

# C++応用プログラミング演習

## 第一回レポート課題

HT17A053 田仲凌真

[ht17a053@oecu.jp](mailto:ht17a053@oecu.jp)

## 1. 作成したプログラムのソースコード

```
// cpp2.cpp
#include <iostream>
#include <stdio.h>
#include <stdbool.h>
#define GP2_PRIME_UPPER_LIMIT 1000000
using namespace std;

bool is_prime(const unsigned n) {
    switch (n) {
        case 0:
        case 1: return false;
        case 2:
        case 3: return true;
    }

    if (n % 2 == 0) return false;
    if (n % 3 == 0) return false;

    if (n % 6 != 1 && n % 6 != 5) return false;
    for (unsigned i = 5; i*i <= n; i += 6) {
        if (n%i == 0) return false;
        if (n % (i + 2) == 0) return false;
    }
    return true;
}

int nth_prime(int a, int d, int n) {

    int current = a;
    int count = 0;

    do {
        if (is_prime(current)) ++count;
        if (count == n) return current;
        current += d;
    }
```

```

        } while (current <= GP2_PRIME_UPPER_LIMIT);

        return -1;
    }

int main()
{
    std::cout << nth_prime(367, 186, 151) << std::endl;
    std::cout << nth_prime(179, 10, 203) << std::endl;
    std::cout << nth_prime(271, 37, 39) << std::endl;
    std::cout << nth_prime(103, 230, 1) << std::endl;
    std::cout << nth_prime(27, 104, 185) << std::endl;
    std::cout << nth_prime(253, 50, 85) << std::endl;
    cin.get();
    return 0;
}

```

## 2. ソースコードの解説

与えられた正整数にたいして、その等差数列に含まれる n 番目の素数を求めるプログラミングの記述。

define で数字の最大値を記述。

bool is\_prime 関数で等差数列の記述を行う。

int nth\_prime 関数で等差数列の中から素数のみを拾う。

main 関数では与えられたサンプルの数字のインプットとアウトプットと比較し、プログラムが正しく動いているかどうかを判断するための関数。

## 3. 実行結果

```

92809
6709
12037
103
93523
14503

```

#### 4. 考察

サンプルとして渡されたインプットの値とアウトプットの値と同じ組み合わせのものがプログラムの出力により得ることができた。よってこのプログラムの処理は正しいと判断することができる。

bool 関数の動作を軽くする処理もうまく働いていることを確認することができた。

#### 5. 感想

今回の課題はプログラミングの記述よりも github への提出のほうが苦になった。あまりコマンドプロンプトでファイルなどの移動をしないので、少し手間取るところもあったが C++ のソースファイルを無事に git のレポジトリに上げることができた時はさすがにほっとした。まだ一つ目の課題なので、二つ目の課題もこんかいのように無事に提出できるように個人的に git を扱う練習をしておこうと今回の課題を進めていくうちに思った。