID,IPC_RMID,NULL);
    return 0;
}
```

Figure 12 - Revised race.c file for Q4 2 of 2



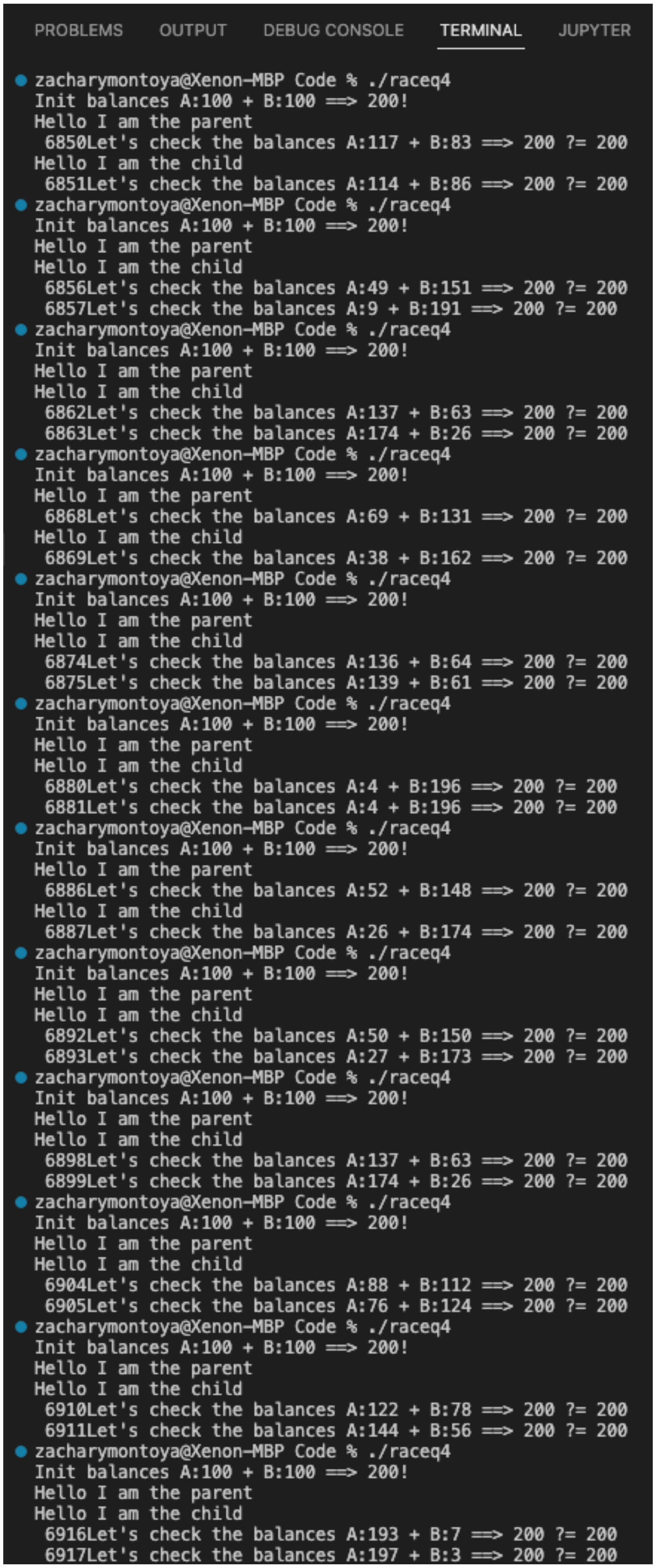
```
  zacharymontoya@Xenon-MBP Code % gcc -o raceq4 raceq4.c
  raceq4.c:89:5: warning: 'sem_destroy' is deprecated [-Wdeprecated-declarations]
      sem_destroy(mutex);
      ^
  /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/semaphore.h:53:26: note: 'sem_destroy' has been explicitly marked deprecated here
  int sem_destroy(sem_t *) __deprecated;
                           ^
  /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/cdefs.h:204:40: note: expanded from macro '__deprecated'
● zacharymontoya@Xenon-MBP Code % !g
  gcc -o raceq4 raceq4.c
  raceq4.c:78:5: warning: 'sem_destroy' is deprecated [-Wdeprecated-declarations]
      sem_destroy(mutex);
      ^
  /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/semaphore.h:53:26: note: 'sem_destroy' has been explicitly marked deprecated here
  int sem_destroy(sem_t *) __deprecated;
                           ^
  /Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/cdefs.h:204:40: note: expanded from macro '__deprecated'
  #define __deprecated    __attribute__((__deprecated__))
                                         ^
  1 warning generated.
1 ● zacharymontoya@Xenon-MBP Code % ./raceq4
  Init balances A:100 + B:100 ==> 200!
  Hello I am the parent
   6717Let's check the balances A:66 + B:134 ==> 200 ?= 200
  Hello I am the child
   6718Let's check the balances A:67 + B:133 ==> 200 ?= 200
2 ● zacharymontoya@Xenon-MBP Code % ./raceq4
  Init balances A:100 + B:100 ==> 200!
  Hello I am the parent
  Hello I am the child
   6724Let's check the balances A:125 + B:75 ==> 200 ?= 200
   6725Let's check the balances A:117 + B:83 ==> 200 ?= 200
3 ● zacharymontoya@Xenon-MBP Code % ./raceq4
  Init balances A:100 + B:100 ==> 200!
  Hello I am the parent
   6730Let's check the balances A:49 + B:151 ==> 200 ?= 200
  Hello I am the child
   6731Let's check the balances A:49 + B:151 ==> 200 ?= 200
4 ● zacharymontoya@Xenon-MBP Code % ./raceq4
  Init balances A:100 + B:100 ==> 200!
  Hello I am the parent
   6736Let's check the balances A:22 + B:178 ==> 200 ?= 200
  Hello I am the child
   6737Let's check the balances A:18 + B:182 ==> 200 ?= 200
5 ● zacharymontoya@Xenon-MBP Code % ./raceq4
  Init balances A:100 + B:100 ==> 200!
  Hello I am the parent
   6742Let's check the balances A:87 + B:113 ==> 200 ?= 200
  Hello I am the child
   6743Let's check the balances A:74 + B:126 ==> 200 ?= 200
6 ● zacharymontoya@Xenon-MBP Code % ./raceq4
  Init balances A:100 + B:100 ==> 200!
  Hello I am the parent
  Hello I am the child
   6748Let's check the balances A:61 + B:139 ==> 200 ?= 200
   6749Let's check the balances A:22 + B:178 ==> 200 ?= 200
7 ● zacharymontoya@Xenon-MBP Code % ./raceq4
  Init balances A:100 + B:100 ==> 200!
  Hello I am the parent
   6754Let's check the balances A:51 + B:149 ==> 200 ?= 200
  Hello I am the child
   6755Let's check the balances A:50 + B:150 ==> 200 ?= 200
8 ● zacharymontoya@Xenon-MBP Code % ./raceq4
  Init balances A:100 + B:100 ==> 200!
  Hello I am the parent
  Hello I am the child
   6760Let's check the balances A:148 + B:52 ==> 200 ?= 200
   6761Let's check the balances A:187 + B:13 ==> 200 ?= 200
9 ● zacharymontoya@Xenon-MBP Code % ./raceq4
  Init balances A:100 + B:100 ==> 200!
  Hello I am the parent
   6771Let's check the balances A:20 + B:180 ==> 200 ?= 200
  Hello I am the child
   6772Let's check the balances A:21 + B:179 ==> 200 ?= 200
```

Figure 13 – First 10 Outputs of the code modified for question 4



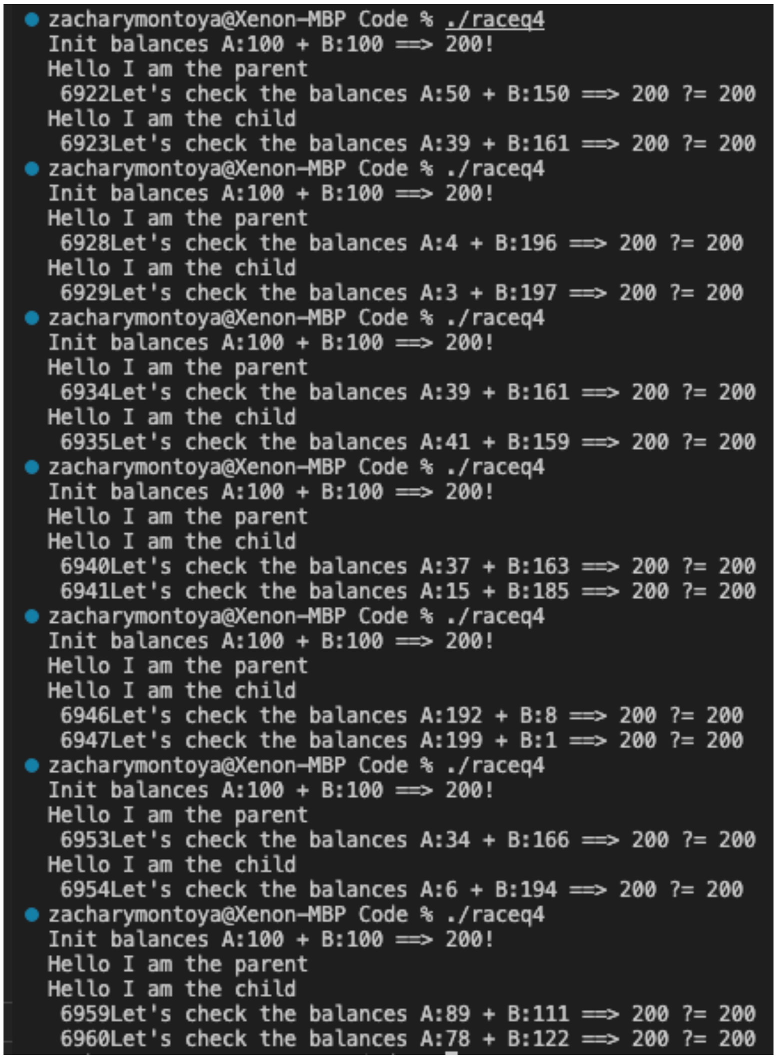
```
10 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
      6777Let's check the balances A:57 + B:143 ==> 200 ?= 200
     Hello I am the child
      6778Let's check the balances A:41 + B:159 ==> 200 ?= 200
11 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6783Let's check the balances A:77 + B:123 ==> 200 ?= 200
      6784Let's check the balances A:54 + B:146 ==> 200 ?= 200
12 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6789Let's check the balances A:72 + B:128 ==> 200 ?= 200
      6790Let's check the balances A:44 + B:156 ==> 200 ?= 200
13 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6795Let's check the balances A:111 + B:89 ==> 200 ?= 200
      6796Let's check the balances A:122 + B:78 ==> 200 ?= 200
14 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6801Let's check the balances A:67 + B:133 ==> 200 ?= 200
      6802Let's check the balances A:34 + B:166 ==> 200 ?= 200
15 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6807Let's check the balances A:108 + B:92 ==> 200 ?= 200
      6808Let's check the balances A:116 + B:84 ==> 200 ?= 200
16 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6813Let's check the balances A:20 + B:180 ==> 200 ?= 200
      6814Let's check the balances A:26 + B:174 ==> 200 ?= 200
17 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6819Let's check the balances A:6 + B:194 ==> 200 ?= 200
      6820Let's check the balances A:4 + B:196 ==> 200 ?= 200
18 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6825Let's check the balances A:54 + B:146 ==> 200 ?= 200
      6826Let's check the balances A:47 + B:153 ==> 200 ?= 200
19 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6832Let's check the balances A:65 + B:135 ==> 200 ?= 200
      6833Let's check the balances A:30 + B:170 ==> 200 ?= 200
20 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6838Let's check the balances A:55 + B:145 ==> 200 ?= 200
      6839Let's check the balances A:64 + B:136 ==> 200 ?= 200
21 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6844Let's check the balances A:0 + B:200 ==> 200 ?= 200
      6845Let's check the balances A:11 + B:189 ==> 200 ?= 200
```

Figure 14 – Next 12 Outputs of the code modified for question 4



```
22 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
      6850Let's check the balances A:117 + B:83 ==> 200 ?= 200
     Hello I am the child
      6851Let's check the balances A:114 + B:86 ==> 200 ?= 200
23 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6856Let's check the balances A:49 + B:151 ==> 200 ?= 200
      6857Let's check the balances A:9 + B:191 ==> 200 ?= 200
24 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6862Let's check the balances A:137 + B:63 ==> 200 ?= 200
      6863Let's check the balances A:174 + B:26 ==> 200 ?= 200
25 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
      6868Let's check the balances A:69 + B:131 ==> 200 ?= 200
     Hello I am the child
      6869Let's check the balances A:38 + B:162 ==> 200 ?= 200
26 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6874Let's check the balances A:136 + B:64 ==> 200 ?= 200
      6875Let's check the balances A:139 + B:61 ==> 200 ?= 200
27 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6880Let's check the balances A:4 + B:196 ==> 200 ?= 200
      6881Let's check the balances A:4 + B:196 ==> 200 ?= 200
28 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
      6886Let's check the balances A:52 + B:148 ==> 200 ?= 200
     Hello I am the child
      6887Let's check the balances A:26 + B:174 ==> 200 ?= 200
29 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6892Let's check the balances A:50 + B:150 ==> 200 ?= 200
      6893Let's check the balances A:27 + B:173 ==> 200 ?= 200
30 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6898Let's check the balances A:137 + B:63 ==> 200 ?= 200
      6899Let's check the balances A:174 + B:26 ==> 200 ?= 200
31 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6904Let's check the balances A:88 + B:112 ==> 200 ?= 200
      6905Let's check the balances A:76 + B:124 ==> 200 ?= 200
32 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6910Let's check the balances A:122 + B:78 ==> 200 ?= 200
      6911Let's check the balances A:144 + B:56 ==> 200 ?= 200
33 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6916Let's check the balances A:193 + B:7 ==> 200 ?= 200
      6917Let's check the balances A:197 + B:3 ==> 200 ?= 200
```

Figure 15 – Next 12 Outputs of the code modified for question 4



```
34 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
      6922Let's check the balances A:50 + B:150 ==> 200 ?= 200
     Hello I am the child
      6923Let's check the balances A:39 + B:161 ==> 200 ?= 200
35 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
      6928Let's check the balances A:4 + B:196 ==> 200 ?= 200
     Hello I am the child
      6929Let's check the balances A:3 + B:197 ==> 200 ?= 200
36 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
      6934Let's check the balances A:39 + B:161 ==> 200 ?= 200
     Hello I am the child
      6935Let's check the balances A:41 + B:159 ==> 200 ?= 200
   ● zacharymontoya@Xenon-MBP Code % ./raceq4
37   Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6940Let's check the balances A:37 + B:163 ==> 200 ?= 200
      6941Let's check the balances A:15 + B:185 ==> 200 ?= 200
   ● zacharymontoya@Xenon-MBP Code % ./raceq4
38   Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6946Let's check the balances A:192 + B:8 ==> 200 ?= 200
      6947Let's check the balances A:199 + B:1 ==> 200 ?= 200
39 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
      6953Let's check the balances A:34 + B:166 ==> 200 ?= 200
     Hello I am the child
      6954Let's check the balances A:6 + B:194 ==> 200 ?= 200
40 ● zacharymontoya@Xenon-MBP Code % ./raceq4
     Init balances A:100 + B:100 ==> 200!
     Hello I am the parent
     Hello I am the child
      6959Let's check the balances A:89 + B:111 ==> 200 ?= 200
      6960Let's check the balances A:78 + B:122 ==> 200 ?= 200
```

Figure 16 – Last 7 Outputs of the code modified for question 4