

Proyecto 5

Algoritmos y Estructuras de Datos I Laboratorio

8 de junio de 2012

El objetivo del proyecto es desarrollar programas en lenguaje C en base al formalismo visto en el teórico de la materia. La idea general es derivar o demostrar los programas y traducirlos al lenguaje C, agregando las partes correspondientes a la entrada/salida con las herramientas que nos brinda este último lenguaje. Para hacerlo hay que tener en cuenta:

- Antes de escribir el programa en lenguaje C terminar el ejercicio correspondiente donde se demuestra o deriva el programa escrito en el lenguaje del teórico a partir de su especificación.

Se debe presentar la derivación o demostración del algoritmo con su resultado final separado, de modo que coincida con el programa escrito en C.

- Los programas deben tomar los datos de entrada del usuario y mostrar los resultados en pantalla.
- Recordar que en la entrada se debe chequear que los valores ingresado cumplan la precondición de la especificación del programa.
- Los programas deben ser compilados con las opciones `-pedantic -Wall -Werror -ansi` del compilador gcc. Al hacerlo no debe aparecer ningún mensaje de error o de alerta.
- No usar ninguna variable global.
- Agregar sentencias `assert` con las anotaciones de las derivaciones donde crea necesario para encontrar errores en los programas.

Ejercicios

1. Hacer el ejercicio 4.a del práctico 4 (menor entero x tal que $x^3 + x \geq N$) y traducirlo al lenguaje C.
2. Hacer el ejercicio 5 del práctico 4 (suma de los valores en un arreglo) y traducirlo al lenguaje C.
3. Hacer el ejercicio 7 del práctico 4 (máximo de los elementos de un arreglo) y traducirlo al lenguaje C. Realizar las modificaciones que se piden en aquel ejercicio para detectar el caso en que no se cumpla la precondición y devolver un mensaje de error.

Nota: Consulte con los docentes del laboratorio sobre funciones de librería en C para mostrar mensajes de error y utilice alguna.

4. Hacer el ejercicio 8 del práctico 4 (cantidad de elementos pares en un arreglo) y traducirlo al lenguaje C **dentro de una función**.

Nota: *La función debe ser llamada dentro de `main` después de haber leído la entrada del algoritmo.*

5. Hacer el ejercicio 10 del práctico 4 (posición de un elemento en un arreglo) y traducirlo al lenguaje C **dentro de una función**. Si el elemento no se encuentra mostrar un mensaje de alerta y devolver normalmente el resultado indicado.

Nota: *Consulte con los docentes del laboratorio sobre funciones de librería en C para mostrar mensajes de alerta y utilice alguna.*

6. Hacer el ejercicio 9 del práctico 4 (cálculo de la desviación estandar) y traducirlo al lenguaje C **dentro de una función**.

7. Hacer el ejercicio 6.a (existencia de un elemento en un arreglo) agregando las modificaciones que se recomiendan en el ejercicio 11 del práctico 4 y traducirlo al lenguaje C **dentro de una función**.

8. Hacer el ejercicio 1 del práctico 4 (algoritmo de la división) y traducirlo al lenguaje C **dentro de una función**.

Nota: *Consulte con los docentes del práctico sobre alternativas para devolver más de un valor desde una función.*

9. Derive y programe en C el programa que calcula fibonacci (ejercicio 14 del práctico 4). La entrada y la salida del algoritmo deben estar programadas en funciones separadas.

Punto *: Existe alguna implementación más eficiente de fibonacci ($\mathcal{O}(\log n)$)? (sin usar funciones de números reales). Si la hay, impleméntela.

Punto *: Discutir e implementar alguna forma de comparar el tiempo de ejecución de este programa con el anterior.

10. Hacer el ejercicio 15 del práctico 4 (mayor diferencia) y traducirlo al lenguaje C. La entrada y la salida del algoritmo deben estar programadas en funciones separadas.

11. Hacer el ejercicio 17 del práctico 4 (cantidad de productos positivos) y traducirlo al lenguaje C. La entrada y la salida del algoritmo deben estar programadas en funciones separadas.

12. Hacer el ejercicio 22 del práctico 4 (producto vectorial) y traducirlo al lenguaje C.