Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
IC-4700 Lenguajes de programación
Asignación #2 (proyecto para aprender características imperativas y secuenciales del lenguaje Go)

Historial de revisiones:
- 2020.11.02: Versión base (v0).
- 2020.11.22, 24: Delimitar trabajo (v1).

Lea con cuidado este documento. Si encuentra errores en el planteamiento[1], por favor comuníqueselos inmediatamente al profesor.

Objetivo
Al concluir esta asignación, Ud. estará familiarizado con las principales características de programación *secuencial* del lenguaje **Go** (Golang). El aprendizaje será auto-guiado.

Bases
- Libros y documentos sobre el lenguaje de programación **Go** (Golang).

Desarrollo
Cada grupo de estudiantes trabajará los aspectos indicados a continuación, conforme la distribución indicada más abajo:

1. Diseñar y construir una función que genere un *arreglo* de tamaño *n* con números pseudo-aleatorios obtenidos mediante el método de *congruencia lineal multiplicativa*, a partir de una *semilla* dada. Los valores deben estar en el intervalo 0 .. 31. La semilla deberá ser un número primo entre 11 y 101. El período debe ser $\geqslant 2048$. *n* puede ser cualquier número en el intervalo 10 .. 10000.
2. Diseñar y construir una función que genere un gráfico de barras a partir de un arreglo de *n* números enteros, cada uno de los cuales pertenece al intervalo 0 .. 31. Las barras deben ser verticales y de tamaño proporcional al número que representan.
3. Diseñar y construir una función que ordene ascendentemente un arreglo de enteros mediante el método de ordenamiento (cuadrático) de *intercambio* (conocido como el 'método de la burbuja').
4. Diseñar y construir una función que ordene ascendentemente un arreglo de enteros mediante el método de ordenamiento (cuadrático) de *selección*.
5. Diseñar y construir una función que ordene ascendentemente un arreglo de enteros mediante el método de ordenamiento (cuadrático) de *inserción*.
6. Diseñar y construir una función que ordene ascendentemente un arreglo de enteros mediante el método de ordenamiento *n* log (*n*) conocido como *Quicksort*.
7. Diseñar y construir una función que ordene ascendentemente un arreglo de enteros mediante el método de ordenamiento *n* log (*n*) conocido como *Treesort* (versión *Treesort 3*).
8. Diseñar y construir una función que ordene ascendentemente un arreglo de enteros mediante el método de ordenamiento *n* log (*n*) conocido como *Heapsort*.

Distribución
Considere los ítemes de la lista anterior.
- Si el grupo es de 1 miembro: escoge 1 ítem *k* del conjunto {3, 4, 5}, desarrolla en Go los algoritmos correspondientes a los ítemes *k* y *k* + 3.
- Si el grupo es de 2 miembros: escoge 2 ítemes distintos *j* y *k* del conjunto {3, 4, 5}, desarrolla en Go los algoritmos correspondientes a los ítemes *j*, *k*, *j* + 3, *k* + 3.
- Si el grupo es de 3 miembros: desarrolla en Go los algoritmos correspondientes a los ítemes 3 al 8.
- Si el grupo es de 4 miembros: investiga sobre los métodos de ordenamiento *Mergesort* (para arreglos) y *Smoothsort* (debido a E.W.D. Dijkstra) y desarrolla uno de ellos en Go.

Todos los grupos deben desarrollar los ítemes 1 y 2, aunque trabajen individualmente.

[1] El profesor es un ser humano, falible como cualquiera.

Informe técnico
Deberán preparar un informe técnico que incluya:

- Portada que identifique a los autores del informe, con sus carnets.
- Introducción al informe.
- [por completar]
- Referencias: los libros, revistas y sitios Web que utilizaron durante la investigación y el desarrollo de su proyecto. Citar toda fuente consultada.

Archivos por entregar

- Deben guardar su trabajo en *una* carpeta comprimida (formato **`.zip`**) según se indica abajo[2]. Esto debe incluir:
  - Informe técnico, en un solo documento, según se indicó arriba. El documento debe estar en formato `.pdf`.
  - Apéndices con código fuente de los programas desarrollados por su grupo.

Entrega
Fecha límite: **por definir,** antes de las 23:55. No se recibirán trabajos después de la fecha y la hora indicadas.

Los grupos pueden ser de *hasta* **4** personas.

Debe enviar por correo-e el ***enlace***[3] a un archivo comprimido almacenado en la nube con todos los elementos de su solución a estas direcciones: `itrejos@itcr.ac.cr` y `joscaes12@gmail.com` (José Alfredo Campos Espinoza).

El asunto (*subject*) debe ser:
`IC-4700 – Asignación 2–` carnet + carnet + carnet + carnet.

Los carnets deben ir ordenados ascendentemente.

Si su mensaje no tiene el asunto en la forma correcta, su trabajo será castigado con −10 puntos; podría darse el caso de que su proyecto no sea revisado del todo (y sea calificado con 0) sin responsabilidad alguna del profesor o del asistente (caso de que su mensaje fuera obviado por no tener el asunto apropiado). Si su mensaje no es legible (por cualquier motivo), o contiene un virus, o es entregado en formato **`.rar`**, la nota será 0.

La redacción y la ortografía deben ser correctas. La citación de sus fuentes de información debe ser acorde con los lineamientos de la Biblioteca del TEC. El profesor tiene *altas expectativas* respecto de la calidad de los trabajos escritos y de la programación producidos por estudiantes universitarios de la carrera de Ingeniería en Computación del Tecnológico de Costa Rica. Los profesores esperamos que los estudiantes tomen en serio la comunicación profesional.

Referencias

- https://en.wikipedia.org/wiki/Linear_congruential_generator

---

[2] **No use** formato **`.rar`**, porque es rechazado por el sistema de correo-e del TEC.
[3] Los sistemas de correo han estado rechazando el envío o la recepción de carpetas comprimidas con componentes ejecutables. Suban su carpeta comprimida (en formato **`.zip`**) a algún 'lugar' en la nube y envíen el hipervínculo al profesor y a su asistente mediante un mensaje de correo con el formato indicado. Deben mantener la carpeta viva hasta 4 de febrero del 2021.

Ponderación
Este proyecto tiene un valor del 15% de la calificación del curso.