Fecha: 27 de junio 2020

Ciclo:

ICC02\_Desarrollo de aplicaciones web II

ICB0\_Desarrollo de aplicaciones multiplataforma

ICA0\_Administración de sistemas informáticos en red

Proyecto: ICX0\_P4 Programa comandos personalizados para el sistema operativo

Nombre y apellidos:

Prueba de validación (Junio 2020)

### Instrucciones para el estudiante

* La duración de la prueba es de **30 minutos** (incluye también subir la prueba al espacio reservado para ello).
* El valor numérico de cada pregunta se indica en cada una de ellas.
* La nota mínima para superar la prueba es de 5 sobre 10.
* El peso de esta prueba en relación a la nota final del proyecto o seminario es del 10%.
* Las preguntas deberán responderse en el espacio recomendado.
* ¿Está permitido consultar o utilizar algún material/recurso durante la prueba?
  + No
* Para la entrega de la prueba deberás:
  + Hacerlo por el canal indicado en el aula virtual.
  + Formato: .doc
  + Nombre del archivo: (NombrePrueba)\_Apellido1Apellido2\_Nombre

### Enunciados

**Pregunta 1.** Según lo documentado en el producto 1, qué herramienta usarías para detectar un error de programación que detectas que aparece en un bucle cuando la variable contador “i” que se incrementa desde 0 y con paso 1 tiene un valor mayor a 100? (**2,5 puntos**) **(resolver en un máximo de 10 líneas aprox.)**

Teniendo en cuenta que uso Visual Studio 2019, utilizaría la herramienta de debuggación, que consiste en hacer la reproducción del código paso a paso (f10 o f11), para ello, arrancaría el código desde el cursor o el puntero donde lo hayamos situado, para ver ese trozo de código o desde el principio, y en el paso que me interesa iría, reproduciendo y viendo si la variable i, va cogiendo el valor que le corresponde o incrementando 1 realmente. También podemos usar los puntos de interrupción para ver que valor adquieren las variables y si vamos por buen camino.

**Pregunta 2.** Según lo tratado en los productos 2 y 3, muestra la línea de código programado en C que permite lanzar a una aplicación un ping y redirigir la salida estándar de este hacia un archivo de texto situado en la carpeta “D:/redirección” de nombre “comandoredirección.txt”. Solamente se te pide que muestres una línea de código. (**2,5 puntos**) **(resolver en una línea)**

system (“ping 192.168.1.1 > D:/redireccion/comandoredireccion.txt”);

**Pregunta 3.** Según lo tratado en los productos 2 y 3, si suponemos que en el archivo “archivos.txt” que se encuentra en “d:/archivos”, tenemos la redirección de la salida estándar del comando “dir d:\data”. Muestra el código de no más de 5 lineas que te permite analizar si en el directorio “data” se encuentra un archivo llamado “alumnos.txt”. Se pide que muestres únicamente como la aplicación abre y analiza el código. (**2,5 puntos**) **(resolver en un máximo de 10 líneas aprox.)**

archivo = fopen("D:/archivos/archivos.txt", "r");

while (!feof(archivo)){

fgets(linea, 100, archivo)

if(strstr(linea, "alumnos.txt"){Printf("El archivo alumnos.txt está en este directorio\n");

}else{Printf("El archivo no está en el directorio!\n"}}

\*en el strstr he puesto "alumnos.txt" pero lo más recomendable es guardarlo en una variable y comparar variables.

**Pregunta 4.** ¿Qué funciones de la librería string.h has utilizado en los productos 2 y 3, y con qué propósito? (**2,5 puntos**) **(resolver en un máximo de 10 líneas aprox.)**

Variable = string, ya que son de la biblioteca string.

Strcat(variable, variable o string) para concatenar Strings.

Strcpy(varible, variable o string) para copiar una string en una variable.

Strstr(variable1, variable2) para comprobar si variable2 se encuentra en variable1.

Strtok(variable1, variable2) para asignar a otra variable, que el valor de ese string (variable1) llega hasta encontrar variable2, por ejemplo “Hola pepito” hasta encontrar un espacio, variable2= “ “.