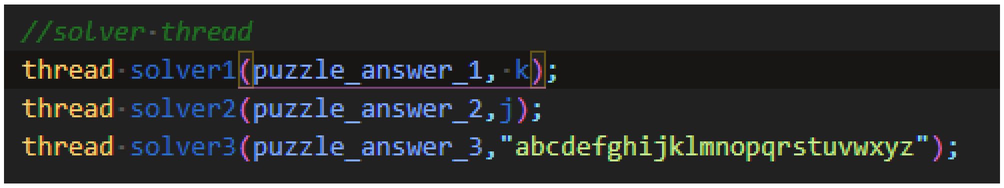
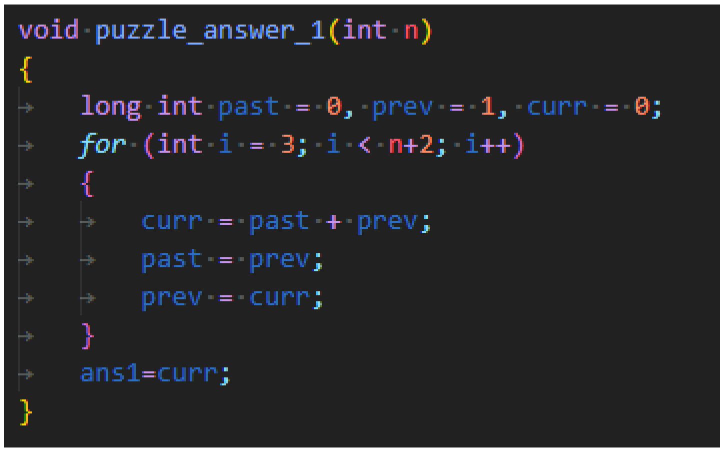
TASK 1 - Brute Force

- The first major execution of this code is to run the **SOLVER Threads** so that they can store the answers to the puzzles that have been programmed in the code and store those answers for later comparison with the **MINER Threads.**



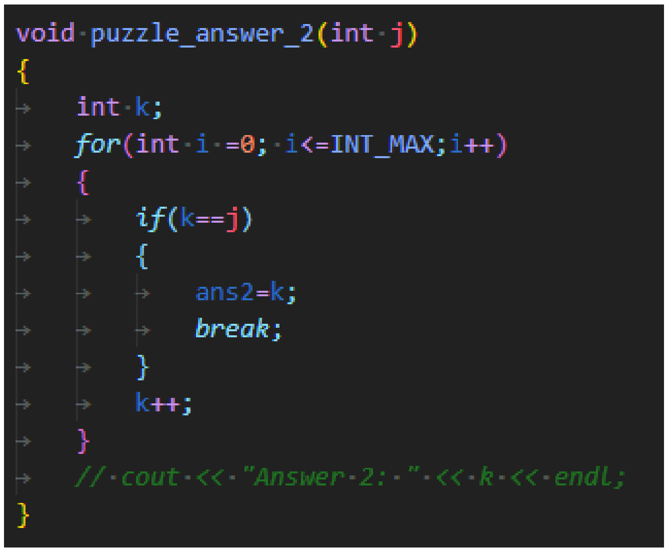
```
//solver thread
thread solver1(puzzle_answer_1, k);
thread solver2(puzzle_answer_2,j);
thread solver3(puzzle_answer_3,"abcdefghijklmnopqrstuvwxyz");
```

  - **Puzzle 1** is a random number generated from 45-80 for the Solver 1 thread to iteratively find the fibonacci number present at that place.



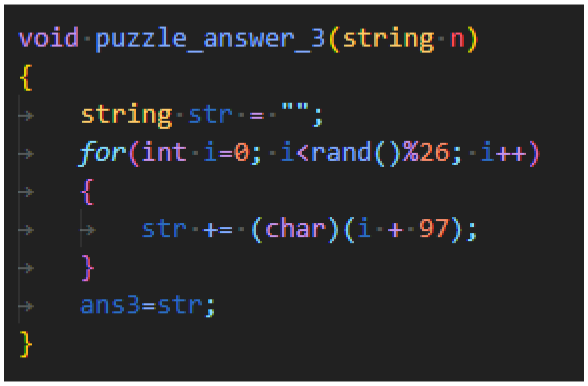
```
void puzzle_answer_1(int n)
{
    long int past = 0, prev = 1, curr = 0;
    for (int i = 3; i < n+2; i++)
    {
        curr = past + prev;
        past = prev;
        prev = curr;
    }
    ans1=curr;
}
```

  - **Puzzle 2** is also a random number generated from 1-100000 and the Solver 2 thread finds that number through iterating through the numbers until the random number matches its current iteration.



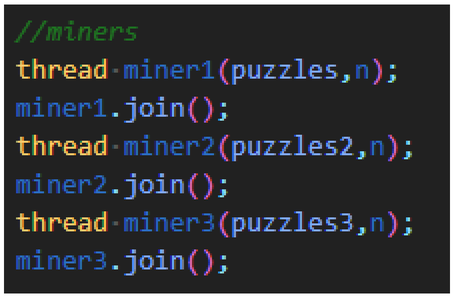
```
void puzzle_answer_2(int j)
{
    int k;
    for(int i =0; i<=INT_MAX;i++)
    {
        if(k==j)
        {
            ans2=k;
            break;
        }
        k++;
    }
    // cout << "Answer 2: " << k << endl;
}
```

- **Puzzle 3** is a string that is generated and a substring is defined , the Solver thread 3 filles a empty string and fills it letter by letter until the substring is found

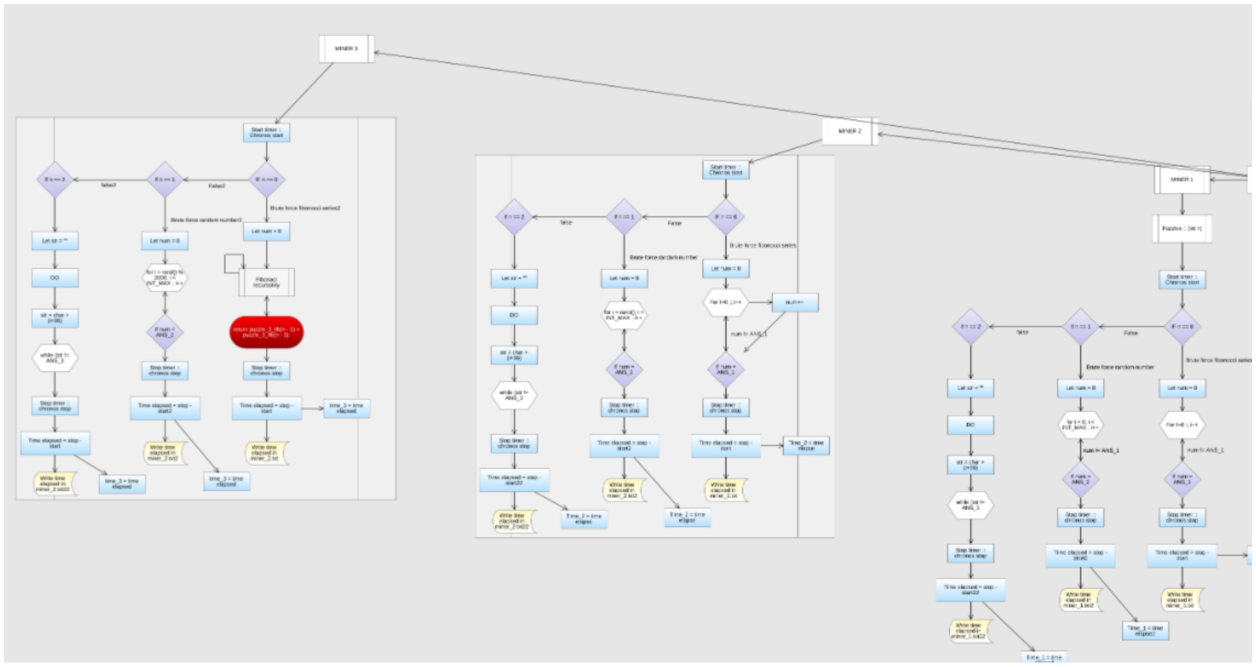


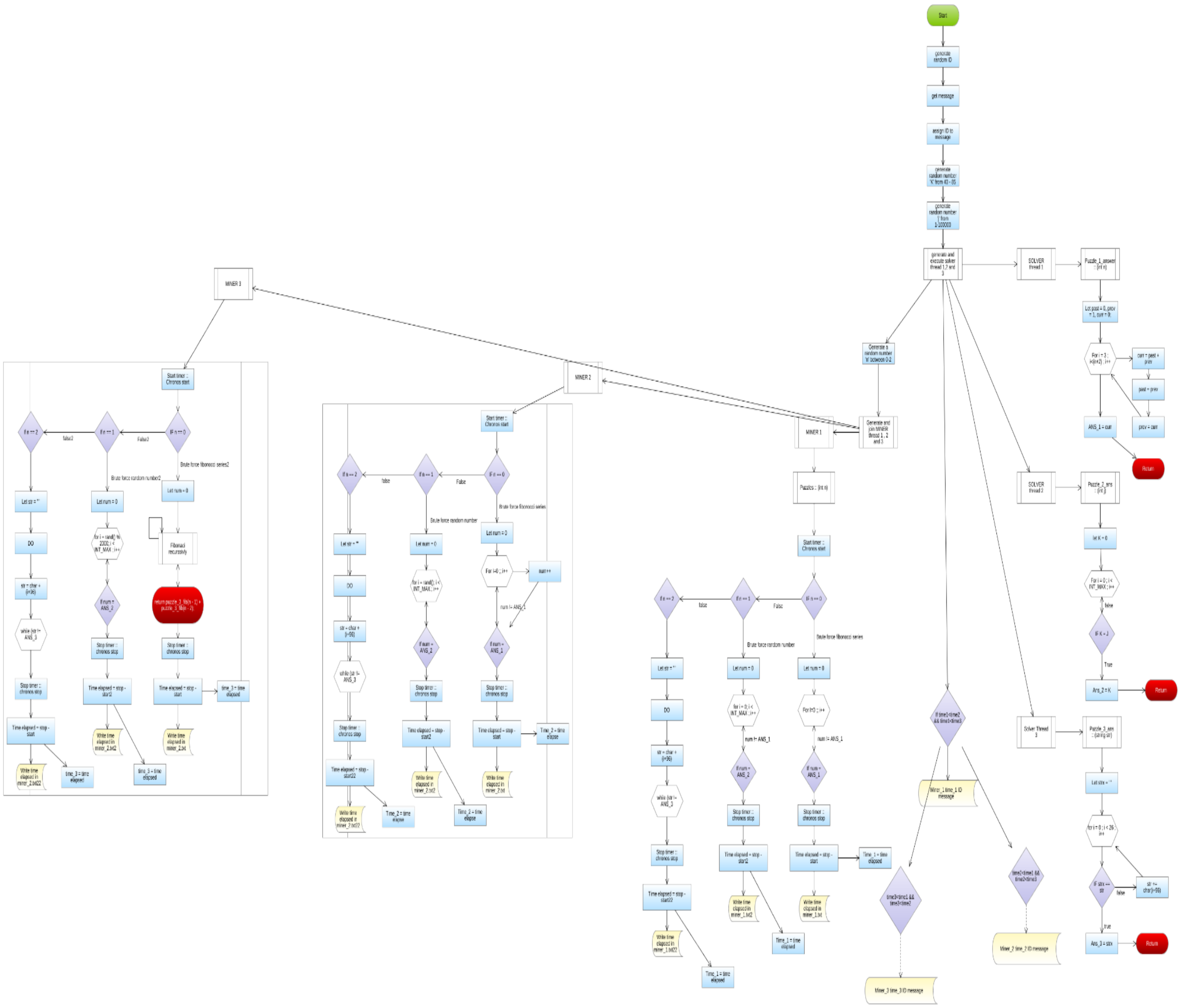
```
void puzzle_answer_3(string n)
{
    string str = "";
    for(int i=0; i<rand()%26; i++)
    {
        str += (char)(i + 97);
    }
    ans3=str;
}
```

- After all the Solver threads have been executed and the answers have been stored the **Miner Threads** are called and are all joined.



```
//miners
thread miner1(puzzles,n);
miner1.join();
thread miner2(puzzles2,n);
miner2.join();
thread miner3(puzzles3,n);
miner3.join();
```




Start
generate random ID
get message
assign ID to message
generate random number 'K' from 40 - 85
generate random number 'j' from 1-100000
generate and execute solver thread 1,2 and 3
SOLVER thread 1
Puzzle_1_answer :: (int n)
Let past = 0, prev = 1, curr = 0;
For i = 3 ; i<(n+2) ; i++
curr = past + prev
past = prev
prev = curr
ANS_1 = curr
Return
SOLVER thread 2
Puzzle_2_ans :: (int j)
let K = 0
For i = 0 ; i < INT_MAX ; i++
false
IF K = J
True
Ans_2 = K
Return
Solver Thread 3
Puzzle_3_ans :: (string str)
Let strx = ''
for i = 0 ; i < 26 ; i++
IF strx == str
false
str += char(i+96)
true
Ans_3 = strx
Return
Generate a random number 'n' between 0-2
Generate and join MINER thread 1 , 2 and 3
MINER 1
Puzzles :: (int n)
Start timer :: Chronos start
IF n == 0
False
If n == 1
false
If n == 2
Brute force fibonocci series
Brute force random number
Let num = 0
For I=0 ;; i++
num != ANS_1
If num = ANS_1
Stop timer :: chronos stop
Time elapsed = stop - start
Time_1 = time elapsed
Write time elapsed in miner_1.txt
Let num = 0
for i = 0; i < INT_MAX ; i++
num != ANS_1
If num = ANS_2
Stop timer :: chronos stop
Time elapsed = stop - start2
Write time elapsed in miner_1.txt2
Time_1 = time elapsed
Let str = ''
DO
str = char + (i+96)
while (str != ANS_3
Stop timer :: chronos stop
Time elapsed = stop - start22
Write time elapsed in miner_1.txt22
Time_1 = time elapsed
If time1<time2 && time1<time3
Miner_1 time_1 ID message
time3<time1 && time3<time2
Miner_3 time_3 ID message
time2<time1 && time2<time3
Miner_2 time_2 ID message
MINER 2
Start timer :: Chronos start
IF n == 0
False
If n == 1
false
If n == 2
Brute force fibonocci series
Brute force random number
Let num = 0
For I=0 ;; i++
num++
num != ANS_1
If num = ANS_1
Stop timer :: chronos stop
Time elapsed = stop - start
Time_2 = time elapse
Write time elapsed in miner_2.txt
Let num = 0
for i = rand(); i < INT_MAX ; i++
If num = ANS_2
Stop timer :: chronos stop
Time elapsed = stop - start2
Write time elapsed in miner_2.txt2
Time_2 = time elapse
Let str = ""
DO
str = char + (i+96)
while (str != ANS_3
Stop timer :: chronos stop
Time elapsed = stop - start22
Write time elapsed in miner_2.txt22
Time_2 = time elapse
MINER 3
Start timer :: Chronos start
IF n == 0
False2
If n == 1
false2
If n == 2
Brute force fibonocci series2
Brute force random number2
Let num = 0
Fibonaci recurssivly
return puzzle_3_fib(n - 1) + puzzle_3_fib(n - 2);
Stop timer :: chronos stop
Time elapsed = stop - start
time_3 = time elapsed
Write time elapsed in miner_2.txt
Let num = 0
for i = rand() % 2000; i < INT_MAX ; i++
If num = ANS_2
Stop timer :: chronos stop
Time elapsed = stop - start2
Write time elapsed in miner_2.txt2
time_3 = time elapsed
Let str = ""
DO
str = char + (i+96)
while (str != ANS_3
Stop timer :: chronos stop
Time elapsed = stop - start
Write time elapsed in miner_2.txt22
time_3 = time elapsed