



## Tarea 1

### Aspectos preliminares

El objetivo de esta evaluación es que profundicen en los aspectos prácticos y teóricos de los tópicos de los capítulos 1 y 2 del curso. En esta evaluación hay preguntas teóricas y de programación, que estarán marcadas con **(T)** o una **(P)** al lado del número que las identifica, para diferenciarlas claramente.

La evaluación está diseñada para que deban buscar material y recursos que no están en el sitio del curso para contestar las preguntas. Tomando esto en consideración, como esta tarea tiene un fin pedagógico, la regla de integridad académica que deben cumplir es la siguiente:

*Toda respuesta a cualquier pregunta debe ser desarrollada y escrita individualmente. Esto significa que al momento de editar la respuesta, no se debe estar mirando o usando un texto escrito por otra persona, un video hecho por otra persona, o cualquier material que haya sido desarrollado por otra persona. Tampoco es válido memorizar una respuesta obtenida de otra fuente y luego escribirla. En otras palabras, lo que sea que se escriba debe provenir directamente de ustedes. Sin embargo, antes de comenzar a escribir cada respuesta pueden estudiar y comentar la pregunta con otros alumnos (no la respuesta, solo la pregunta), leer libros, mirar videos, o realizar cualquier otra acción que los ayude a responder la pregunta. Esta regla aplica tanto a pregunta teóricas como prácticas.*

Finalmente, la entrega deberá incluir un archivo llamado INTEGRIDAD.txt, en donde afirman que han leído, entendido y respetado la regla de arriba. De no cumplirse esto, la tarea no será corregida y obtendrán la nota mínima.

### Entrega

La fecha de entrega es el miércoles 16/09 hasta las 23:59. El formulario para la entrega se encuentra en la siguiente dirección <https://forms.gle/xDJr4ubbCa7YDpGU8>. En este formulario encontrará la descripción del formato de entrega para cada una de las preguntas. Tenga en consideración que las respuestas solo pueden ser entregadas 1 vez.

### 1. Representaciones Numéricas

- a) (T) Asumiendo que se usan registros del mismo tamaño y que procesar floats es siempre más lento que procesar enteros, ¿cuál podría ser una ventaja de representar enteros como floats en un computador? Justifique su respuesta detalladamente. **(1.0 pto.)**
- b) (T) Un programa lee desde la memoria principal de un computador, cuatro palabras contiguas a partir de la dirección 0x36. Al interpretar estos datos como un número entero, el programa obtiene el valor decimal 230. ¿Cuál sería el valor del número leído, si se lee el dato cambiando el endiannes original y luego reinterpreta como float? Justifique su respuesta mostrando los pasos que lo llevaron a ella. **(1.0 pto.)**
- c) (T) En una representación posicional de número enteros de **K** cifras con base **N**, ¿cuántos números son iguales a su complemento a **N**? Justifique su respuesta formal y detalladamente. **(2.0 ptos.)**

- d) (P) Considere una lista  $E$  de números naturales menores o iguales a 100, donde  $E_i$  es el elemento ubicado en el índice  $i$  de esta. Escriba un programa en Python que reciba como entrada una lista  $E$  y retorne una lista  $S$ , donde el elemento en el índice  $i$  de esta corresponda al sucesor de  $2^{E_i}$  en el tipo de dato `double`. **(2.0 ptos.)**

**Nota:** se define como sucesor de un número  $x$  en representación `double`, como el número entero mayor más cercano a  $x$ , que puede ser representado como `double`.

## 2. Programabilidad

De acuerdo a Wikipedia, una máquina de Turing es un “dispositivo que manipula símbolos en una cinta de acuerdo con una tabla de reglas”<sup>1</sup>. Es posible demostrar que una máquina de Turing se puede adaptar para simular la lógica de cualquier algoritmo ejecutable por un computador. En este ejercicio, deberá (de)mostrar la implicación inversa, es decir, que un computador puede ejecutar lo mismo que una máquina de Turing.

- a) (T) Diagrame los componentes de un computador que simule el funcionamiento de una máquina de Turing. Haga explícitas las relaciones entre los elementos de la máquina de Turing y del computador. **(3.0 ptos.)**
- b) (T) Describa las instrucciones para la máquina del ítem anterior. Indique opcodes y señales de control. **(2.0 ptos.)**
- c) (T) ¿Son exactamente equivalentes la máquina de Turing y el computador descrito anteriormente? Justifique su respuesta. **(1.0 pto.)**

## 3. Assembly

Escriba los siguientes programas en el assembly del computador básico. Utilice el emulador del computador básico disponible en el sitio del curso para ejecutar y probar sus soluciones. Para cada pregunta, se indicarán los nombres de los labels donde se encontrarán los datos y donde se deben entregar los resultados. Estos labels **no pueden ser usados en sus respuestas finales**, ya que serán de uso reservado para el corrector automático. Fuera de estos labels, no hay limitante para utilizar otros, tanto para datos como para subrutinas. Con el fin de evitar errores de formato, se subirán al sitio del curso archivos de muestra que podrán utilizar con el emulador.

- a) (P) Dado un grafo no dirigido y dos de sus nodos, escriba un programa que encuentre la distancia (número de aristas) que separan a estos dos nodos en el grafo. Para construir su solución, debe asumir lo siguiente:
- El grafo será codificado como una matriz de adyacencia, almacenada a partir de la dirección indicada por el label `M` en orden de filas. Cada entrada en esta matriz utilizará 1 byte.
  - La cantidad de nodos del grafo se almacenará en la dirección indicada por el label `N`. La cantidad de nodos del grafo siempre estará en el rango  $[2, 12]$ .
  - Los índices de los nodos comienzan en 0.
  - El índice del primer nodo se almacenará en la dirección indicada por el label `n1` y el índice del segundo se almacenará en la dirección indicada por el label `n2`.
  - La distancia entre los nodos debe ser almacenada en la dirección indicada por el label `res`.
- (3.0 ptos.)**
- b) (P) Dada un arreglo de número enteros, donde cada uno se encuentra en el rango  $[-60, 60]$ , encuentre el índice de aquel que corresponde a la mediana de este conjunto. Para construir su solución, debe asumir lo siguiente:
- El arreglo estará codificado de manera continua en memoria, a partir de la dirección indicada por el label `arr`. Cada entrada del arreglo utilizará 1 byte.
  - La cantidad de elementos en el arreglo se almacenará en la dirección indicada por el label `N`. La cantidad de elementos del arreglo siempre estará en el rango  $[0, 50]$ .
  - Los índices de los elementos del arreglo comienzan en 0.
  - El índice de la mediana debe ser almacenado en la dirección indicada por el label `res`.

**(3.0 ptos.)**

---

<sup>1</sup>[https://es.wikipedia.org/wiki/M%C3%A1quina\\_de\\_Turing](https://es.wikipedia.org/wiki/M%C3%A1quina_de_Turing)

## Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

*“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”*

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario.