

Tecnología Digital 1: Introducción a la Programación

Trabajo Práctico 1 - Parte B

Escuela de Negocios, UTDT

Primer semestre 2025

El objetivo de esta entrega es implementar las funciones especificadas en la Parte A, demostrar que las implementaciones son correctas y desarrollar un programa que funcione como interfaz de usuario para utilizar las funcionalidades implementadas.

Para que la resolución de esta etapa sea equitativa para todos los grupos, deberán usar la especificación y los tests de unidad provistos por la cátedra (ver los archivos adjuntos de este enunciado).

Para cada una de las funcionalidades incluidas en la especificación, se pide:

- (a) Pensar un algoritmo y llevarlo a un programa en Python que resuelva el problema.
- (b) Asegurarse de que la función pase los casos de test proporcionados.
- (c) Para las funciones que contengan algún ciclo, demostrar que estos terminan, escribir predicados invariantes que describan el trabajo que realizan, y usarlos para mostrar que los programas son correctos. En el caso de que una función sin ciclo invoque a una función auxiliar que contenga algún ciclo, solo deberán darse estas justificaciones sobre la función auxiliar.

El archivo `tp1.py` será el programa principal del TP. Al ejecutarlo, se deberá mostrar un menú para elegir la funcionalidad que se desea ejecutar. Dependiendo de la opción elegida y los argumentos ingresados, el programa tendrá que mostrar la respuesta que corresponda. A continuación puede observarse parte de una ejecución del programa en la cual el usuario consulta la similitud binaria de prefijo (*simBP*) entre 91 y 380:

```
Funciones disponibles
-----
A. Similitud binaria de prefijo [n,m]
B. Cantidad de enteros con simBP en intervalo [n,a,b,k]
C. Existe entero con simBP en intervalo [n,a,b,k]
D. Entero con mayor simBP en intervalo [n,a,b]
F. Finalizar

Seleccione una opción: A
Ingrese n: 91
Ingrese m: 380
La simBP entre 91 y 380 es 4.

Presione ENTER para continuar.
```

La ejecución concluye únicamente si se selecciona la opción `Finalizar`. En cualquier otro caso, se muestra nuevamente el menú de opciones.

A modo de ejemplo, la siguiente tabla ilustra los mensajes que debe imprimir el programa para algunas funciones y valores ingresados:

Función seleccionada	Argumentos	Mensaje por pantalla
SimBP	91,380	La simBP entre 91 y 380 es 4.
Cantidad con simBP	91,20,1000,3	126 enteros en [20,1000] tienen simBP 3 con 91.
Existe entero con simBP	91,20,1000,3	En [20,1000] hay algún entero con simBP 3 con 91.
Existe entero con simBP	91,100,150,7	En [100,150] no hay enteros con simBP 7 con 91.
Entero con mayor simBP	91,20,50	En [20,50] el entero con mayor simBP con 91 es 45.

Observar que el archivo `tp1.py` provisto con este enunciado implementa casi por completo este menú interactivo. Sólo falta completar el código que se ejecuta al elegir cada opción.

Se deben entregar los siguientes archivos:

- **`simbp.py`**, con el código de las funciones requeridas y cualquier otra función auxiliar que se defina. Es obligatorio indicar el tipo de todas las variables y funciones utilizando sugerencias de tipos (*type hints*). Para las funciones requeridas, se deben utilizar las especificaciones provistas por la cátedra. En caso de incluir funciones auxiliares, sus especificaciones deben seguir el mismo formato. Este archivo debe contener únicamente las funciones; es decir, **no** debe ser directamente ejecutable.
- **`tp1.py`**, con el código principal para ejecutar el programa. Este código es el encargado de imprimir el menú, invocar a las funciones definidas en `simbp.py` y mostrar los resultados por pantalla. Tomar como punto de partida el *template* adjunto.
- **`informe.pdf`**, un documento con las respuestas al punto (c) enunciado arriba. Incluir cualquier aclaración adicional que se considere necesaria sobre cualquier parte del trabajo. Se espera que este documento sea conciso, de no más de cuatro páginas y en formato PDF.

Observaciones:

- Los grupos de esta entrega deben ser los mismos que los de la entrega anterior.
- La fecha límite de entrega es el **martes 15 de abril a las 23:55**.
- Los archivos deben subirse al formulario *TP1: Entrega Parte B* del campus virtual.
- El programa entregado debe correr correctamente en Python3.
- Sólo pueden usarse las instrucciones y estructuras de control vistas en clase. No pueden usarse iteradores (`for`), ni *slicing* sobre strings.
- Para obtener la representación binaria de un número natural, se recomienda usar las funciones `bin` y `replace` de Python. La expresión `bin(n)` devuelve un string con formato `'0bx'`, donde `x` es la representación en base 2 del entero `n`. Para descartar el prefijo `'0b'` puede ejecutarse `bin(n).replace('0b', '')`.
- Puede suponerse que el usuario siempre invocará las funciones de manera correcta. Es decir, si hay errores de tipo o valor de los argumentos provistos, no se espera comportamiento alguno del programa (podría colgarse o terminar en un error, por ejemplo).
- **Forma de evaluación.** Se evaluarán dos aspectos de esta entrega: (i) el *código*: no sólo su correcto funcionamiento, sino también que se sigan las buenas prácticas de programación vistas desde la primera clase: uso adecuado de tipos y estructuras de control, claridad en los nombres de variables y funciones auxiliares, comentarios, reutilización de funciones, etc.; y (ii) las demostraciones de terminación y correctitud de ciclos.