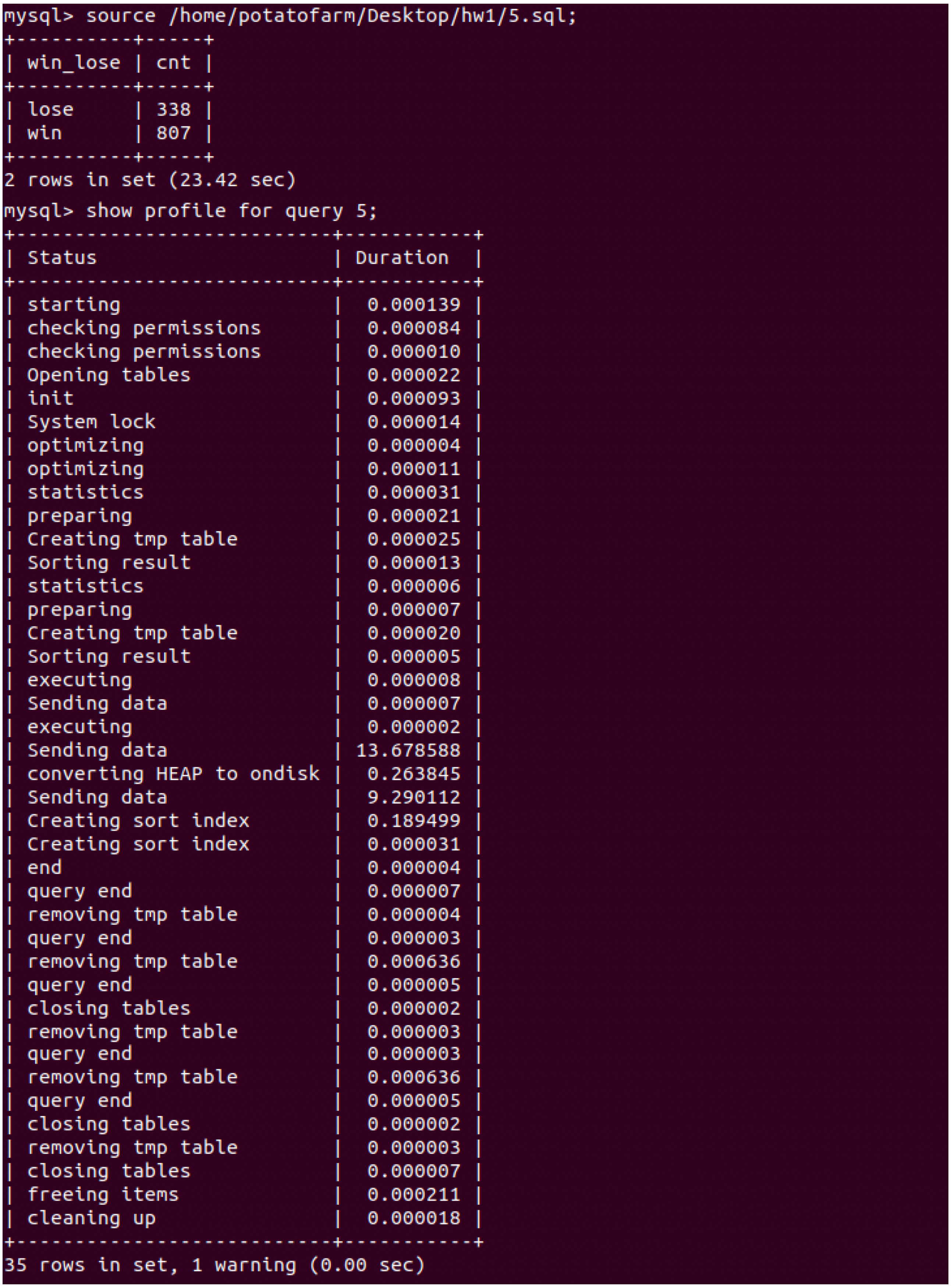
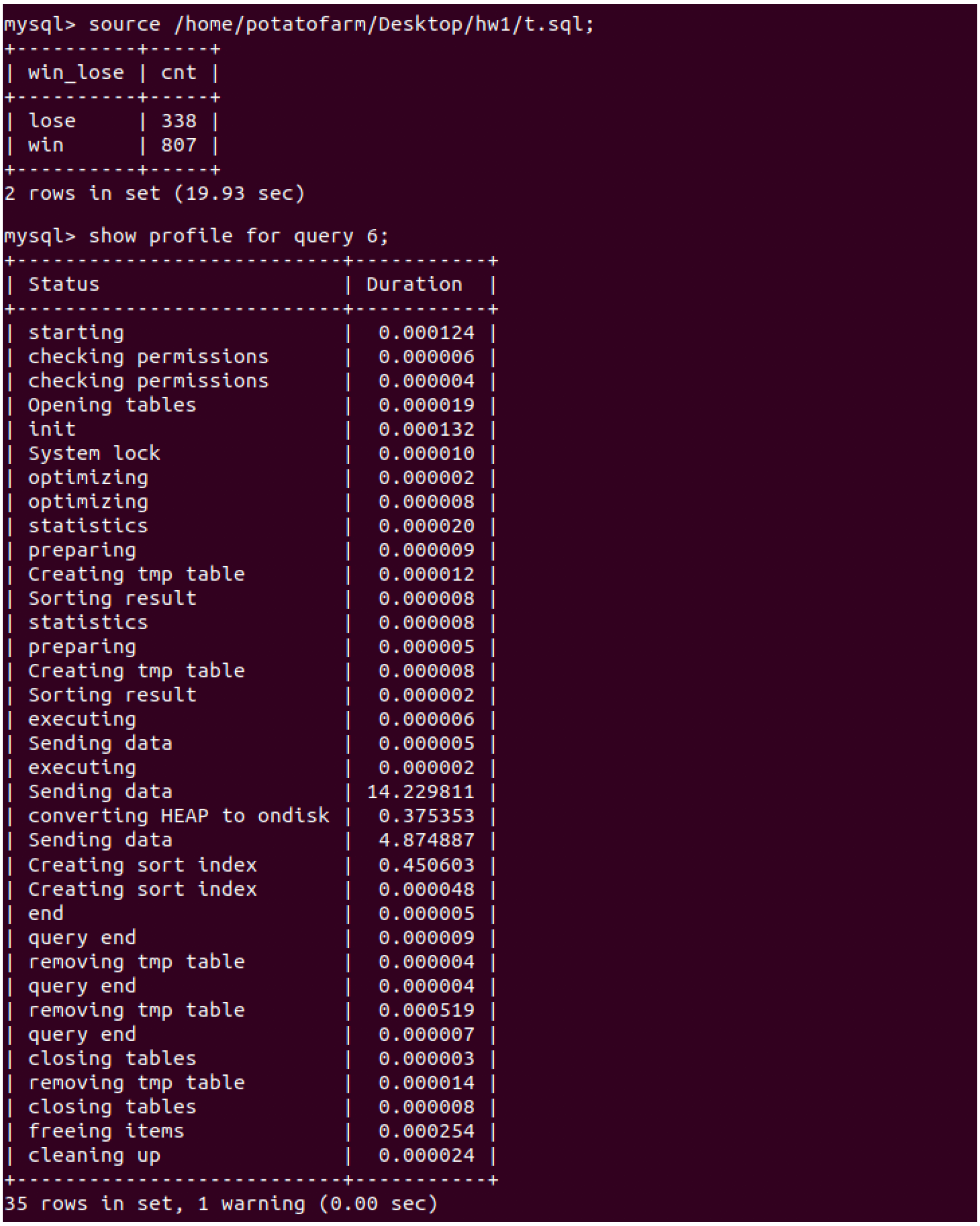
2-1-1

my query:



```
mysql> source /home/potatofarm/Desktop/hw1/5.sql;
+----------+-----+
| win_lose | cnt |
+----------+-----+
| lose     | 338 |
| win      | 807 |
+----------+-----+
2 rows in set (23.42 sec)

mysql> show profile for query 5;
+----------------------------+-----------+
| Status                     | Duration  |
+----------------------------+-----------+
| starting                   |  0.000139 |
| checking permissions       |  0.000084 |
| checking permissions       |  0.000010 |
| Opening tables             |  0.000022 |
| init                       |  0.000093 |
| System lock                |  0.000014 |
| optimizing                 |  0.000004 |
| optimizing                 |  0.000011 |
| statistics                 |  0.000031 |
| preparing                  |  0.000021 |
| Creating tmp table         |  0.000025 |
| Sorting result             |  0.000013 |
| statistics                 |  0.000006 |
| preparing                  |  0.000007 |
| Creating tmp table         |  0.000020 |
| Sorting result             |  0.000005 |
| executing                  |  0.000008 |
| Sending data               |  0.000007 |
| executing                  |  0.000002 |
| Sending data               | 13.678588 |
| converting HEAP to ondisk  |  0.263845 |
| Sending data               |  9.290112 |
| Creating sort index        |  0.189499 |
| Creating sort index        |  0.000031 |
| end                        |  0.000004 |
| query end                  |  0.000007 |
| removing tmp table         |  0.000004 |
| query end                  |  0.000003 |
| removing tmp table         |  0.000636 |
| query end                  |  0.000005 |
| closing tables             |  0.000002 |
| removing tmp table         |  0.000003 |
| query end                  |  0.000003 |
| removing tmp table         |  0.000636 |
| query end                  |  0.000005 |
| closing tables             |  0.000002 |
| removing tmp table         |  0.000003 |
| closing tables             |  0.000007 |
| freeing items              |  0.000211 |
| cleaning up                |  0.000018 |
+----------------------------+-----------+
35 rows in set, 1 warning (0.00 sec)
```

TA’s query:



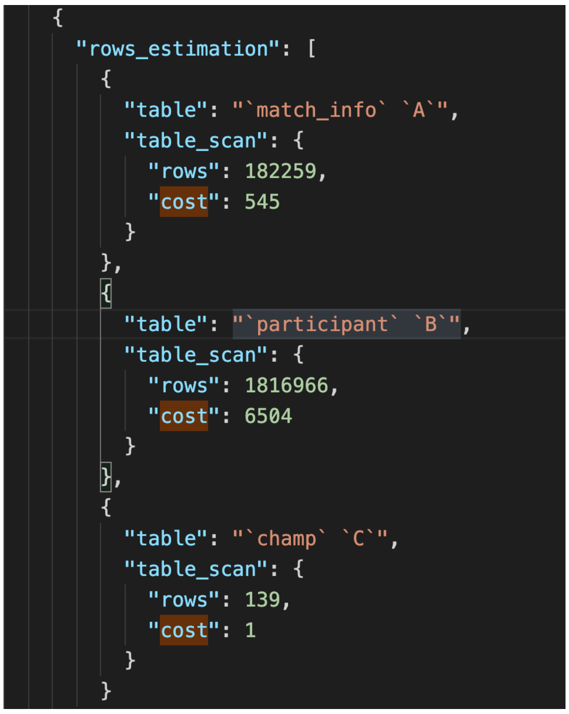
```
mysql> source /home/potatofarm/Desktop/hw1/t.sql;
+----------+-----+
| win_lose | cnt |
+----------+-----+
| lose     | 338 |
| win      | 807 |
+----------+-----+
2 rows in set (19.93 sec)

mysql> show profile for query 6;
+--------------------------+-----------+
| Status                   | Duration  |
+--------------------------+-----------+
| starting                 |  0.000124 |
| checking permissions     |  0.000006 |
| checking permissions     |  0.000004 |
| Opening tables           |  0.000019 |
| init                     |  0.000132 |
| System lock              |  0.000010 |
| optimizing               |  0.000002 |
| optimizing               |  0.000008 |
| statistics               |  0.000020 |
| preparing                |  0.000009 |
| Creating tmp table       |  0.000012 |
| Sorting result           |  0.000008 |
| statistics               |  0.000008 |
| preparing                |  0.000005 |
| Creating tmp table       |  0.000008 |
| Sorting result           |  0.000002 |
| executing                |  0.000006 |
| Sending data             |  0.000005 |
| executing                |  0.000002 |
| Sending data             | 14.229811 |
| converting HEAP to ondisk |  0.375353 |
| Sending data             |  4.874887 |
| Creating sort index      |  0.450603 |
| Creating sort index      |  0.000048 |
| end                      |  0.000005 |
| query end                |  0.000009 |
| removing tmp table       |  0.000004 |
| query end                |  0.000004 |
| removing tmp table       |  0.000519 |
| query end                |  0.000007 |
| closing tables           |  0.000003 |
| removing tmp table       |  0.000014 |
| closing tables           |  0.000008 |
| freeing items            |  0.000254 |
| cleaning up              |  0.000024 |
+--------------------------+-----------+
35 rows in set, 1 warning (0.00 sec)
```

結論：我是以show profile來檢查我和助教的query花費最多時間的是哪個部分，整體而言，助教的query比我的快了大約3.5秒，而最明顯差距的部分是sending data，我的是13.7+9.3=23，助教的則是

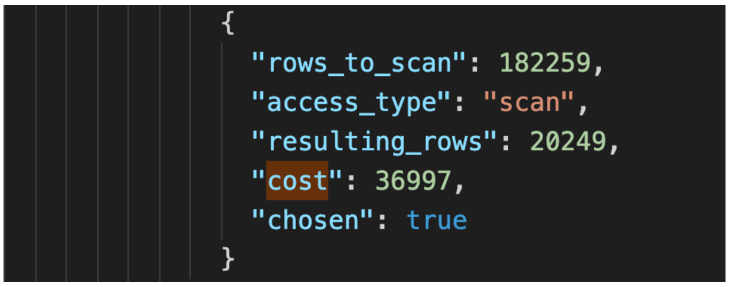
14.2+4.9=19.1，上網查後發現，sending data不只如字面上傳遞資料，還包括收集資料，因此可能我的query在收集傳遞資料上花了太多時間，仔細比對兩份code，我的code查詢的時候只有一層，導致來回取資料的成本比較大，可能是原因之一（參考網路資料，discription拆分成多個表可以減少sending data時間）

2-1-2.

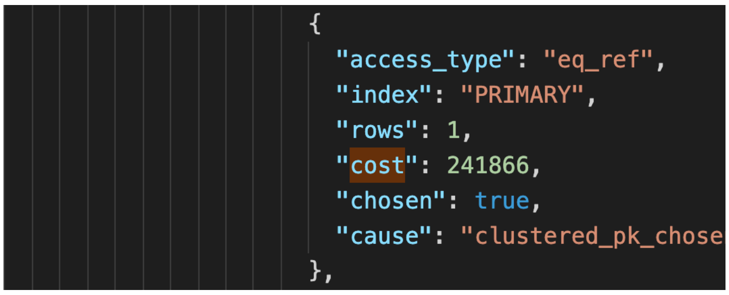
my query:



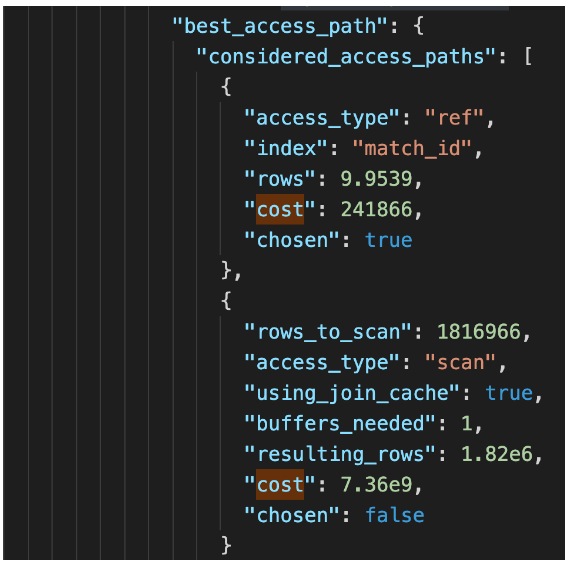
```
{
  "rows_estimation": [
    {
      "table": "`match_info` `A`",
      "table_scan": {
        "rows": 182259,
        "cost": 545
      }
    },
    {
      "table": "`participant` `B`",
      "table_scan": {
        "rows": 1816966,
        "cost": 6504
      }
    },
    {
      "table": "`champ` `C`",
      "table_scan": {
        "rows": 139,
        "cost": 1
      }
    }
```



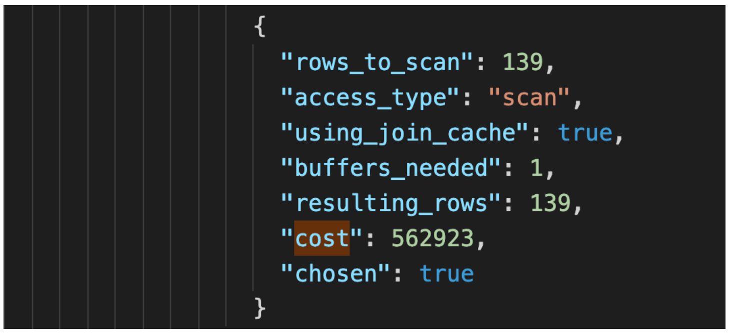
```
{
  "rows_to_scan": 182259,
  "access_type": "scan",
  "resulting_rows": 20249,
  "cost": 36997,
  "chosen": true
}
```



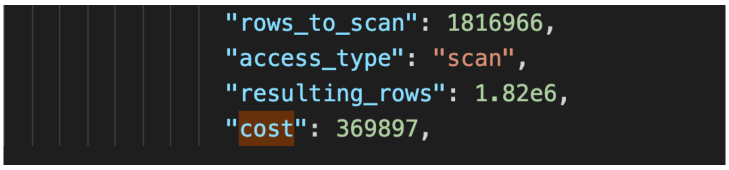
```
{
  "access_type": "eq_ref",
  "index": "PRIMARY",
  "rows": 1,
  "cost": 241866,
  "chosen": true,
  "cause": "clustered_pk_chose
},
```



```
"best_access_path": {
  "considered_access_paths": [
    {
      "access_type": "ref",
      "index": "match_id",
      "rows": 9.9539,
      "cost": 241866,
      "chosen": true
    },
    {
      "rows_to_scan": 1816966,
      "access_type": "scan",
      "using_join_cache": true,
      "buffers_needed": 1,
      "resulting_rows": 1.82e6,
      "cost": 7.36e9,
      "chosen": false
    }
```



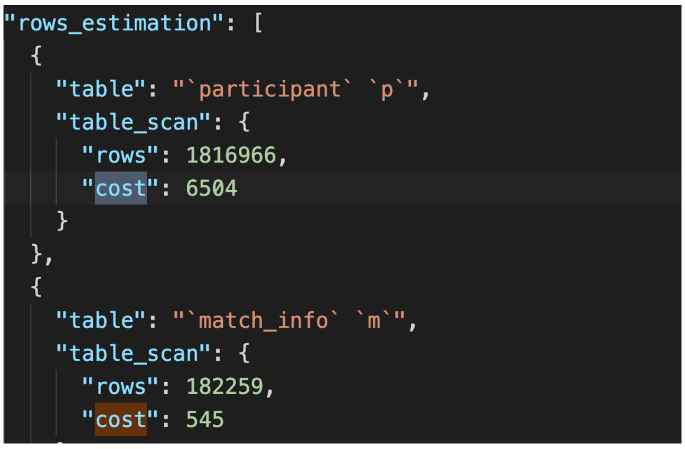
```
{
  "rows_to_scan": 139,
  "access_type": "scan",
  "using_join_cache": true,
  "buffers_needed": 1,
  "resulting_rows": 139,
  "cost": 562923,
  "chosen": true
}
```

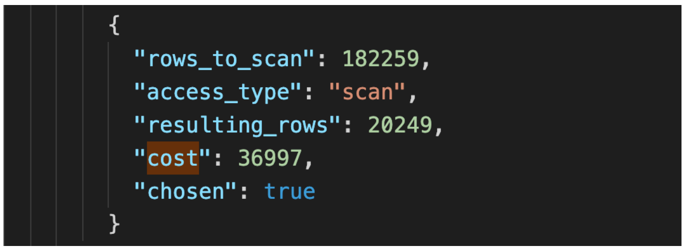


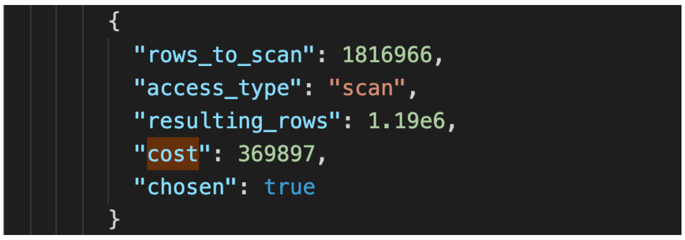
```
"rows_to_scan": 1816966,
"access_type": "scan",
"resulting_rows": 1.82e6,
"cost": 369897,
```

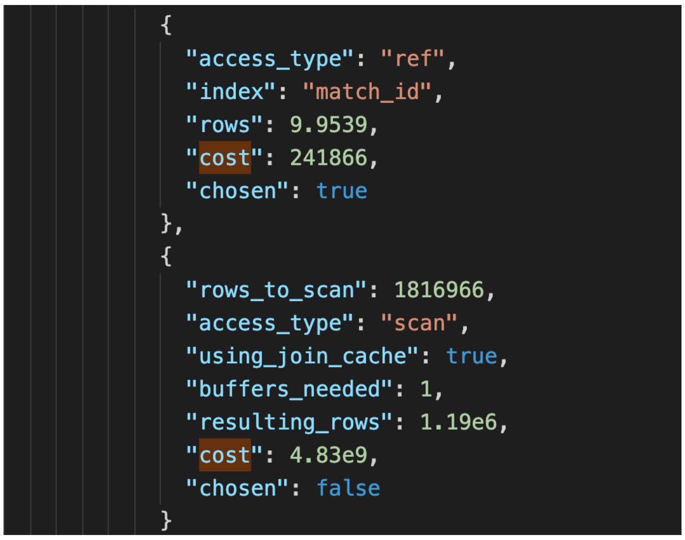
total cost = 545+6504+1+241866+36997+241866+562923+369897
= 1460599

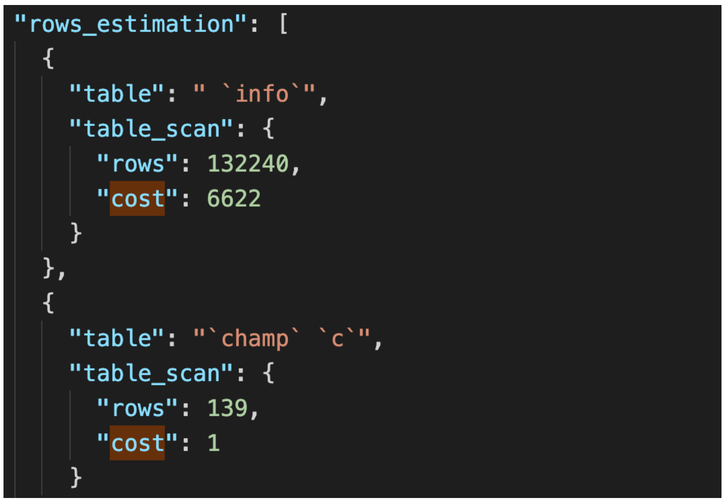
TA’s query:

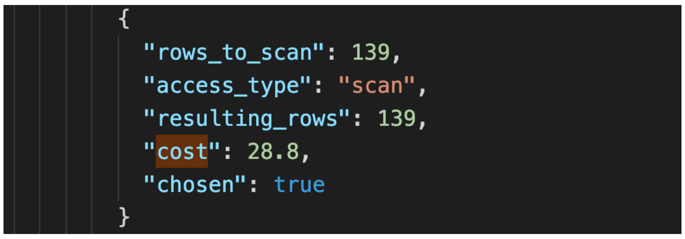

```
"rows_estimation": [
  {
    "table": "`participant` `p`",
    "table_scan": {
      "rows": 1816966,
      "cost": 6504
    }
  },
  {
    "table": "`match_info` `m`",
    "table_scan": {
      "rows": 182259,
      "cost": 545
```

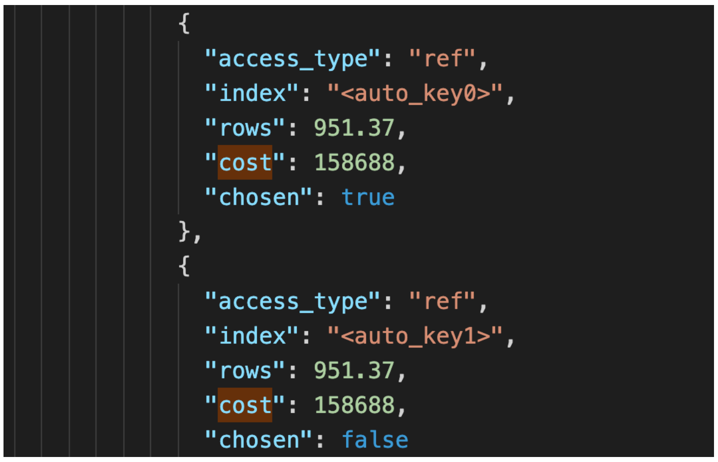

```
{
  "rows_to_scan": 182259,
  "access_type": "scan",
  "resulting_rows": 20249,
  "cost": 36997,
  "chosen": true
}
```

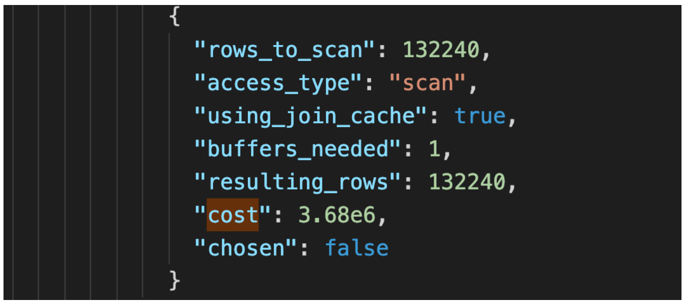

```
{
  "rows_to_scan": 1816966,
  "access_type": "scan",
  "resulting_rows": 1.19e6,
  "cost": 369897,
  "chosen": true
}
```


```
{
  "access_type": "ref",
  "index": "match_id",
  "rows": 9.9539,
  "cost": 241866,
  "chosen": true
},
{
  "rows_to_scan": 1816966,
  "access_type": "scan",
  "using_join_cache": true,
  "buffers_needed": 1,
  "resulting_rows": 1.19e6,
  "cost": 4.83e9,
  "chosen": false
}
```


```
"rows_estimation": [
  {
    "table": " `info`",
    "table_scan": {
      "rows": 132240,
      "cost": 6622
    }
  },
  {
    "table": "`champ` `c`",
    "table_scan": {
      "rows": 139,
      "cost": 1
    }
```


```
{
  "rows_to_scan": 139,
  "access_type": "scan",
  "resulting_rows": 139,
  "cost": 28.8,
  "chosen": true
}
```

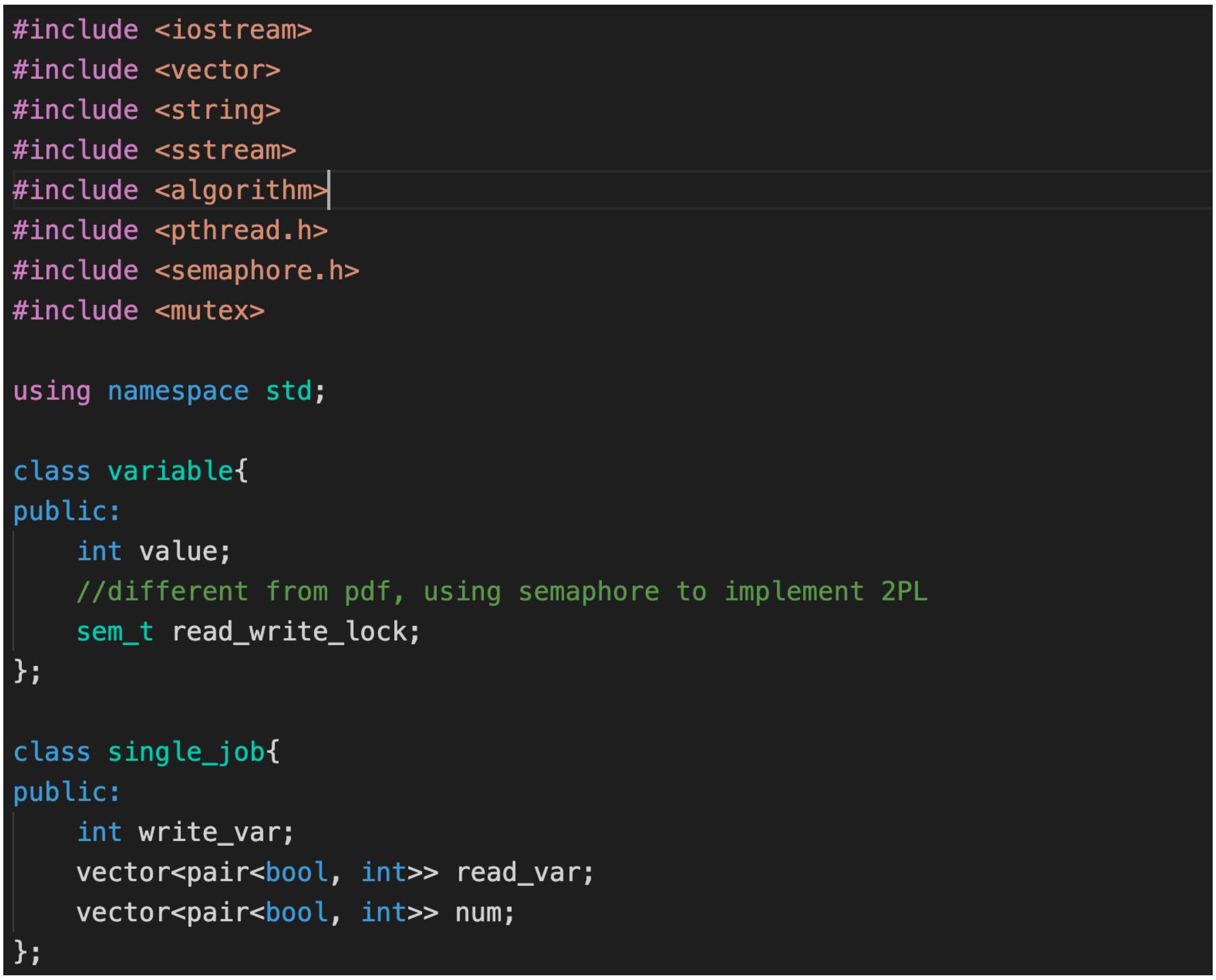

```
{
  "access_type": "ref",
  "index": "<auto_key0>",
  "rows": 951.37,
  "cost": 158688,
  "chosen": true
},
{
  "access_type": "ref",
  "index": "<auto_key1>",
  "rows": 951.37,
  "cost": 158688,
  "chosen": false
```


```
{
  "rows_to_scan": 132240,
  "access_type": "scan",
  "using_join_cache": true,
  "buffers_needed": 1,
  "resulting_rows": 132240,
  "cost": 3.68e6,
  "chosen": false
}
```

total cost =
6504+545+36997+369897+241866+6622+1+28.8+158688+15868
8 = 979836

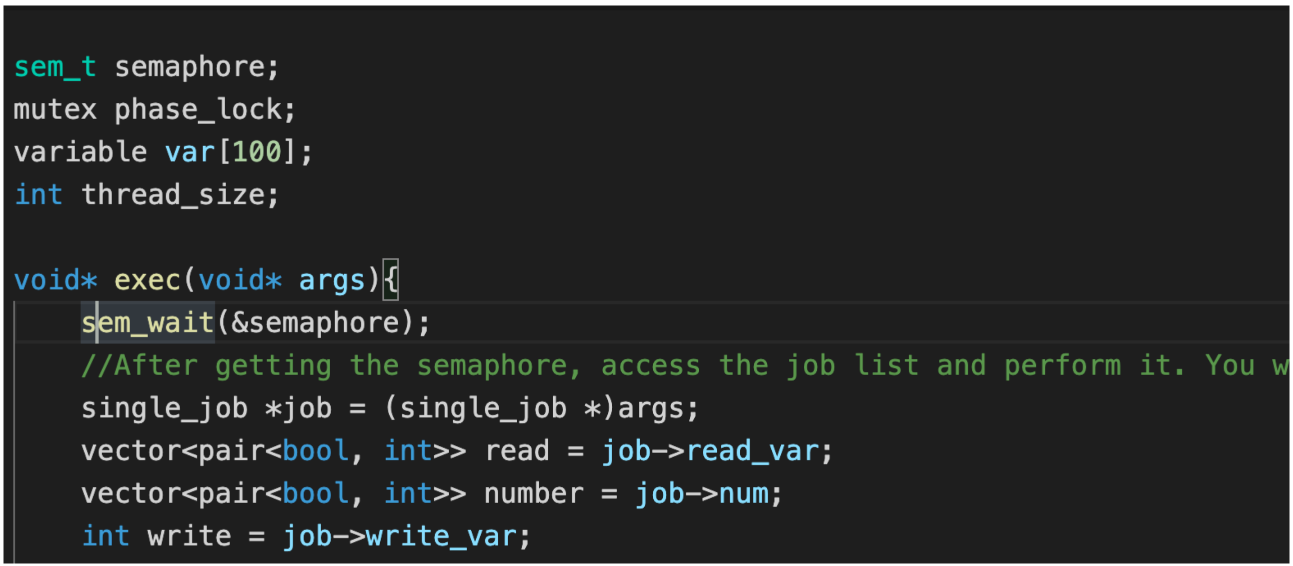
結論：以trace optimizer分析，TA的query cost相較我的query cost
少了約45000，TA的寫法更有效率

2-2-1.



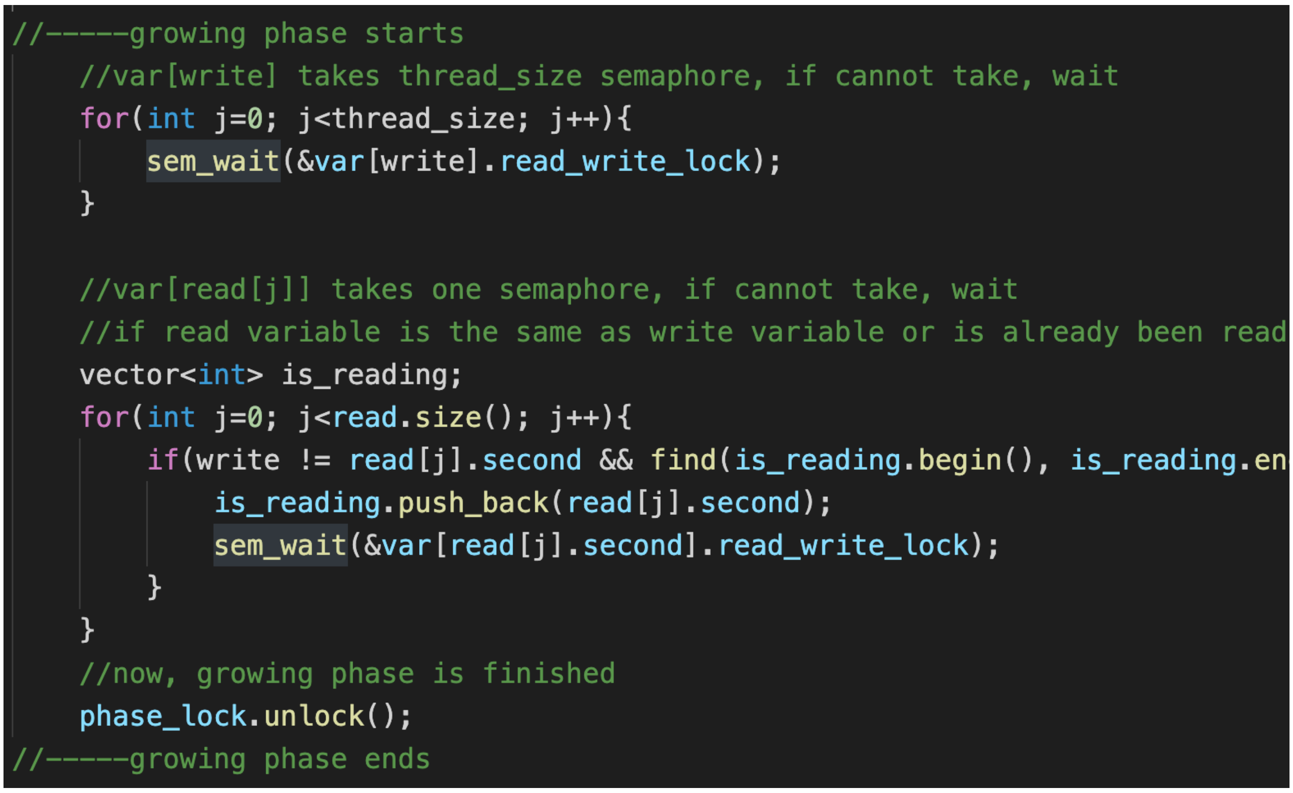
```cpp
#include <iostream>
#include <vector>
#include <string>
#include <sstream>
#include <algorithm>
#include <pthread.h>
#include <semaphore.h>
#include <mutex>

using namespace std;

class variable{
public:
    int value;
    //different from pdf, using semaphore to implement 2PL
    sem_t read_write_lock;
};

class single_job{
public:
    int write_var;
    vector<pair<bool, int>> read_var;
    vector<pair<bool, int>> num;
};
```

建立兩個class，class variable包含value和以semaphore實作的read_write_lock，原本想以mutex實作，但是因為mutex無法區分完整的shrinking phase和growing phase（lock後必須立刻unlock以確保不會干擾，而semaphore的sem_wait沒有這個問題），class single_job則包含要被寫入的variable，要被讀取的variable和純數字



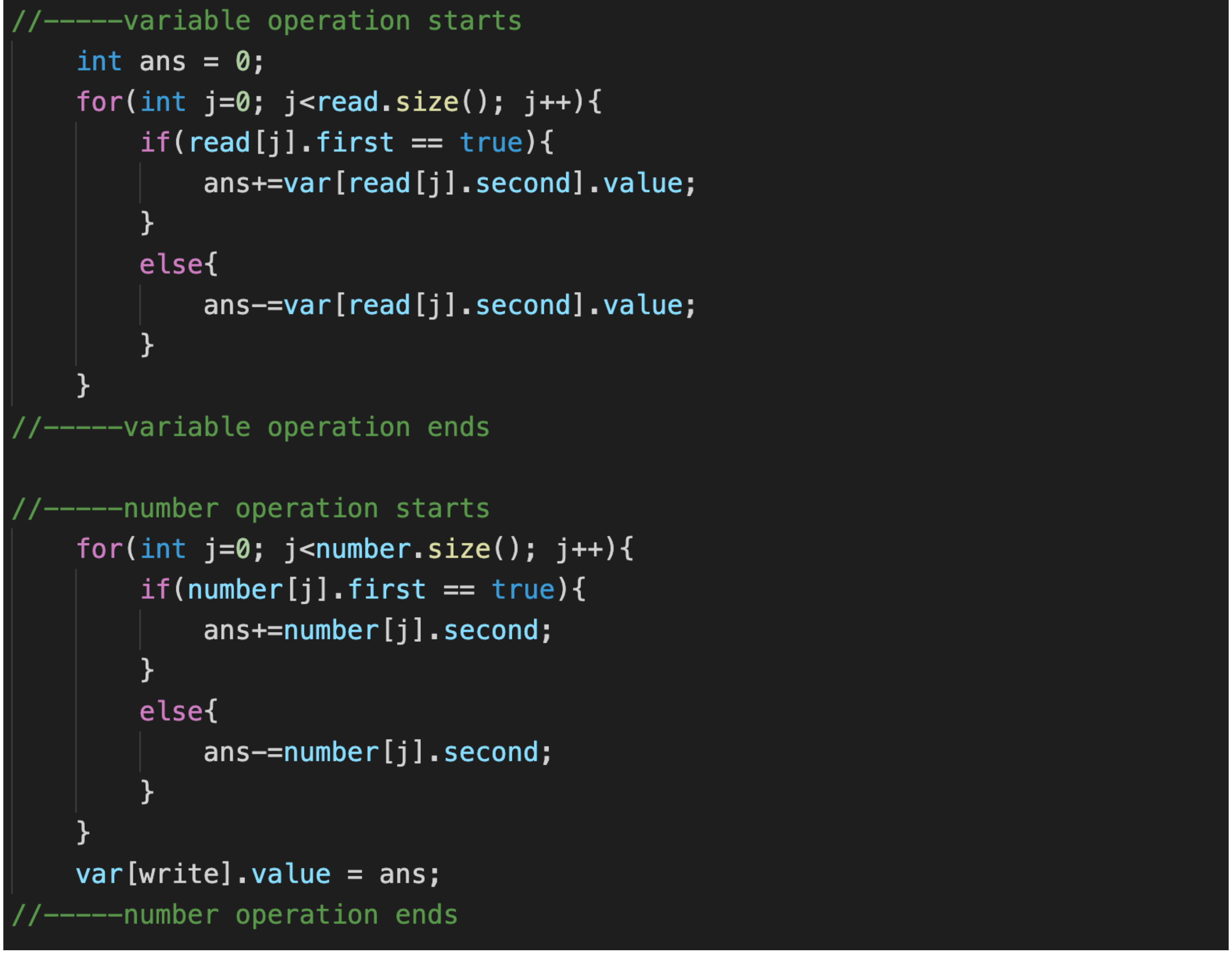
```cpp
sem_t semaphore;
mutex phase_lock;
variable var[100];
int thread_size;

void* exec(void* args){
    sem_wait(&semaphore);
    //After getting the semaphore, access the job list and perform it. You w
    single_job *job = (single_job *)args;
    vector<pair<bool, int>> read = job->read_var;
    vector<pair<bool, int>> number = job->num;
    int write = job->write_var;
```

主要執行2PL的函式是exec，在得到semaphore後，將要write和read的vector讀出來



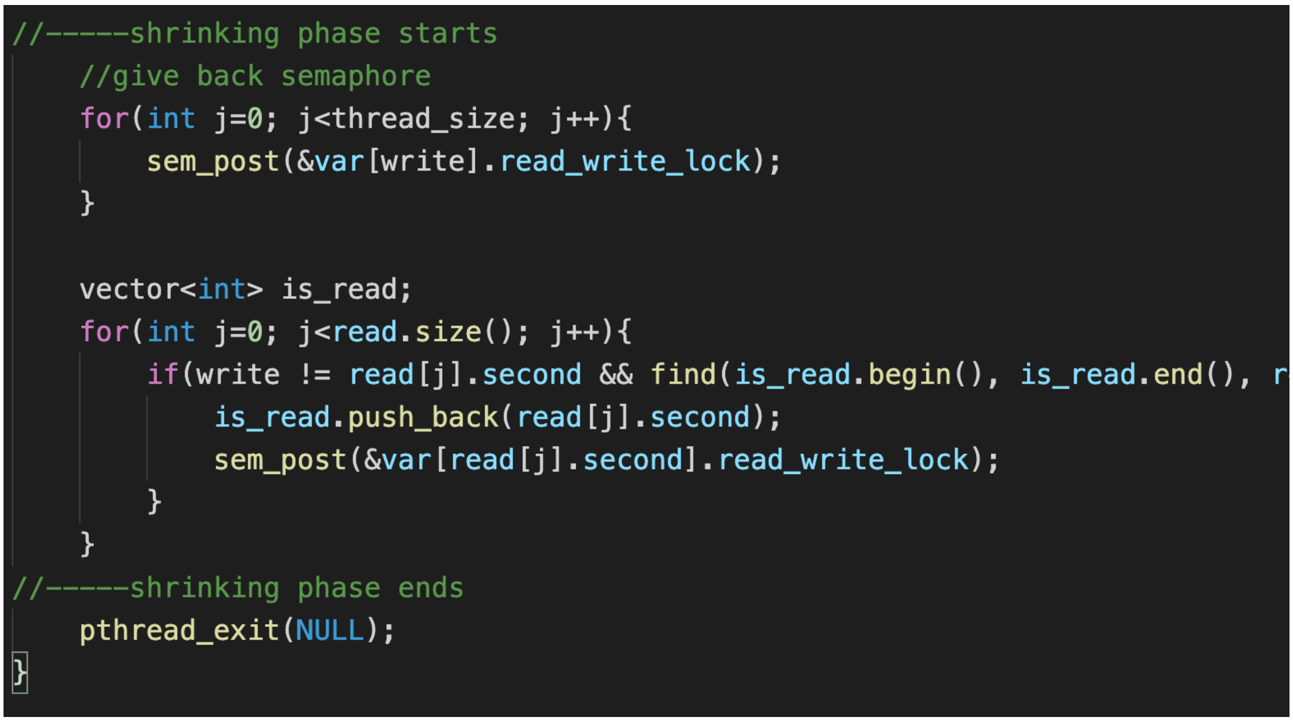
```
//-----growing phase starts
    //var[write] takes thread_size semaphore, if cannot take, wait
    for(int j=0; j<thread_size; j++){
        sem_wait(&var[write].read_write_lock);
    }

    //var[read[j]] takes one semaphore, if cannot take, wait
    //if read variable is the same as write variable or is already been read
    vector<int> is_reading;
    for(int j=0; j<read.size(); j++){
        if(write != read[j].second && find(is_reading.begin(), is_reading.en
            is_reading.push_back(read[j].second);
            sem_wait(&var[read[j].second].read_write_lock);
        }
    }
    //now, growing phase is finished
    phase_lock.unlock();
//-----growing phase ends
```

這段是growing phase，如果是要write的variable，就取thread_size的semaphore，相當於exclusive lock，如果是要read的variable，就取一個semaphore，相當於shared lock，但是重複read的variable和同時read and write的variable可以跳過，避免自己卡到自己，當growing phase結束後，就把phase_lock unlock，這樣下一組job才可以開始growing phase



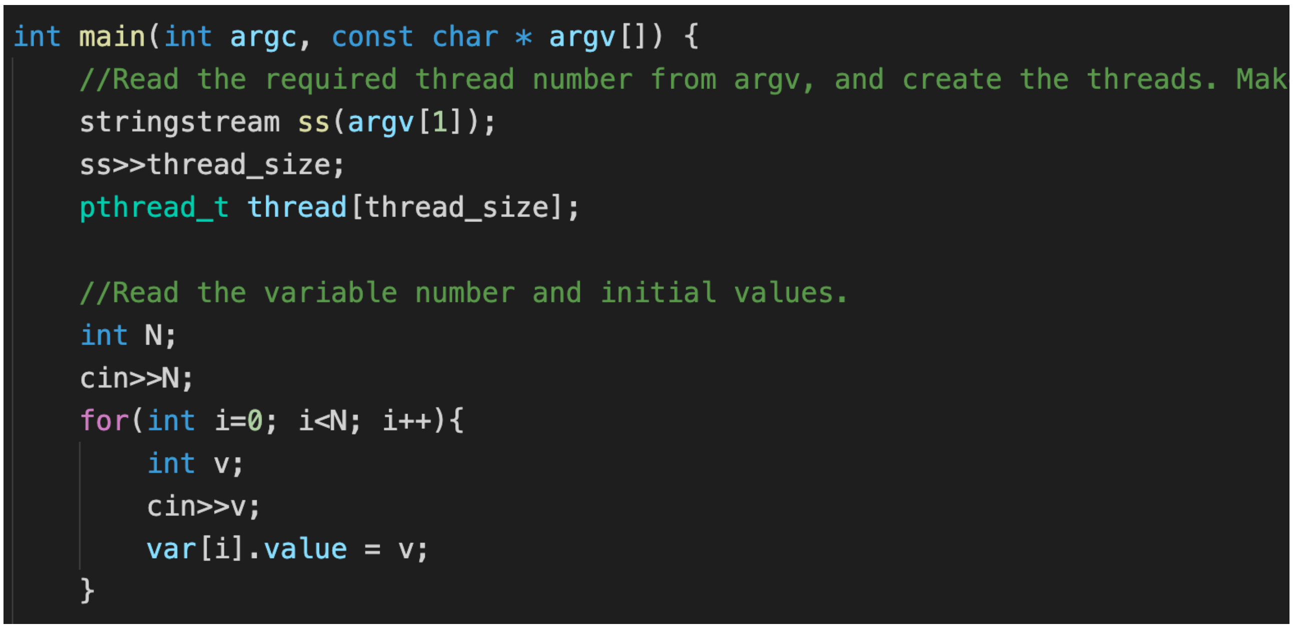
```
//------variable operation starts
    int ans = 0;
    for(int j=0; j<read.size(); j++){
        if(read[j].first == true){
            ans+=var[read[j].second].value;
        }
        else{
            ans-=var[read[j].second].value;
        }
    }
//------variable operation ends

//------number operation starts
    for(int j=0; j<number.size(); j++){
        if(number[j].first == true){
            ans+=number[j].second;
        }
        else{
            ans-=number[j].second;
        }
    }
    var[write].value = ans;
//------number operation ends
```

這段是在做command的加減



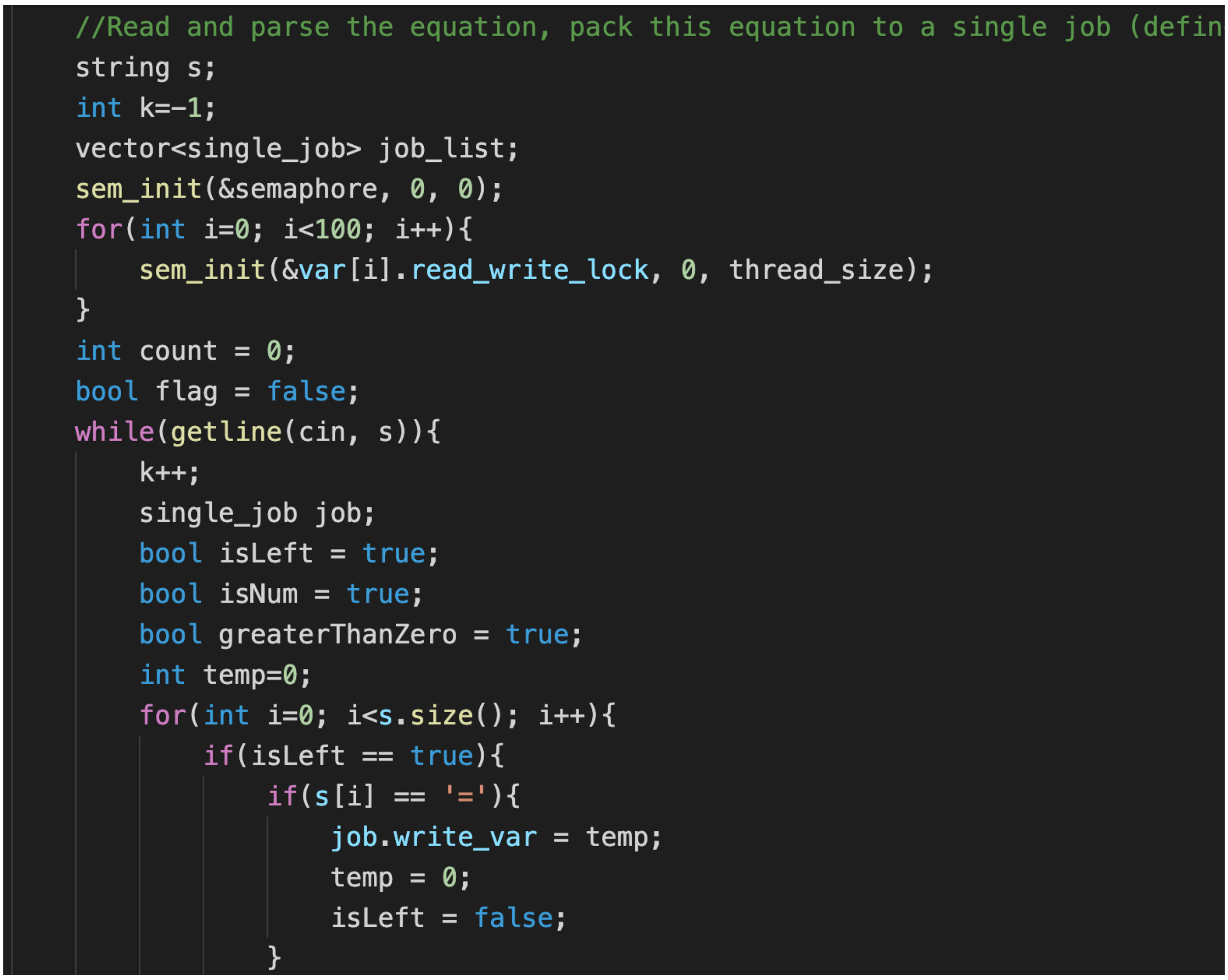
```
//------shrinking phase starts
    //give back semaphore
    for(int j=0; j<thread_size; j++){
        sem_post(&var[write].read_write_lock);
    }

    vector<int> is_read;
    for(int j=0; j<read.size(); j++){
        if(write != read[j].second && find(is_read.begin(), is_read.end(), r
            is_read.push_back(read[j].second);
            sem_post(&var[read[j].second].read_write_lock);
        }
    }
//------shrinking phase ends
    pthread_exit(NULL);
}
```

這段開始做shrinking phase，和growing phase幾乎一樣，差別是shrinking phase把semaphore還回去

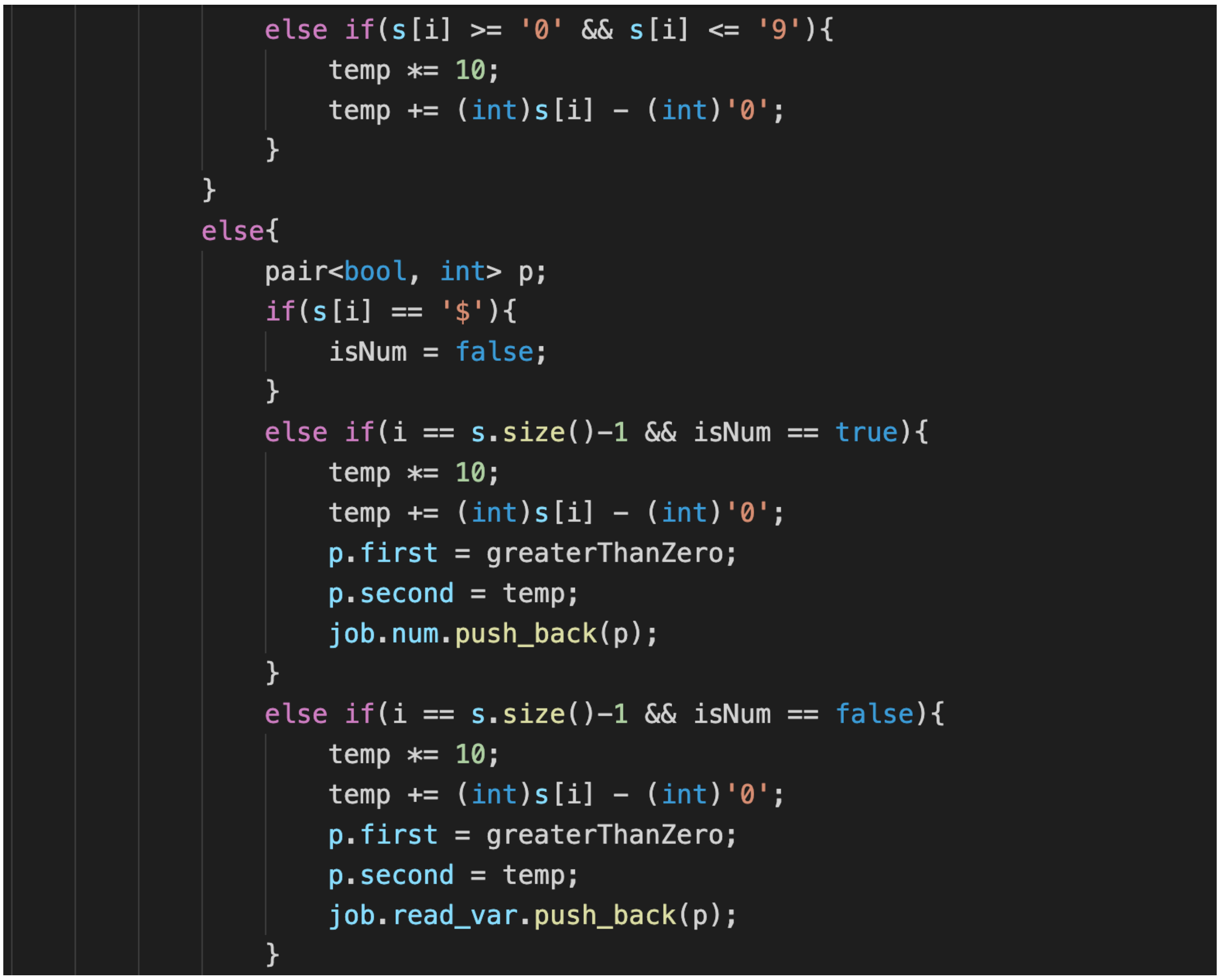


```cpp
int main(int argc, const char * argv[]) {
    //Read the required thread number from argv, and create the threads. Mak
    stringstream ss(argv[1]);
    ss>>thread_size;
    pthread_t thread[thread_size];

    //Read the variable number and initial values.
    int N;
    cin>>N;
    for(int i=0; i<N; i++){
        int v;
        cin>>v;
        var[i].value = v;
    }
```

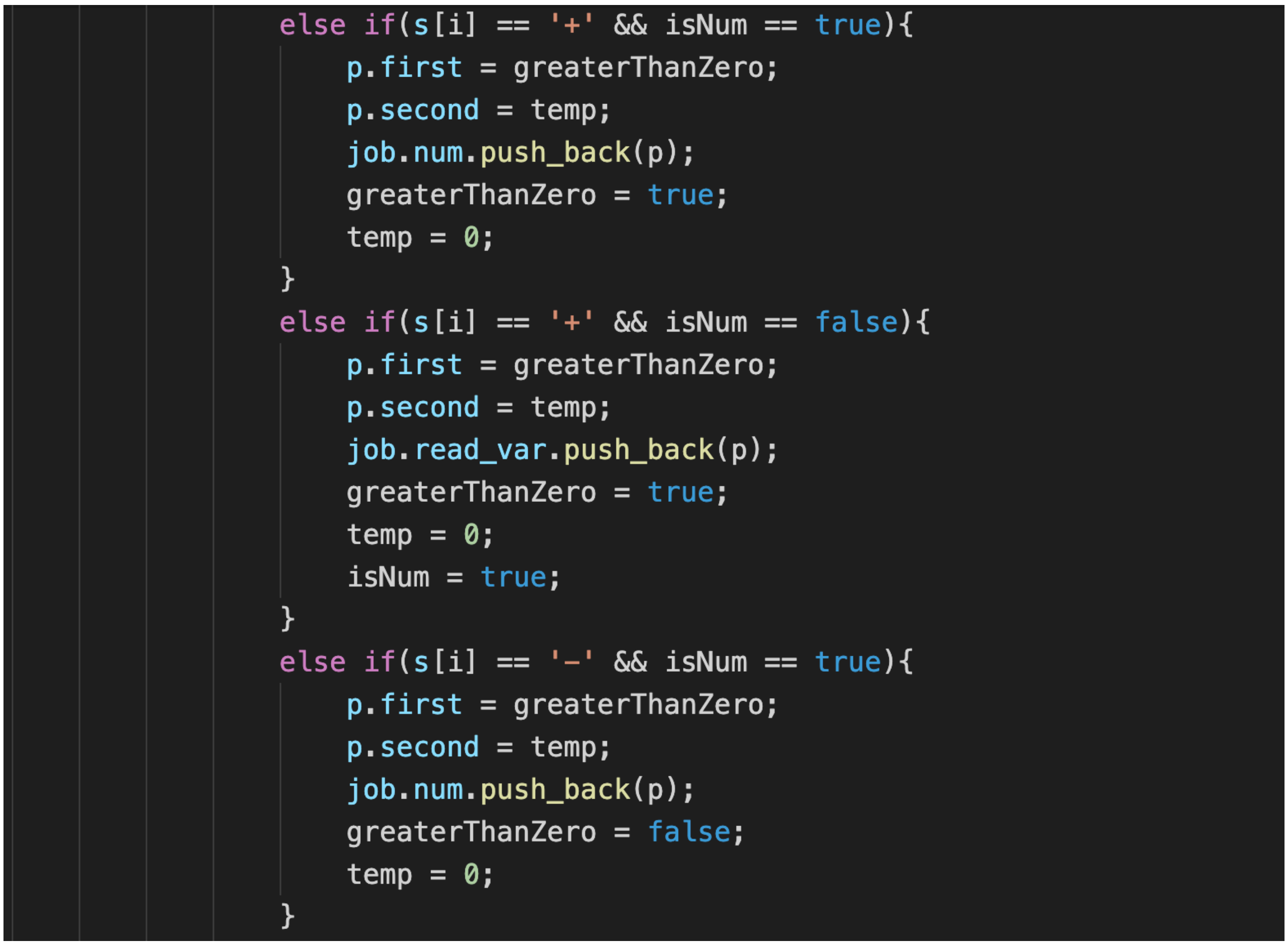
main function，依據argv[1]建立thread
然後讀N和初始值



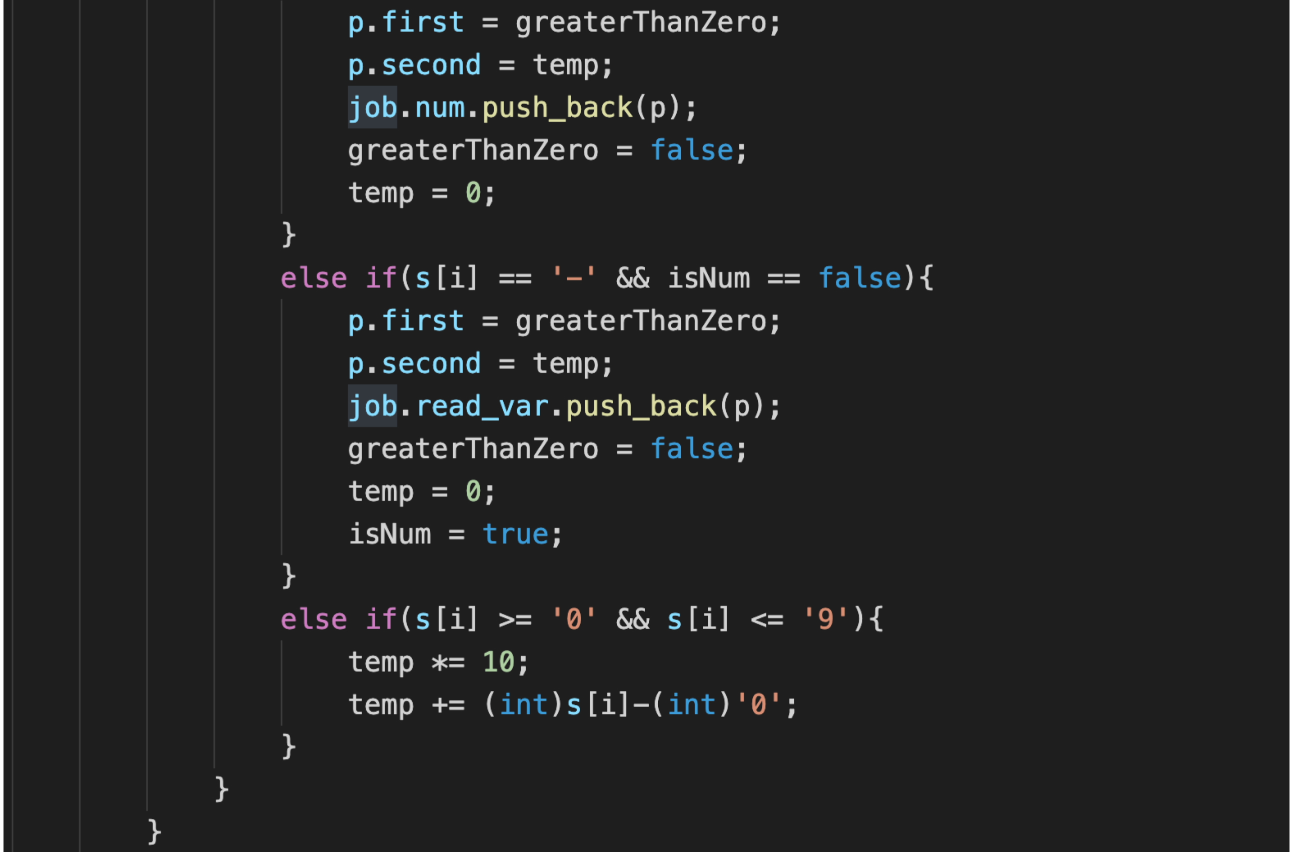
```cpp
    //Read and parse the equation, pack this equation to a single job (defin
    string s;
    int k=-1;
    vector<single_job> job_list;
    sem_init(&semaphore, 0, 0);
    for(int i=0; i<100; i++){
        sem_init(&var[i].read_write_lock, 0, thread_size);
    }
    int count = 0;
    bool flag = false;
    while(getline(cin, s)){
        k++;
        single_job job;
        bool isLeft = true;
        bool isNum = true;
        bool greaterThanZero = true;
        int temp=0;
        for(int i=0; i<s.size(); i++){
            if(isLeft == true){
                if(s[i] == '='){
                    job.write_var = temp;
                    temp = 0;
                    isLeft = false;
                }
```



```
                else if(s[i] >= '0' && s[i] <= '9'){
                    temp *= 10;
                    temp += (int)s[i] - (int)'0';
                }
            }
            else{
                pair<bool, int> p;
                if(s[i] == '$'){
                    isNum = false;
                }
                else if(i == s.size()-1 && isNum == true){
                    temp *= 10;
                    temp += (int)s[i] - (int)'0';
                    p.first = greaterThanZero;
                    p.second = temp;
                    job.num.push_back(p);
                }
                else if(i == s.size()-1 && isNum == false){
                    temp *= 10;
                    temp += (int)s[i] - (int)'0';
                    p.first = greaterThanZero;
                    p.second = temp;
                    job.read_var.push_back(p);
                }
```

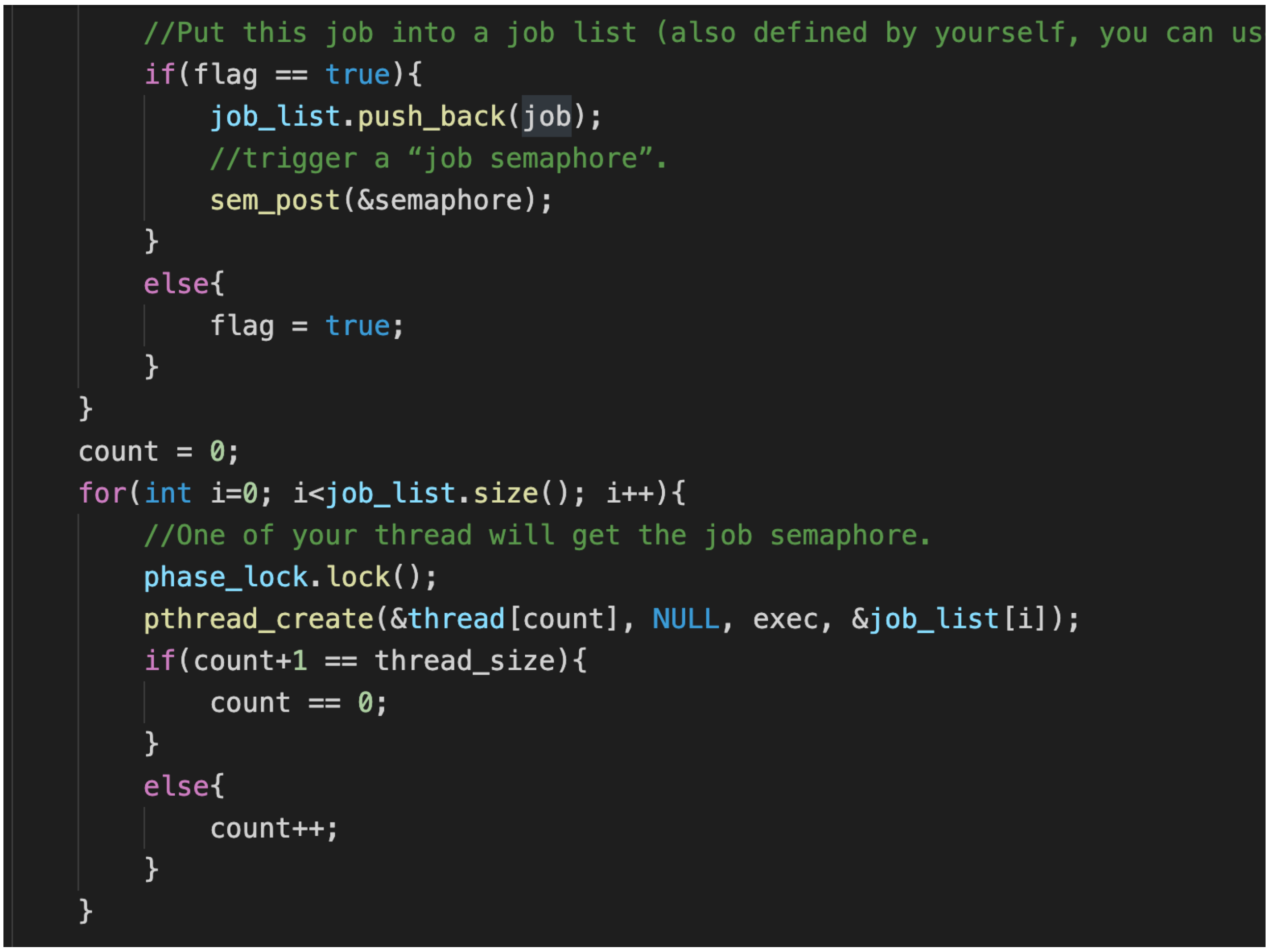


```
                else if(s[i] == '+' && isNum == true){
                    p.first = greaterThanZero;
                    p.second = temp;
                    job.num.push_back(p);
                    greaterThanZero = true;
                    temp = 0;
                }
                else if(s[i] == '+' && isNum == false){
                    p.first = greaterThanZero;
                    p.second = temp;
                    job.read_var.push_back(p);
                    greaterThanZero = true;
                    temp = 0;
                    isNum = true;
                }
                else if(s[i] == '-' && isNum == true){
                    p.first = greaterThanZero;
                    p.second = temp;
                    job.num.push_back(p);
                    greaterThanZero = false;
                    temp = 0;
                }
```



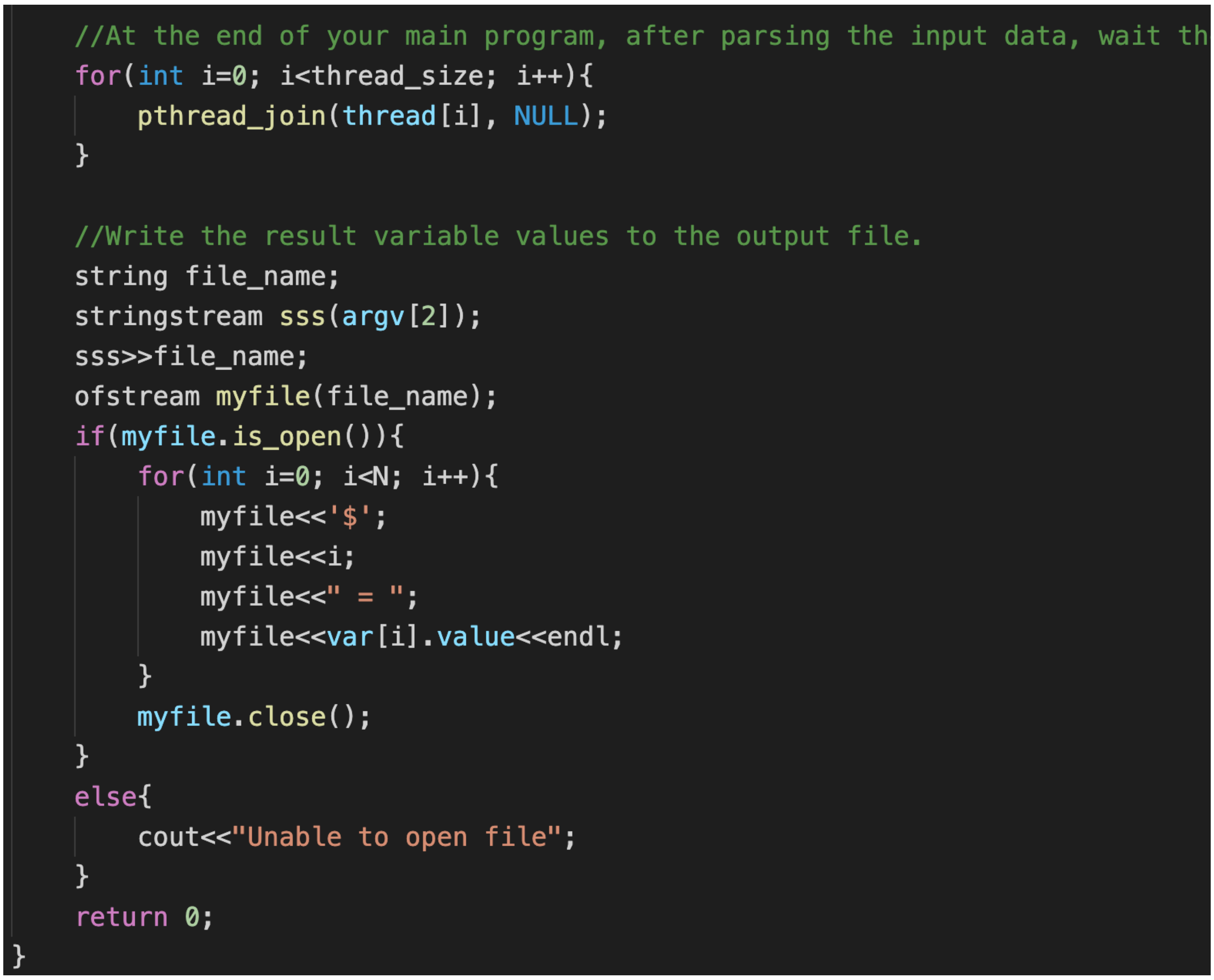
```
                    p.first = greaterThanZero;
                    p.second = temp;
                    job.num.push_back(p);
                    greaterThanZero = false;
                    temp = 0;
                }
                else if(s[i] == '-' && isNum == false){
                    p.first = greaterThanZero;
                    p.second = temp;
                    job.read_var.push_back(p);
                    greaterThanZero = false;
                    temp = 0;
                    isNum = true;
                }
                else if(s[i] >= '0' && s[i] <= '9'){
                    temp *= 10;
                    temp += (int)s[i]-(int)'0';
                }
            }
        }
```

對command做字串處理，分為write_var、read_var和num，將他們存入job後，放進job_list中



```
        //Put this job into a job list (also defined by yourself, you can us
        if(flag == true){
            job_list.push_back(job);
            //trigger a “job semaphore”.
            sem_post(&semaphore);
        }
        else{
            flag = true;
        }
    }
    count = 0;
    for(int i=0; i<job_list.size(); i++){
        //One of your thread will get the job semaphore.
        phase_lock.lock();
        pthread_create(&thread[count], NULL, exec, &job_list[i]);
        if(count+1 == thread_size){
            count == 0;
        }
        else{
            count++;
        }
    }
```

上方的判斷式是為了不讀第一行（換行字串），先將phase_lock上鎖，代表進入growing phase，接著就開始進入thread運算，盡可能讓command進入thread

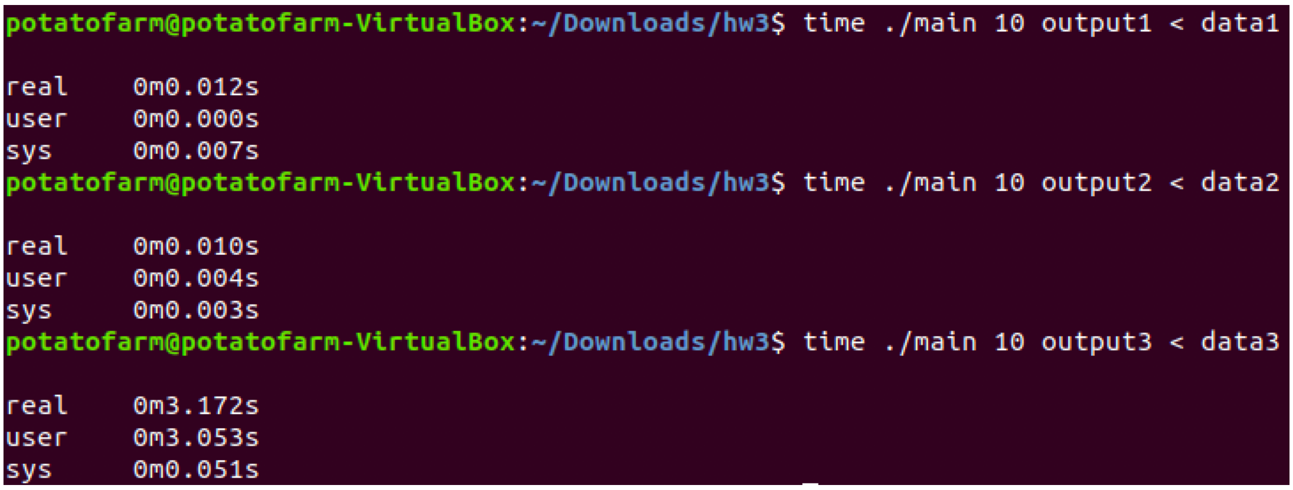


```
    //At the end of your main program, after parsing the input data, wait th
    for(int i=0; i<thread_size; i++){
        pthread_join(thread[i], NULL);
    }

    //Write the result variable values to the output file.
    string file_name;
    stringstream sss(argv[2]);
    sss>>file_name;
    ofstream myfile(file_name);
    if(myfile.is_open()){
        for(int i=0; i<N; i++){
            myfile<<'$';
            myfile<<i;
            myfile<<" = ";
            myfile<<var[i].value<<endl;
        }
        myfile.close();
    }
    else{
        cout<<"Unable to open file";
    }
    return 0;
}
```

最後判斷是否全部thread都已執行完畢，是的話就output到指定的file中

2-2-2.

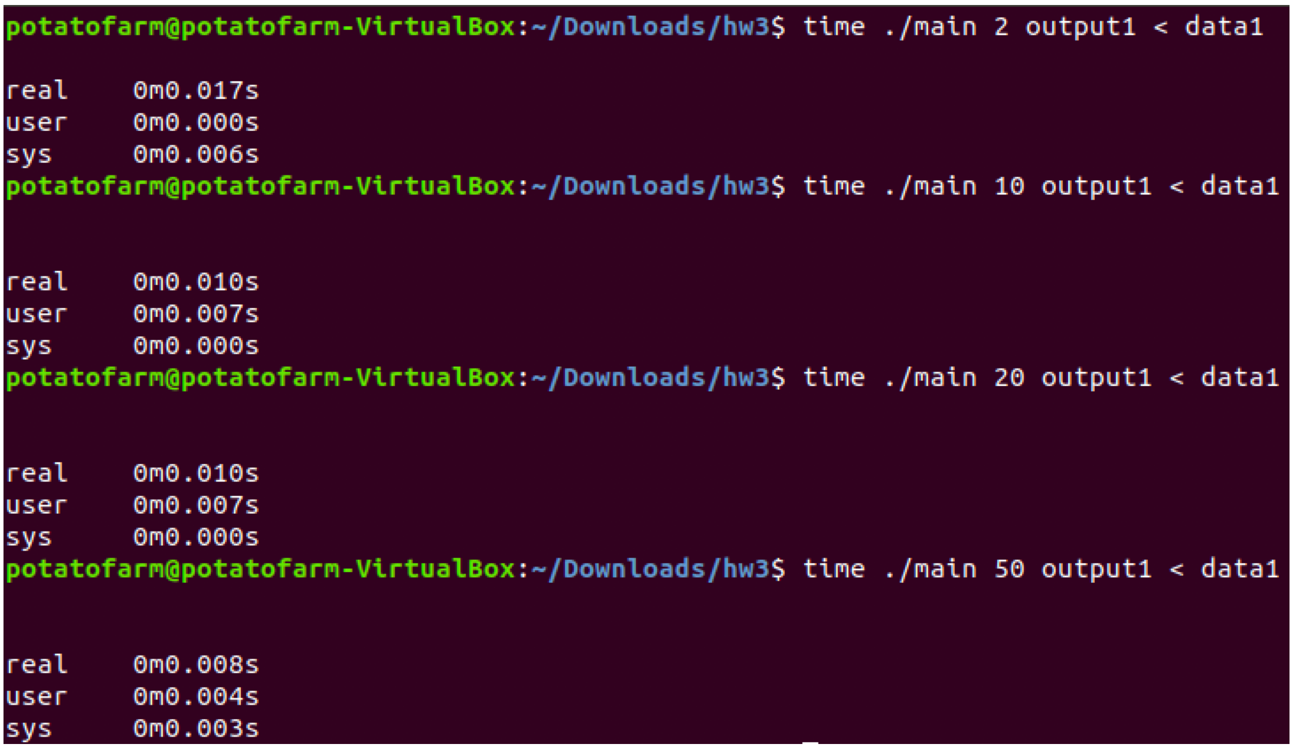
exec time in different data:



```
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ time ./main 10 output1 < data1

real	0m0.012s
user	0m0.000s
sys	0m0.007s
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ time ./main 10 output2 < data2

real	0m0.010s
user	0m0.004s
sys	0m0.003s
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ time ./main 10 output3 < data3

real	0m3.172s
user	0m3.053s
sys	0m0.051s
```

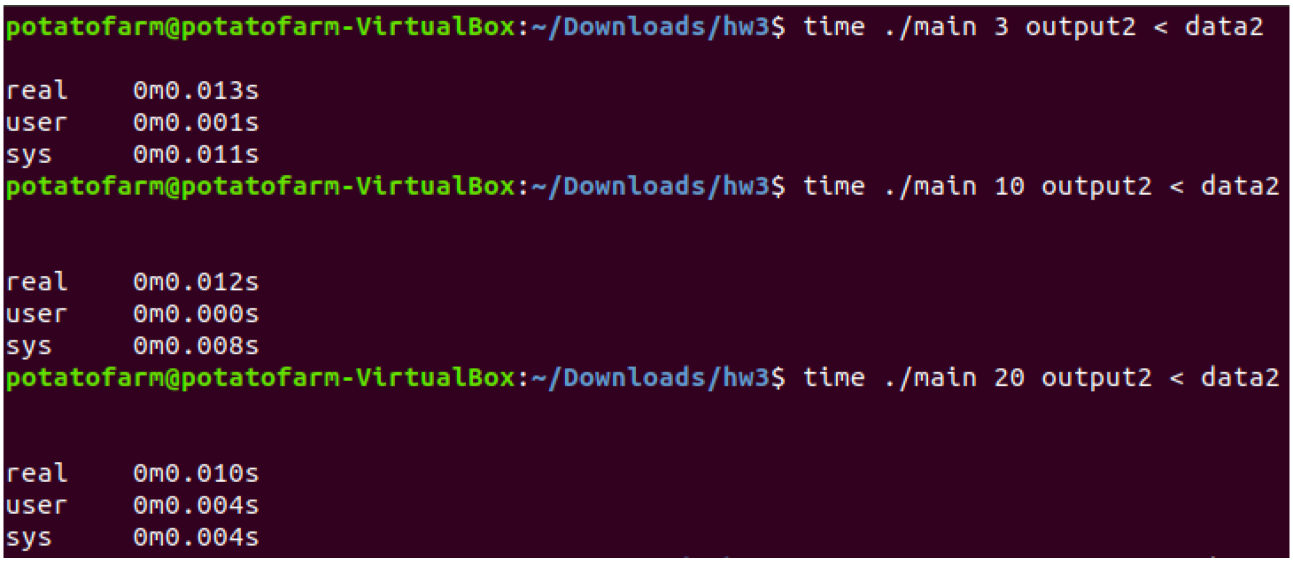
exec time in different thread number:

data1:

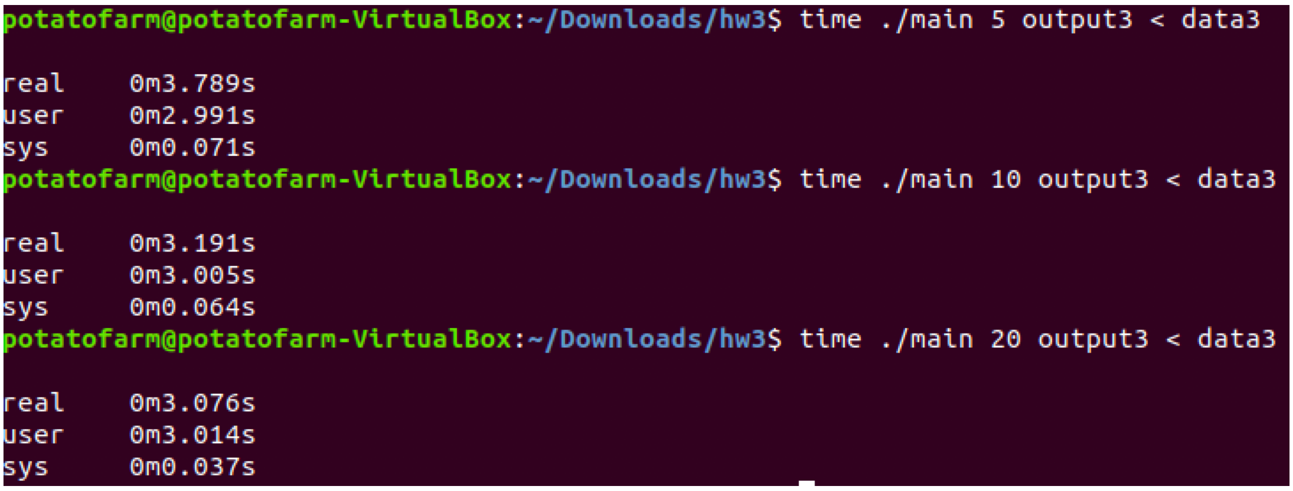


```
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ time ./main 2 output1 < data1

real	0m0.017s
user	0m0.000s
sys	0m0.006s
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ time ./main 10 output1 < data1


real	0m0.010s
user	0m0.007s
sys	0m0.000s
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ time ./main 20 output1 < data1


real	0m0.010s
user	0m0.007s
sys	0m0.000s
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ time ./main 50 output1 < data1


real	0m0.008s
user	0m0.004s
sys	0m0.003s
```

data2:

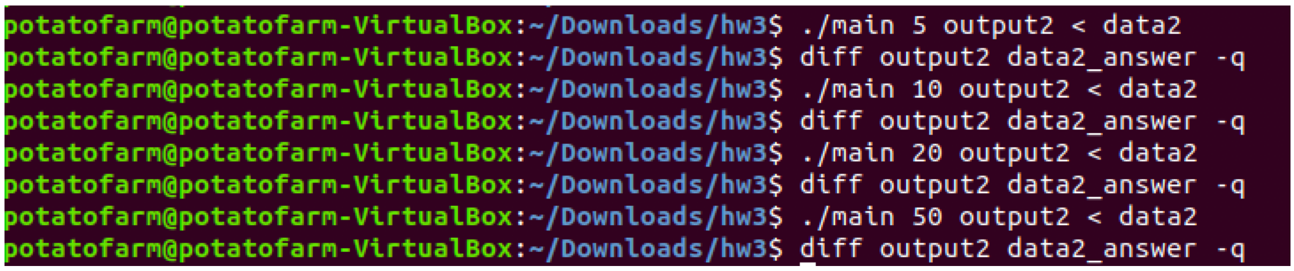

```
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ time ./main 3 output2 < data2

real	0m0.013s
user	0m0.001s
sys	0m0.011s
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ time ./main 10 output2 < data2


real	0m0.012s
user	0m0.000s
sys	0m0.008s
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ time ./main 20 output2 < data2


real	0m0.010s
user	0m0.004s
sys	0m0.004s
```

data3:


```
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ time ./main 5 output3 < data3

real	0m3.789s
user	0m2.991s
sys	0m0.071s
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ time ./main 10 output3 < data3

real	0m3.191s
user	0m3.005s
sys	0m0.064s
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ time ./main 20 output3 < data3

real	0m3.076s
user	0m3.014s
sys	0m0.037s
```

2-2-3.

it works fine



```
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ ./main 5 output2 < data2
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ diff output2 data2_answer -q
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ ./main 10 output2 < data2
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ diff output2 data2_answer -q
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ ./main 20 output2 < data2
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ diff output2 data2_answer -q
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ ./main 50 output2 < data2
potatofarm@potatofarm-VirtualBox:~/Downloads/hw3$ diff output2 data2_answer -q
```