

Colocador de fichas

Reporte de Auditoría

Arburola León, Samantha Patricia
Murillo Vargas, Jose Pablo
Villalobos Rodríguez, Roger Andrés

Abril 2018

Colocador de fichas	1
Resumen de la anomalías de inspección	3
a. Funciones	3
b. Prácticas Obligatorias	4
Número de errores reportados: 20	5
Calificación: 6	5
c. Estándar GNU	6
Número de errores reportados: 16	6
d. Aspectos sobre los documentos	7
i. Informe escrito sobre el proyecto	7
ii. Archivo readme.txt	8
e. Resultados de Cppcheck	8
f. Resultados de Splint	9
Una estimación del esfuerzo de retrabajo	11

A. Resumen de la anomalías de inspección

a. Funciones

Lista de chequeo presencia de funciones o rutinas que pueden ayudar a generar una solución para la especificación del cliente y cumpla con el requerimiento funcional

Requerimiento funcional	Cumple		Observaciones
	Si	No	
Recibir parámetro de archivo de prueba desde la terminal		X	La función main() no recibe parámetros por lo cual no está recibiendo el archivo por línea de comandos.
Abrir y validar existencia de archivo de prueba	X		El nombre del archivo a leer está alambrado dentro del código.
Validar datos dentro de archivo de prueba		X	No hay validación en cuanto al tipo datos dentro del archivo (ver si es un número), no verifica si ya hay una ficha en la misma posición.
Guardar datos de entrada especificados dentro de una estructura de datos	X		Los datos se guardan en una matriz donde: 1 es que no hay una ficha 0 es donde hay una ficha
Realizar acomodo óptimo de fichas	X		Se realiza una inspección donde se determina cual es la mejor columna y fila para poner las fichas y se va guardando la cantidad de movimientos
Mostrar resultado con las fichas acomodadas y el número de movimientos realizados		X	Solo se muestra la cantidad menor de movimientos realizados para el acomodo de fichas.

Número de funciones presentes sobre el total mencionado: 3/6

Requerimiento	Cumple		Observaciones
	Si	No	
No usar variables globales	X		
Definición y uso de funciones	X		1. Funciones con más de una tarea. 2. Mejorar el nombre de las funciones
Documentación interna para cada función		X	1. Hay código comentado. 2. Hay comentarios redundantes respecto a la variable. 3. Comentarios sobre la función mal ubicados 4. No hay consistencia en cuanto a documentación interna de las funciones

Número de requerimientos adicionales cumplidos: 2/3

Total de requerimientos cumplidos: 5/9

Métrica de completitud de sistema: 55.56

b. Prácticas Obligatorias

Resultados sobre Prácticas obligatorias sugeridas por Tecnowired.com

Práctica	Cumple		Observaciones
	Si	No	
Indentación dentro de funciones, registros y estructuras de control	X		Se hace un muy buen uso de indentación en el código fuente, hace falta en la instrucciones dentro de la estructura de control "do" (32-36)

Uso de llaves en toda estructura de control	X		Se hace un uso consistente en cuanto al posicionamiento de llaves
Escribir sólo una instrucción por línea	X		
Incluir un espacio después de las comas		X	Solo la línea 105, 173, 181, 182, 218, 226, 227 no cumple con esta práctica
Separar los operadores binarios con espacios a ambos lados		X	Las líneas: 57, 77, 78, 85, 103, 104, 114, no tienen separados los operadores
Las variables de índices dan comienzo con i	X		
Uso de formato CamelCase para variables con nombre compuesto		X	Las variables definidas en las líneas 70, 160, 161, 205 y 206 no cumplen con la práctica
Uso de #define para variables constantes		X	No se usó variables constantes
Cada bloque de código comienza con un bloque de comentario donde explica el propósito, parámetros de entrada, posibles valores de salida,		X	Las funciones validarMatriz(5) cargaMatriz(16), getdif(127), no tienen comentarios iniciales que cumplan con la práctica especificada Ninguna función explica los valores de entrada

Número de errores reportados: 20

Calificación: 6

c. Estándar GNU

Resultados sobre estilo GNU de Richard Stallman

Práctica	Cumple		Observaciones
	Si	No	
Usar, a lo sumo, 79 caracteres por línea de código		X	1. Línea 32: 95 2. Línea 62: 145 3. Línea 83: 83 4. Línea 125: 103 5. Línea 181: 87 6. Línea 196: 106 7. Línea 226: 82 8. Línea 241: 107 9. Línea 244: 114
Poner el corchete del cuerpo de una función en la primera columna debajo de la declaración de este		X	No cumple en las funciones: main, transformarArreglo, cargaMatriz y validarMatriz
Comenzar los nombres de funciones a declarar en la columna uno		X	No cumple para ninguna de las funciones
Un comentario inicial en el archivo que explica su propósito		X	No se encuentra
Incluir comentarios en cada función de qué hace, que parámetros recibe y qué valores puede retornar		X	No se encuentra
Declaración explícita del tipado de todos los objetos	X		
Uso de minúsculas para la declaración de variables	X		
Uso de guión bajo para nombres compuestos		X	Solo se encuentra en: <ul style="list-style-type: none"> - getmin_c - col_resultado - fil_resultado - getmin_f

Número de errores reportados: 16

Calificación: 7

Métrica de cumplimiento con estándares de codificación: 6.5

d. Aspectos sobre los documentos

i. Informe escrito sobre el proyecto

Documentación	Cumple		Observaciones
	Si	No	
Revisión de estrategia planteada ante lo codificado en el archivo fuente	0.5		
Documentación o referencia en cuanto a uso particular de funciones o algoritmos	1		
Explicar qué uso de paso de parámetros usaron	1		
Evidencia de pruebas realizadas		0	solo se evidencia 1 prueba
En caso de existir, justificación de aspectos de la especificación que no fueron desarrollados			No aplica
Bibliografía	0.5		Recordar traducir a español cuando se utiliza una herramienta para crear la bibliografía

Total de Puntos: 3

Calificación: 60

ii. Archivo readme.txt

Documentación	Cumple		Observaciones
	Si	No	
Descripción de ambiente de desarrollo utilizado	1		Acertado
Descripción de cómo correr el programa	1		Se puede mejorar

Nota: Se recomienda indicar en el readme.txt el nombre del programa y su objetivo, así como no solo poner los nombres sino escribir un membrete incluyendo universidad, proyecto y curso.

Calificación: 100

Métrica de calidad de documentación: 80

e. Resultados de Cppcheck

```
joses-MBP:~ josemurillo$ cppcheck /Users/josemurillo/Downloads/Grupo07-Pablo-Gerardo-Kevin/Proyecto 01 Impe
olgrupo07.c:59]: (error) Resource leak: matriz
joses-MBP:~ josemurillo$
```

La variable “matriz” que fue usado como puntero para el archivo a leer no fue cerrado correctamente con la instrucción fclose().

Total de errores reportados: 1

Calificación: 99

f. Resultados de Splint

```
Possibly null storage matriz passed as non-null param:
validarMatriz (matriz)
A possibly null pointer is passed as a parameter corresponding to a formal
parameter with no /*@null@*/ annotation. If NULL may be used for this
parameter, add a /*@null@*/ annotation to the function parameter declaration.
(Use -nullpass to inhibit warning)
Downloads/Grupo07-Pablo-Gerardo-Kevin/Proyecto 01 Imperativo Grupo 07 -
Pablo - Gerardo - Kevin/proyecto1grupo07.c:27:14:
Storage matriz may become null
Downloads/Grupo07-Pablo-Gerardo-Kevin/Proyecto 01 Imperativo Grupo 07 - Pablo -
Gerardo - Kevin/proyecto1grupo07.c:28:5: Return value (type int) ignored:
validarMatriz(ma...
Result returned by function call is not used. If this is intended, can cast
result to (void) to eliminate message. (Use -retvalint to inhibit warning)
Downloads/Grupo07-Pablo-Gerardo-Kevin/Proyecto 01 Imperativo Grupo 07 - Pablo -
Gerardo - Kevin/proyecto1grupo07.c:32:33:
Format argument 1 to fscanf (#[\n] expects char * gets char [80] *:
&lectura
Type of parameter is not consistent with corresponding code in format string.
(Use -formattype to inhibit warning)
Downloads/Grupo07-Pablo-Gerardo-Kevin/Proyecto 01 Imperativo Grupo 07 -
Pablo - Gerardo - Kevin/proyecto1grupo07.c:32:26: Corresponding format code
Downloads/Grupo07-Pablo-Gerardo-Kevin/Proyecto 01 Imperativo Grupo 07 - Pablo -
Gerardo - Kevin/proyecto1grupo07.c:32:5: Return value (type int) ignored:
fscanf(matriz, "...
Downloads/Grupo07-Pablo-Gerardo-Kevin/Proyecto 01 Imperativo Grupo 07 - Pablo -
Gerardo - Kevin/proyecto1grupo07.c:34:5: Assignment of size_t to int:
largoLinea = strlen(lectura)
To allow arbitrary integral types to match any integral type, use
+matchanyintegral.
Downloads/Grupo07-Pablo-Gerardo-Kevin/Proyecto 01 Imperativo Grupo 07 - Pablo -
Gerardo - Kevin/proyecto1grupo07.c:52:9: Unrecognized identifier:
transformarArreglo
Identifier used in code has not been declared. (Use -unrecog to inhibit
warning)
Downloads/Grupo07-Pablo-Gerardo-Kevin/Proyecto 01 Imperativo Grupo 07 - Pablo -
Gerardo - Kevin/proyecto1grupo07.c:52:9:
Variable linea used before definition
An rvalue is used that may not be initialized to a value on some execution

Downloads/Grupo07-Pablo-Gerardo-Kevin/Proyecto 01 Imperativo Grupo 07 - Pablo -
Gerardo - Kevin/proyecto1grupo07.c:57:13:
Variable dimension used before definition
Downloads/Grupo07-Pablo-Gerardo-Kevin/Proyecto 01 Imperativo Grupo 07 - Pablo -
Gerardo - Kevin/proyecto1grupo07.c:23:9:
Variable llave declared but not used
A variable is declared but never used. Use /*@unused@*/ in front of
declaration to suppress message. (Use -varuse to inhibit warning)
Downloads/Grupo07-Pablo-Gerardo-Kevin/Proyecto 01 Imperativo Grupo 07 - Pablo - Gerardo - Kevin/proyecto1grupo07.c: (in function transformarArreglo)
Downloads/Grupo07-Pablo-Gerardo-Kevin/Proyecto 01 Imperativo Grupo 07 - Pablo -
Gerardo - Kevin/proyecto1grupo07.c:74:5: Unrecognized identifier: itoa
Downloads/Grupo07-Pablo-Gerardo-Kevin/Proyecto 01 Imperativo Grupo 07 - Pablo -
Gerardo - Kevin/proyecto1grupo07.c:77:13:
```

El error de la línea 28 presenta que, aunque se hace una validación de abrir el archivo, se ignora el resultado de este y se sigue ejecutando el resto de instrucciones que se encargan de leer desde el archivo, a pesar de que este puede estar vacío.

El error que se muestra sobre que la función “transformarArreglo” es usado pero no declarado se debe a porque la declaración de la función se encuentra después en el código.

El posible error presente en la línea 34 reportado se debe a que `strlen()` retorna el tipo de datos `size_t` y se asigna a una variable tipo `int`. No se realizó el casting de `(int)` antes de retornar el resultado de `strlen()`.

El error reportado sobre las variables “línea” y “dimensión” de ser usadas antes de su definición se debe a que, primero, no están inicializadas en su declaración. La otra causa se debe a que se espera que ya para cuando se realiza la segunda rutina de “if”, esas variables deben haber sido asignadas si o si, no se hace validación sobre el contenido de estas variables antes de llamar la función “transformarArreglo” con esas variables como parámetros.

Se reporta que la variable “llave”, declarada en la línea 23, nunca fue usada.

Se reporta el error de que la función “itoa” no fue declarada dentro del archivo, esto se debe a que no se incluyó el archivo “`stdlib.h`”

Total de errores reportados: 7

Calificación: 93

Métrica de ausencia o carencia de errores potenciales: 96

Una estimación del esfuerzo de retrabajo

Debido a la cantidad de errores y realizando cálculos consensuados sobre la duración de la corrección de cada parte errónea del trabajo se demorara lo siguiente:

Observaciones por resolver	Duración
Recibir parámetro de archivo de prueba desde la terminal	2 horas
Validación de datos dentro del archivo de prueba	2 horas
Mostrar resultado con las fichas acomodadas	6 horas
Arreglo de la documentación interna	1 hora
Arreglar la indentación	5 minutos
Arreglar los espacios después de las comas y en operaciones binarias	30 minutos
Corrección de nombres de variables compuestas	10 minutos
Corrección del tamaño máximo por línea	10 minutos
Arreglar el posicionamiento de los corchetes	5 minutos
Revisión de estrategia planteada ante lo codificado en el archivo fuente	1 hora
Evidencia de pruebas realizadas	1 hora
Bibliografía	10 minutos
Tiempo total de retrabajo	14 horas con 10 minutos