

1. Objetivos

Ejercitar conceptos básicos de programación C++, implementando un programa y su correspondiente documentación que resuelva el problema descrito más abajo.

2. Alcance

Este trabajo práctico es de elaboración grupal, evaluación individual, y de carácter obligatorio para todos alumnos del curso.

3. Requisitos

El trabajo deberá ser entregado personalmente, en la fecha estipulada, con una carátula que contenga los datos completos de todos los integrantes, un informe impreso de acuerdo con lo que mencionaremos en la sección 5, y con una copia digital de los archivos fuente necesarios para compilar el trabajo.

4. Descripción

En este trabajo implementaremos la transformada discreta de Fourier (DFT, por sus siglas en inglés). Este algoritmo aplica una transformación a una secuencia de N puntos según la fórmula

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad k = 0, \dots, N-1; \quad W_N = e^{-j(2\pi/N)} \quad (1)$$

Para obtener la secuencia original, se puede antitransformar utilizando la siguiente fórmula (IDFT):

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn} \quad n = 0, \dots, N-1 \quad (2)$$

Esta última es muy parecida a la primera salvo por un factor de $\frac{1}{N}$ y el signo del exponente de las W_N .

4.1. Formato de entrada y salida

El archivo de entrada será un archivo de texto con pares ordenados de complejos (`re`, `im`), separados por espacios, con una señal por línea. La salida tendrá el mismo formato, siendo cada línea la transformada o antitransformada de la señal correspondiente, según el orden de entrada (la primera señal en la salida es la transformada de la primera señal de la entrada, lo mismo para la segunda y así sucesivamente).

4.2. Interfaz

La interacción con el programa es a través de la línea de comando. Se incluirán las siguientes opciones:

- `-i`, o `--input`, donde se pasará el nombre del archivo con la señal de entrada de acuerdo al formato explicado en la sección 4.1. En caso de ser `-` o no estar especificado se deberá leer desde la entrada estándar.
- `-o`, o `--output`, donde se pasará el nombre del archivo donde deben ser escritas las señales de salida. Otra vez, en caso de ser `-` o no estar especificado se escribirá sobre la salida estándar.

-
- `-m`, o `--method`, donde se especificará qué transformada se hará a la señal, siendo las posibilidades DFT o IDFT. En caso de no ser especificado, se tomará por defecto la DFT.

4.3. Ejemplos

Se incluye para el trabajo práctico un conjunto de archivos de texto con señales y sus transformadas en el formato requerido para éste.

El ejemplo más simple consiste en una entrada vacía. Observar que la salida es también vacía:

```
$ touch entrada1.txt
$ ./tp0 -i entrada1.txt -o salida1.txt
$ cat salida1.txt
$
```

Los siguientes ejemplos son de transformadas simples. Se puede ver que al transformar una señal y, a continuación, antitransformarla, se recupera la señal original:

```
$ cat cte4.txt
(1, 0) (1, 0) (1, 0) (1, 0)
$ ./tp0 -i cte4.txt -o Tcte4.txt -m DFT
$ cat Tcte4.txt
(4, 0) (0, 0) (0, 0) (0, 0)
$

$ cat cte4.txt
(4, 0) (0, 0) (0, 0) (0, 0)
$ ./tp0 -i Tcte4.txt -o cte4.txt -m IDFT
$ cat Tcte4.txt
(1, 0) (1, 0) (1, 0) (1, 0)
$
```

5. Informe

El informe deberá incluir:

- Documentación relevante al diseño e implementación del programa.
- Documentación relevante a los algoritmos involucrados en la solución del trabajo.
- Documentación relevante al proceso de compilación: cómo obtener el ejecutable a partir de los archivos fuente.
- Las corridas de prueba, con los comentarios pertinentes.
- El código fuente, en lenguaje C++ (en dos formatos, digital e impreso).
- Este enunciado.

6. Fechas

La última fecha de entrega es el jueves 19 de abril.