//

// This is example code from Chapter 2.2 "The classic first program" of

// "Programming -- Principles and Practice Using C++" by Bjarne Stroustrup

//

// keep\_window\_open() added for TDT4102, excercise 0

// This program outputs the message "Hello, World!" to the monitor

#include "std\_lib\_facilities.h"

//------------------------------------------------------------------------------'

// C++ programs start by executing the function main

//oppgave 2a - maxoftwo

int maxOfTwo(int a, int b) {

if (a > b) {

cout << a << '\n';

} else {

cout << b << '\n';

}

return 0;

}

//oppgave 2c - Fibonacci

int fibonacci(int n) {

int a = 0;

int b = 1;

cout << "Fibonacci numbers: " << '\n';

for (int i = 1; i < n+1; i++) {

cout << i << ':' << b << '\n';

int temp = b;

b += a;

a = temp;

}

cout << "----";

return b;

}

//oppgave 2d - squareNumberSum

int squareNumberSum(int n) {

int totalsum = 0;

for (int i = 1; i < n+1; ++i) {

totalsum += i\*i;

cout << i\*i << '\n';

}

return totalsum;

}

//oppgave 2e - trekanttall

int trianglebelow(int n) {

int acc = 1, num = 2;

cout << "Triangle numbers below: " << n << '\n';

while (acc < n) {

cout << acc << '\n';

acc += num;

num += 1;

}

return 0;

}

//oppgave 2f - primtall 1

bool isPrime(int n) {

for (int i = 2; i < i+1; ++i) {

if (n % i == 0) {

return false;

} else {

return true;

}

}

return 0;

}

//oppgave 2g - primtall 2

int naivePrimeNumberSearch(int n) {

for (int i = 2; i < n; ++i) {

if (isPrime(i)) {

cout << i << " is a prime." << '\n';

}

}

return 0;

}

//oppgave 2h - findgreatestdivisor

int findgreatestdivisor(int n) {

for (int divisor = n-1; divisor > 0; divisor -= 1) {

if (n % divisor == 0) {

return divisor;

}

}

return 0;

}

//main

int main()

{

cout << "Oppgave a)\n";

cout << maxOfTwo(5, 6) << '\n';

cout << "Oppgave c)\n";

cout << fibonacci(10) << '\n';

cout << "Oppgave d)\n";

cout << squareNumberSum(5) << '\n';

cout << "Oppgave e)\n";

cout << trianglebelow(12) << '\n';

cout << "Oppgave f)\n";

cout << isPrime(5) << '\n';

cout << "Oppgave g)\n";

cout << naivePrimeNumberSearch(10) << '\n';

cout << "Oppgave h)\n";

cout << findgreatestdivisor(12) << '\n';

}