Logotipo, nombre de la empresa

Descripción generada automáticamente

**EVALUACIÓN PARCIAL – RESOLUCIÓN DE CASOS**

ALUMNO:

* Di Salvio Tiago
  + 42727210
  + [tiago.disalvio@alumnos.uai.edu.ar](mailto:tiago.disalvio@alumnos.uai.edu.ar)

CARRERA: Ingeniería en Sistemas Informáticos

ASIGNATURA: Programación Estructurada

FECHA: 16/07/2021

***1) ¿Qué tipos de ciclos de repetición conoce? Explíquelos y clasifíquelos.***

Ciclo exacto (FOR); Posee internamente un “contador” que incrementa la cantidad de veces que se ejecutó el programa con un condicional que al llegar al número deseado me saque del ciclo.

Chart

Description automatically generated

Ciclo inexacto (WHILE-DO); Se ingresa un dato, se controla el valor, si cumple los requisitos, se ingresa al programa, una vez terminado, se ingresa un nuevo dato y se repite el proceso. Cuando este no se cumpla, finaliza el proceso.

Gráfico, Diagrama

Descripción generada automáticamente

Ciclo inexacto (DO-WHILE); Es similar al (WHILE-DO) aunque la revisión se efectúa cuando es ejecutado al menos una vez el proceso.

Gráfico

Descripción generada automáticamente

***2) ¿Qué es un vector? ¿Cómo se declara?***

Un vector o array es una estructura homogénea, compuesta por varios elementos, todos del mismo tipo y almacenados consecutivamente en memoria. Cada elemento puede ser accedido directamente por el nombre de la variable “array” seguido de uno o más subíndices encerrados entre corchetes.

La declaración de un array especifica el nombre del array, el número de elementos y el tipo. Según su dimensión, cabe distinguir entre arrays unidimensionales y multidimensionales.

Unidimensionales:

Mayormente se utilizan ciclos FOR aprovechando su índice con el índice del array.

Cada ciclo permite acceder a una posición diferente dentro del array.

Texto

Descripción generada automáticamente

Multidimensionales:

Mayormente se utilizan ciclos FOR aprovechando su índice con el índice del array.

Necesita de un ciclo por cada dimensión que posea.

Texto

Descripción generada automáticamente

***3) ¿Cuál es la diferencia entre búsqueda secuencial y búsqueda binaria? Explique y de un ejemplo.***

El método de **búsqueda secuencial** (valido), es el menos eficiente.

Se basa en comparar el valor que se desea buscar con cada uno de los valores array. (no tiene por qué estar clasificado).

El seudocódigo para este método de búsqueda puede ser el siguiente:

***<función Búsqueda\_S (array a, valor que queremos buscar)>***

***i = 0***

***DO WHILE (no encontrado)***

***IF (valor = a[i])***

***encontrado***

***ENDIF***

***i = i+1***

***ENDDO***

***END <Búsqueda\_S>***

Un método eficiente de búsqueda, que puede aplicarse a los arrays clasificados, es la **búsqueda binaria.**

Consiste en lo siguiente: Selecciona el elemento del centro del array. Si el valor a buscar no coincide con el elemento seleccionado y es mayor a él, continua la búsqueda en la segunda mitad del array. Si, por el contrario, el valor a buscar es menor que el valor del elemento seleccionado, la búsqueda continua en la primera mitad del array.

En ambos casos, se halla el elemento central, repitiéndose así el ciclo

Este proceso se repite hasta que se encuentre el valor a buscar, o bien hasta que el intervalo de búsqueda sea nulo.

El seudocódigo para este algoritmo puede ser el siguiente:

***<función Búsqueda\_B (array a, valor que queremos buscar)>***

***DO WHILE (exista un intervalo donde buscar)***

***x = elemento mitad del intervalo de búsqueda***

***IF (valor = x) THEN***

***encontrado***

***ELSE***

***IF (valor > x) THEN***

***buscar “valor” en la segunda mitad del intervalo de búsqueda***

***ENDIF***

***IF (valor < x) THEN***

***buscar “valor” en la primera mitad del intervalo de búsqueda***

***ENDIF***

***ENDIF***

***ENDDO***

***END <Búsqueda\_B>***

***4) ¿Qué es una función? Compárela con un procedimiento.***

Una función es una colección de sentencias que ejecutan una tarea específica. En un programa en C se distinguen dos clases de funciones, definidas por el usuario y de bibliografía.

Los procedimientos solo procesan datos, pero no devuelven ningún resultado. Tanto las funciones como los procedimientos deben ser declarados primero, para luego ser definidos.

La petición corresponde al “llamado” de la función para hacer una tarea en particular.

***5) ¿Qué es un parámetro? ¿Qué tipos de parámetros hay?***

Es una variable que puede pasar su valor a una subrutina. Dichas subrutinas poseen una firma, la cual indica si espera o no un parámetro.

Los argumentos son los valores que se transitan a una subrutina, deben ser compatibles con los tipos de parámetros de la definición de esta.

El lenguaje en C solo posee pasaje de parámetros por copia o por valor. Por ende, no es posible cambiar el valor de un parámetro desde una función.

Existen dos tipos de parámetros;

Por referencia: Dentro de un parámetro en una función indica una referencia a la memoria, es decir un puntero. Que suele utilizar cuando nosotros tenemos una función que devuelve o necesita devolver mas de un valor o bien cuando tenemos una función que por su estructura no puede devolver ningún valor, pero nosotros necesitamos acceder a algún resultado.

Texto

Descripción generada automáticamente

Por valor: Se efectúa una copia del valor de la variable y se pasa como argumento.

Podrán ser modificados, pero cuando la función devuelve el control a la función solicitada, recuperan los valores originales que tenían antes de ser modificados por la función.

***6) Explique el algoritmo de ordenamiento por burbujeo.***

Hay muchas formas de clasificar datos y una de las más conocidas es la clasificación por el método de la burbuja.

Para ordenar una lista de menor a mayor, partiendo de que los datos a ordenar están en una lista de *n* elementos:

1. Comparamos el primer elemento con el segundo, el segundo con el tercero, etc.

Cuando el resultado de una comparación sea “mayor que” intercambian, los valores de los elementos comparados. Con esto conseguimos llevar el mayor a la posición *n*.

1. Repetimos el punto a, los n-1 primeros elementos de la lista. Con esto conseguimos llevar el valor mayor a la posición *n*.
2. Repetimos el punto a, los n-2 primeros elementos de la lista y así sucesivamente.
3. El proceso termina al finalizar de repetir el proceso descrito, n-1 veces, o cuando al finalizar la ejecución del *iesimo* proceso no haya habido ningún cambio.

El seudocódigo para este algoritmo puede ser el siguiente:

***< función clasificar (array “a” de “n” elementos) >***

***[“a” es un array cuyos elementos son a0, a1, …, an-1]***

***n = n-1***

***DO WHILE (“a” no esté ordenado y n > 0)***

***i = 1***

***DO WHILE (i <= n)***

***IF (a[i-1] > a[i]) THEN***

***permutar a [i-1] con a[i]***

***ENDIF***

***i = i + 1***

***ENDDO***

***n = n-1***

***ENDDO***

***END < clasificar >***