

# Universidad Nacional de Asunción - Facultad Politécnica

## Primer Examen Final de Programación

Fecha: 18 de junio de 2022

### Directivas y Recomendaciones

- Los comentarios aclaratorios pueden ayudar en la corrección del examen, influyendo favorablemente en su calificación.
- La duración del examen es de 120 minutos.
- El código fuente para cada tema debe subirse al aula virtual de la materia (en EDUCA), en la actividad correspondiente en el VPL.

### Ejercicio 1: Procesamiento de datos de satélites (100%)

En este ejercicio trabajaremos con los datos de telemetría enviados por satélites en código Morse. Cada uno de los mensajes tiene el siguiente formato:

Identificador del Satélite			% Batería		Es Día?
Z	P	6	5	2	1

Se utiliza el código Morse para transmitir estos mensajes. Esta codificación asigna una serie de puntos y rayas a cada letra del alfabeto y a cada dígito. Los caracteres de interés para este problema son:

Caracter	Código	Caracter	Código
1	.- - - -	6	- . . . .
2	. . - - -	7	- - . . .
3	. . . - -	8	- - - . .
4	. . . . -	9	- - - - .
5	. . . . .	0	- - - - -
Z	- - . .	P	. - - .

**Observación:** Si se reciben otra combinación de puntos y rayas, se asignará el caracter 'X'.

Usted cuenta con un archivo de texto que contiene una serie de líneas. Cada una de estas líneas contiene un conjunto de caracteres (en código Morse), los cuales están separados por el caracter ';' (punto y coma). Se presenta a continuación un archivo de ejemplo:

datos.txt	
1	- - . . ; . - - . ; - . . . . ; . . . . . ; . . - - - ; . - - - -
2	- - . . ; . - - . ; - . . . . ; - - - - . ; . - - - - ; - - - - -
3	. - - - ; . - - . ; - . . . . ; . . . . . ; . . - - - ; . - - - -
4	. - - - ; . - - . ; - . 5 . . . ; . . . . . ; 6 . . - - - ; . - - - -

Su trabajo consiste en escribir un programa en C que procese los datos del archivo de entrada (se debe solicitar al usuario que ingrese el nombre del archivo); e imprima los datos correspondientes del satélite en el formato que se especifica. El resultado del procesamiento del archivo de ejemplo es:

```
Ingrese el nombre del archivo: datos.txt
```

```
Resultados del procesamiento:
```

```
ZP6, 52, 1, 'Es el GuaraniSat1'
```

```
ZP6, 91, 0, 'Es el GuaraniSat1'
```

```
XP6, 52, 1
```

```
La línea no es válida...
```

```
Process returned 0 (0x0)    execution time : 2.597 s
```

```
Press any key to continue.
```

En el ejemplo presentado, se puede notar lo siguiente:

- Si los caracteres de una línea del archivo de entrada no corresponden a caracteres válidos para el código Morse ('-' y '.') ni al separador de caracteres (';'), entonces se señala en la salida que dicha línea no es válida. En el archivo del ejemplo, la línea 4 no es válida.
- Para cada línea en formato válido, se imprime el identificador del satélite, seguido del % de batería, y finalmente se coloca el valor que indica si es de día.
- El identificador **ZP6** corresponde al satélite paraguayo **GuaraniSat1**. Por lo tanto, debe agregarse un mensaje adicional en pantalla para las líneas del archivo que correspondan a este satélite.

### Observaciones:

- Por simplicidad, se puede suponer que el archivo de entrada se encuentra en la misma carpeta que el ejecutable del programa.
- Se puede suponer que el tamaño máximo de una línea es de 100.
- Se puede suponer que si una línea es válida (en cuanto a los caracteres), contiene todos los datos del satélite en el formato especificado al inicio (exactamente 6 caracteres en código Morse, donde los 3 últimos representan a dígitos del 0 al 9). Por lo tanto, no se necesita realizar validaciones adicionales.
- Recuerde que el archivo mostrado en el enunciado es un ejemplo. Su programa debería funcionar con cualquier archivo que cumpla con las especificaciones del problema.

### Criterios de corrección

- Sintaxis: 10%
- Estrategia de solución: 65%
  - Lectura de los datos del archivo: 10%
  - Determinación de una línea válida: 15%
  - Separación de los caracteres en código Morse: 15%
  - Conversión de un caracter en código Morse: 15%
  - Impresión de los resultados en pantalla, en el formato especificado: 10%
- Implementación: 25%
  - Implementación y uso de funciones: 20%
  - Presentación del código (uso de comentarios y tabulación): 5%