



IIC2115 – Programación como Herramienta para la Ingeniería (I/2020)

## Actividad de Participación 2 - Estructuras de Datos

### Objetivos

- Aplicar estructuras de datos para la resolución eficiente de problemas.

### Instrucciones

Esta es una evaluación que implica realizar la programación de las soluciones de manera estrictamente individual. Así, si usted utiliza código de terceros para la realización de esta evaluación, habrá cometido una falta que el profesor, de saberlo, deberá reportar a la dirección de pregrado de la Escuela. Teniendo esto presente, ud. podrá intercambiar información con terceros acerca de esta evaluación y también buscar la información en otros recursos, sin que esto sea considerado una falta a la ética académica.

**Antes y después de realizar la actividad, deberá responder en el Siding un cuestionario relativo a estas instrucciones. El no responder estos cuestionarios en los tiempos establecidos implicará que su solución no será corregida.**

### Entrega del código

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **P02**.
- **Entrega:** lunes 30 de marzo a las 16:50 hrs.
- **Formato de entrega:** archivo python notebook (**.ipynb**) con la solución al problema. El archivo debe estar ubicado en la carpeta **P02**. No se debe subir ningún otro archivo a la carpeta. Utilice múltiples celdas de texto y código para facilitar la revisión de su programa.

- **Descuentos:** Entregas que no cumplan el formato de entrega tendrán un descuento de 0.5 ptos.
- **Entregas atrasadas, con errores de sintaxis y/o que generen excepciones serán calificadas con nota 1.0.**

## Introducción

En esta actividad deberán solucionar dos problemas de programación, que pueden ser resueltos de manera eficiente si se utilizan estructuras de datos, en vez de un enfoque de fuerza bruta. Para cada problema, se indicará que estructura de datos deberán utilizar.

## El problema: corrección de claves secretas corruptas

Considere la siguiente situación: ud. recibe una clave secreta en forma de número natural, que debe ser utilizada para desactivar una bomba. Lamentablemente, por problemas en el canal de comunicación, la clave recibida tiene algunos errores. Después de realizar análisis estadísticos, se han identificado dos posibles tipos de corrupción en el mensaje. Para cada uno de ellos, escriba un programa que permita obtener la clave correcta a partir del número original. Lo casos son los siguientes:

- a) (Stack) El valor recibido tiene dígitos contiguos duplicados. Elimínelos y entregue como respuesta el número resultante. Un ejemplo de ejecución del algoritmo es el siguiente:

### Código

```
def eliminar_duplicados_contiguos(n):  
    #solucion al problema  
  
    n = 1242241  
    n_ = eliminar_duplicados_contiguos(n)  
    print(n_)
```

### Salida

```
124241
```

- b) (Diccionario) El valor recibido tiene dígitos duplicados. Elimínelos manteniendo sólo la primera aparición de cada dígito y entregue como respuesta el número resultante. Un ejemplo de ejecución del algoritmo es el siguiente:

### Código

```
def eliminar_duplicados(n):  
    #solucion al problema  
  
    n = 1242241  
    n_ = eliminar_duplicados(n)  
    print(n_)
```

### Salida

124

## Corrección

Para la corrección de esta actividad, se revisarán dos aspectos, cada uno con igual valor en la nota (50%). El primero serán los contenidos y mecanismos utilizados para resolver cada uno de los problemas propuestos. De este modo, es importante que comente correctamente su código para que sea más sencilla la corrección. Recuerde que debe utilizar las estructuras de datos adecuadas a cada problema. Además, se evaluará el orden de su trabajo.

El segundo aspecto será la correctitud de los resultados, mediante la evaluación de distintos valores de entrada, que se entregarán al programa de acuerdo al formato indicado en cada problema. Para obtener el puntaje máximo, el resultado entregado por su solución para cada valor de entrada debe ser correcto.

## Política de Integridad Académica

*“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. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por “copia” o “plagio” se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.

```
def calcular_promedio():
    notas = [6, 4, 2, 3, 5]
    n = 5
    i = 0
    promedio = 0
    while i < n:
        promedio += notas[i]
        i += 1
    promedio = promedio / n
    print(promedio)
```