CONTROL 1

INFO 088 – Taller de Estructura de Datos y Algoritmos

Académicos: Erick Araya, Héctor Ferrada.

Julio 28, 2020

Ejecución: **./problema *n***

En el fuente está declarada la siguiente estructura:

```
struct Datos{
        int n;   // largo de los arreglos A y C
        int *A;  // arreglo de números aleatorios en [0...MAX]
        int *C;  // arreglo que contiene los largos de los números en A
};
```

En el main(), lea **n** desde los argumentos del programa y cree la estructura **S** de tipo **Datos**. Copie el valor **n** leído en **S**, y cree los arreglos **A** y **C** de largos **n**, luego invoque a los siguientes métodos:

- ***(2 Punto) void llenaEstructura(Datos S)***. Llene **A** con enteros aleatorios en el rango [0...MAX] (MAX está declarado en la cabecera del programa). Llene también **C**. Este arreglo almacena la cantidad de dígitos que tienen los valores de **A**; por tanto, calcule cuántos dígitos tiene cada **A[i]**, **0<=i<n,** y almacene este valor en **C[i]**.

- ***(2 Puntos) void creaLista(Datos S, nodo **L)***. En base a la estructura nodo vista en clases, se pide crear una lista simplemente enlazada **L** (el struct está en el fuente) con los datos que tiene **S** de la siguiente manera:

  Para cada valor **A[i]**, **0<=i<n**, agregue a **L** cada uno de los **C[i]** dígitos que **A[i]** posee. Los dígitos que almacene **L** deben quedar en el mismo orden, de izquierda a derecha, que aparecen en **A**. Por ejemplo, para **n=7**, suponga que **S** tiene los siguientes datos:

  S = (A[i], C[i]) = (**9383**, 4) (**886**, 3) (**2777**, 4) (**6915**, 4) (**7793**, 4) (**8335**, 4) (**5386**, 4)

  Entonces, L = **9 3 8 3 8 8 6 2 7 7 7 6 9 1 5 7 7 9 3 8 3 3 5 5 3 8 6**

- ***(2 Puntos) void datosRecursivos(Datos S, nodo **P).*** Crear una nueva lista simplemente enlazada **P** cuyos datos son 1 o 0, de acuerdo al siguiente criterio: Si **A[i]** es primo, el nodo de la lista enlazada tendrá el valor **1**; si **A[i]** no es primo, el nodo de la lista enlazada tendrá el valor **0**. Los dígitos **0** o **1** que almacene **P** deben quedar en el mismo orden de los datos de **A[i]**. Para determinar si **A[i]** es primo, deberá crear una función ***booleana RECURSIVA esPrimo(int num, otros parámetro que necesite)*** que entregará **true** si **A[i]** es primo y **false** si no lo es, la cual debe invocar desde ***datosRecursivos(...)*** . Para el ejemplo anterior, con **n = 7**, la lista **P** será:

  P = **0 0 1 0 1 0 0**

En el fuente dispone de los métodos: appendToListL(), appendToListR(), imprimeEstructura() y printList(), los cuales puede utilizar para para desarrollar lo pedido y visualizar el contenido de sus estructuras de datos.