

Fecha de entrega: 19 de junio del 2015, (24 horas)

OBJETIVOS DE LA TAREA

Realizar actividades de mantenimiento de software: específicamente mejoras (nuevas funcionalidades) al programa 3.

Investigación y utilización de técnica de backtracking (vuelta atrás) para buscar soluciones de problemas con restricciones.

Utilización de estructuras de datos nativas de Python tales como las secuencias y los diccionarios.

Utilización de TDA (Tipos de Datos Abstractos) basados en las estructuras de datos nativas.

Utilización de archivos.

Aplicar buenas prácticas de programación: documentación interna y externa del programa, reutilización de código, funciones, nombres significativos, eficiencia del programa, evaluar alternativas, etc.

Además se busca fomentar en el estudiante la investigación: podrían haber temas que necesita estudiar aparte del material tratado en el curso.

Requisitos para revisar el proyecto:

- **Deben estar terminadas las partes del programa 3 necesarias para probar las nuevas funcionalidades.**
- **Cumplir con todos los “ASPECTOS PARA EL DESARROLLO DEL TRABAJO” indicados al final de este documento.**

DEFINICIÓN DEL PROYECTO

Este programa consiste en agregar funcionalidades al programa 3 (pasatiempo aritmético KenKen). Seguimos trabajando con la actividad de mantenimiento de software. Para este caso específico el mantenimiento se refiere a mejoras ya que se están agregando nuevas funcionalidades.

REQUERIMIENTOS DEL PROGRAMA

MEJORAS (funcionalidades nuevas)

- 1- Los siguientes botones se agregarán a la opción de jugar:



Este botón se puede usar en cualquier momento que el juego haya iniciado: su función es que el programa automáticamente solucione el juego. Cualquier jugada hecha por el jugador se borrará para que el programa puedan ofrecer la solución final. El tiempo máximo permitido para dar la solución son 3 segundos. Los juegos que usen este botón no entran al Top 10. Si está en multinivel dicho nivel tampoco se usa para el Top 10. En caso de usar reloj este se detiene.

POSIBLES
JUGADAS

Este botón se puede usar en cualquier momento que el juego haya iniciado. Tiene dos funciones:

- a) Despliega en una sola ventana la lista con todas las jugadas posibles que se pueden hacer en la jaula donde se haya seleccionado una casilla. Por ejemplo para una jaula de dos casillas, operación 3-, dificultad 5x5, la lista sería:
1 4
4 1
2 5
5 2
- b) Opcionalmente el jugador puede indicar que una jugada seleccionada de la lista pase a la cuadrícula del juego. La colocación de la jugada en la jaula es de izquierda a derecha, de arriba a abajo.

ASPECTOS PARA EL DESARROLLO DEL TRABAJO

- El desarrollo es en el lenguaje Python versión 3.x, utilizando el material estudiado en el curso más los temas investigados.
- Para la interfaz con el usuario debe utilizar el modo gráfico (GUI).
- Puede hacer cambios a la interfaz gráfica pero cumpliendo con los requerimientos (funcionalidades) del programa.
- Debe enviar por medio del tecDigital (Evaluaciones, Programa 4), una carpeta comprimida (kenken_nombre_estudiante.rar o .zip) que contenga los siguientes archivos y todos aquellos elementos que sean necesarios para que el programa se pueda ejecutar:
 - Programa fuente (nombre: kenken.py).
 - Manual de usuario (nombre: kenken_manual_de_usuario.PDF).

Tome la documentación del programa 3 y complémntela con las nuevas funcionalidades. Explica paso a paso cómo usar cada una de las funcionalidades del programa. Apóyese en imágenes, capturas de pantallas, menús, diagramas y los aspectos que considere van a servir como una guía útil para que el usuario maneje el programa. Secciones mínimas de este manual:

 - Portada.
 - Contenido.
 - Guía de uso de las funciones del programa.
 - Sección de solución de problemas.
 - Documentación del trabajo (nombre: kenken_documentación.PDF).
 - Portada.
 - Contenido.
 - Descripción del enunciado del proyecto.
 - Algoritmo para el botón “SOLUCIONAR JUEGO”.
 - Explicación detallada de la técnica de backtracking.
 - Explicación detallada del algoritmo y las estructuras de datos usadas en el desarrollo de este botón.
 - Temas investigados (material no estudiado en el curso).

PROGRAMA 4: mejoras al pasatiempo aritmético KenKen (15%)

- Por cada uno de estos temas debe poner el marco teórico: de qué trata, cómo se usa, bibliografía.
 - Nuevas características usadas de interfaces gráficas de usuario (GUI) en Python.
 - Cualquier otro aspecto que el programador utilizó y no ha sido tratado en el curso.
- Conclusiones del trabajo realizado (incluya aprendizajes obtenidos).
- Estadística de tiempos: un cuadro que muestre el detalle de las actividades que realizó y las horas invertidas en cada una de ellas. La estadística permite medir el esfuerzo dedicado al trabajo en términos de actividades y tiempos, lo cual puede ser una base para calcular el esfuerzo requerido en futuros trabajos.

Ejemplos de actividades:

Actividad Realizada	Horas
Análisis de requerimientos	X
Diseño de algoritmos	X
Investigación de ...	X
Programación	X
Documentación interna	X
Pruebas	X
Elaboración del manual de usuario	X
Etc.	X
TOTAL	X

- Rúbrica de evaluación y análisis de resultados
 - Tome la rúbrica de evaluación y por cada concepto calificado indique el tipo de avance (T/P/N) y el análisis de resultados
 - Avance (T)otalmente desarrollado. No hace falta análisis excepto que requiera hacer alguna observación.
 - Avance (P)arcialmente desarrollado. En el análisis indicar: ¿qué hace?, ¿qué falta?
 - Avance (N)o desarrollado. En el análisis indicar el motivo.
 - Partes que desarrolló adicionales a los requerimientos.

PROGRAMA 4: mejoras al pasatiempo aritmético KenKen (15%)

Concepto	Puntos	Puntos obte- nidos	Avan- ce T/P/N	Análisis de resultados
Solucionar juego. No se aceptan soluciones que tarden más de 3 segundos. No ingresan al Top 10. Reloj se detiene.	50			
Posibles jugadas: desplegar	20			
Posibles jugadas: seleccionar	15			
Ayuda	1			
Documentación interna	2			
Manual de Usuario	5			
Documentación del trabajo	7			
TOTAL	100			
Funcionalidades desarrolladas adicionalmente				

IMPORTANTE: CONOCIMIENTO DE LA SOLUCIÓN PRESENTADA. En la revisión del trabajo, el estudiante debe demostrar un completo dominio de la solución que implementó, tanto desde el punto de vista técnico (uso de Python) como de la funcionalidad del programa. La revisión se puede hacer individualmente o en grupos, examinando el programa o temas aplicados en el programa.