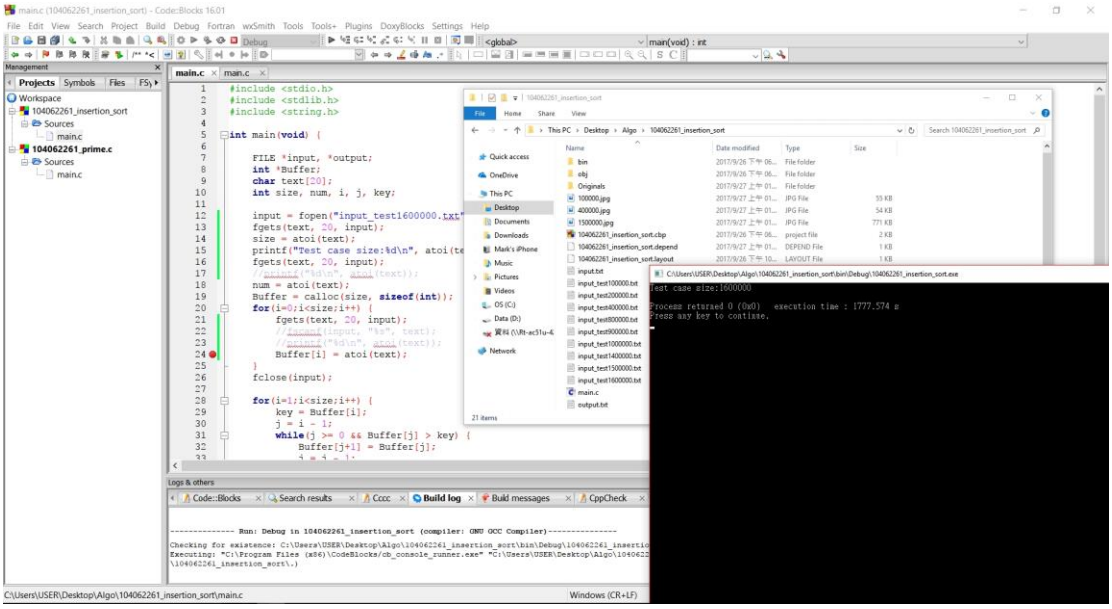
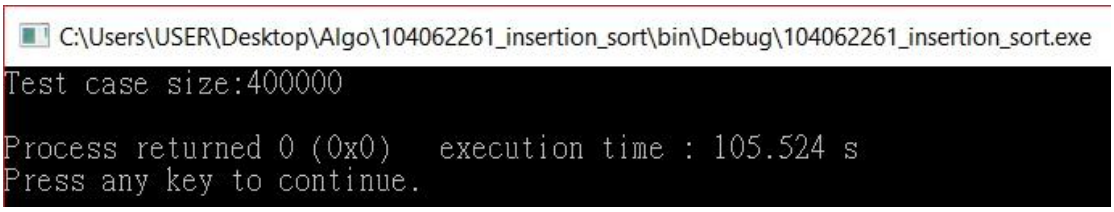
104062261 Report

一、Insertion Sort

1. CPU (included clock): 2.50 GHz
2. RAM: 16 GB
3. How many numbers can be sorted in 30mins:



I can sort about 1.6 million numbers in 30 minutes. First, I sorted some 10000 numbers test case to check my program is correct, then I sorted a 40000 numbers test case to get running time for guessing how many numbers I can sort in 30 minutes.



We knew that the time complexity of insertion sort is $O(n^2)$, so we can guess out we sort 1.6million numbers need about 1700 seconds($4^2 \times 105.525 = 1688.384$), and the result is quite close to the 1700 seconds and not over 30 minutes, but sometime it can be even lower than 1700 seconds.

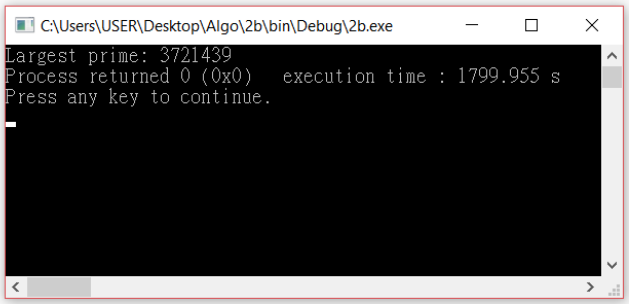
二、Prime

1. Time Complexity of your algorithm? $O(n^{\frac{1}{2}})$
2. What is the limit of your input number? 1~18446744073709551615
3. The largest prime that your algorithm can find within 30 minutes:

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void) {
    unsigned long long num = 2, prime, i;
    int is_prime;
    clock_t t1;

    t1 = clock();
    while( ((clock() - t1)/(double)(CLOCKS_PER_SEC)) < 1799.9 ) {
        is_prime = 0;
        for(i=2;i<num/2;i++){
            if(num%i==0){
                is_prime = 1;
                break;
            }
        }
        if(is_prime == 0) prime = num;
        num++;
    }
    printf("Largest prime: %llu", prime);
    return 0;
}
```



The largest prime number I can find within 30 minutes is 3,721,439. I set 2 for the opening number, and limited the time not over 1,800 seconds, so that the program will keep finding largest prime number until the time limit.