Trabajo Práctico 1
Números Primos

Introducción a la Computación

1er cuatrimestre 2018

Observaciones generales:

- El trabajo se debe realizar en grupos de dos personas.
- El código fuente debe estar bien comentado. Cualquier aclaración adicional que se considere necesaria debe ser incluida como comentarios.
- El código fuente debe enviarse por mail a la lista de docentes de la materia: `icm-doc@dc.uba.ar`, indicando los integrantes del grupo. El subject/asunto del mail debe seguir el formato GrupoXX-TP1 donde XX corresponde al número del grupo asignado. Por ejemplo Grupo01-TP1.
- El programa entregado debe compilar correctamente con el `g++` instalado en las computadoras del laboratorio Turing. Se admiten carpetas comprimidas ZIP de proyectos *CLion* como modalidad de entrega.
- Se evaluará la correctitud, claridad y modularidad del código entregado.
- La fecha límite de entrega es el Domingo 15 de abril a las 23:59.

El estudio de los números primos es una parte importante de la teoría de números, la rama de la matemática que comprende el estudio de los números enteros. Los números primos están presentes en algunas conjeturas centenarias tales como la hipótesis de Riemann y la conjetura de Goldbach. Durante mucho tiempo se consideró que la aplicación de los números primos era muy limitada fuera de la matemática pura. Sin embargo, el desarrollo de la criptografía durante la década del '70 ha despertado un renovado interés sobre estos números: gran parte de los sistemas modernos de seguridad informática están basados en el hecho de que no se conoce un método eficiente para descomponer un número entero en sus factores primos. El presente trabajo tiene como objetivo la implementación de un programa que brinde funcionalidad de interés para quienes trabajan con estos números.

A continuación se enuncian algunas definiciones y resultados relevantes para el presente trabajo.

Definición 1. Un número $n \in \mathbb{N}$ es *primo* si y sólo si tiene sólo dos divisores positivos distintos: 1 y n.

Teorema 1. Existen infinitos números primos.

Teorema 2. Todo $n \in \mathbb{N}$ se puede representar de forma única como producto de factores primos.

Se pide implementar en C++ funciones para:

- dado $n \in \mathbb{N}$, determinar si es primo;
- dado $n \in \mathbb{N}$, obtener la cantidad de números primos menores o iguales a n.
- dado $n \in \mathbb{N}$, obtener la cantidad de divisores primos de n.
- dados $n, i \in \mathbb{N}$, obtener el i-ésimo divisor primo de n;
- dados $n, i \in \mathbb{N}$, obtener la potencia del i-ésimo divisor primo de n en la descomposición de n en factores primos.

Las funciones deben seguir la siguiente signatura:

```
bool esPrimo(int n)
int  cantidadPrimosMenoresOIguales(int n)
int  cantidadDivisoresPrimos(int n)
int  iesimoDivisorPrimo(int n, int i)
     // -1 si n no tiene i divisores primos
int  potenciaIesimoDivisorPrimo(int n, int i)
     // -1 si n no tiene i divisores primos
```

Nota: Los factores primos deben enumerarse siempre en orden creciente. Ver ejemplos de uso más abajo.

El programa presentado deberá poder ejecutarse desde la línea de comandos y recibirá como parámetros la función que se desea ejecutar más los argumentos necesarios para dicha función. A continuación se exponen usos del programa con el resultado esperado:[1]

- `./ejecutar esPrimo 2` → "`si`"
- `./ejecutar esPrimo 6` → "`no`"
- `./ejecutar cantidadPrimosMenoresOIguales 6` → "`3`"
- `./ejecutar cantidadPrimosMenoresOIguales 7` → "`4`"
- `./ejecutar cantidadDivisoresPrimos 6` → "`2`"
- `./ejecutar cantidadDivisoresPrimos 7` → "`1`"
- `./ejecutar iesimoDivisorPrimo 10 1` → "`2`"
- `./ejecutar iesimoDivisorPrimo 10 2` → "`5`"
- `./ejecutar iesimoDivisorPrimo 10 3` → "`10 no tiene 3 divisores primos`"
- `./ejecutar potenciaIesimoDivisorPrimo 48 1` → "`4`"
- `./ejecutar potenciaIesimoDivisorPrimo 48 2` → "`1`"
- `./ejecutar potenciaIesimoDivisorPrimo 48 3` → "`48 no tiene 3 divisores primos`"

Observaciones:

- Definir todas las funciones auxiliares que se consideren necesarias.
- Sólo está permitido importar las bibliotecas `iostream` y `string`.
- La impresión por pantalla (`cout`) solamente puede ser invocada desde el cuerpo de la función `main`.
- Puede suponerse que el usuario siempre invocará al programa de manera correcta. Es decir, si hay errores en la cantidad, tipo o valor de los parámetros de entrada, no se espera comportamiento alguno del programa (puede colgarse o morir, por ejemplo).

[1] Se asume en estos ejemplos que el archivo ejecutable resultado de la compilación se llama "ejecutar".