

Algoritmos y Estructuras de Datos I - Laboratorio

Proyecto 5

Programación imperativa en C

1. Objetivo

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.
- Se debe chequear que los valores ingresados cumplan la precondition de la especificación del programa.
- Los programas deben ser compilados con las opciones `-Wall -Werror -pedantic` del compilador gcc.
- Agregar sentencias `assert` con las anotaciones de las derivaciones donde crea necesario para encontrar errores en los programas.

2. Ejercicios

Los ejercicios mencionados son del práctico 4.

1. Hacer el ejercicio 4.a (menor entero x tal que $x^3 + x \geq N$) y traducirlo al lenguaje C.
2. Hacer el ejercicio 5 (suma de los valores en un arreglo) y traducirlo al lenguaje C.
3. Hacer el ejercicio 7 (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 precondition y devolver un mensaje de error.
4. Hacer el ejercicio 8 (cantidad de elementos pares en un arreglo) y traducirlo al lenguaje C **dentro de una función**.
5. Hacer el ejercicio 9 (cálculo de la desviación estandar) y traducirlo al lenguaje C **dentro de una función**.
6. Hacer el ejercicio 10 (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.
7. Hacer el ejercicio 11.a (todos elementos positivos) y traducirlo al lenguaje C **dentro de una función**.

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

Nota: *Usar un tipo estructurado (struct) para devolver mas de un valor desde una función.*

9. **(Punto ★)** Hacer los ejercicios siguientes y traducirlos al lenguaje C. En cada caso, la entrada y la salida del algoritmo deben estar programadas en funciones separadas.

- a) Ejercicio 14 (Fibonacci)
- b) Ejercicio 15 (mayor diferencia)
- c) Ejercicio 17 (cantidad de productos positivos)
- d) Ejercicio 22 (producto escalar)