

UTN – FRBA – Algoritmos y Estructura de Datos – Examen Final – 15/07/2016

Apellido y nombre: _____ Legajo: _____ Cursó con Prof: _____
Cantidad de hojas entregadas: _____ Nota: _____ Evaluó Prof: _____

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, ó C++, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros.

Elecciones Consejeros Universitarios – Consejo Departamental

Temas evaluados: Abstracción, estructura de datos contiguas.

Definición del problema

La UTN.BA debe evaluar los votos de las elecciones de las autoridades de los consejos. Se deben elegir representantes de los distintos claustros: Docentes, Graduados y Estudiantes. Usted es parte del equipo que desarrolla la aplicación de designación de cargos para el departamento de Sistemas de la Facultad..

Dinámica de la asignación de cargos

Se utiliza el sistema d'Hondt que es un método de promedio mayor para asignar cargos en sistemas de representación proporcional. Se caracteriza por dividir los totales de los votos obtenidos por cada lista (desde uno a la cantidad de cargos. 1 a 5 en este caso), produciéndose secuencias de cocientes decrecientes para cada una. La asignación se hace a los promedios más altos.

Ejemplo: El claustro docente otorga 5 cargos de consejeros departamentales

	Lista 1 (50 Votos)	Lista 2 (35 votos)	Lista 3 (15 Votos)
1	50.0 – 1ro.	35.0 – 2do.	15.0
2	25.0 – 3ro.	17.5 – 4to.	7.5
3	16.6 – 5to.	11.6	5.0
4	12.5	8.7	3.7
5	10.0	7.0	3.0

La lista Nro. 1 obtiene la designación de tres consejeros, la lista Nro. 2 dos y la lista Nro. 3 ninguno, el orden de los docentes es el que se indica en la tabla

Problema

Los consejeros titulares que se deben designar son: 5 consejeros del claustro docente, dos de graduados y tres estudiantes, para ello cuenta con:

1. Un array Docentes con los nombres de los candidatos y de la lista de 15 posiciones, agrupados por lista (5 candidatos de cada una de las tres listas participantes) en orden decreciente de cantidad de votos obtenidos por lista. Tres variables globales con los votos obtenidos por cada lista (Votos1, Votos2, Votos3).
2. Un array Graduados con los nombres de los candidatos de 4 posiciones, agrupados por lista (2 candidatos de cada una de las dos listas participantes) en orden decreciente de cantidad de votos obtenidos por lista. Dos variables globales con los votos obtenidos por cada lista. (no es de aplicación para este momento)
3. Un array Estudiantes con los nombres de los candidatos de 6 posiciones, agrupados por lista (3 candidatos de cada una de las dos listas participantes) en orden decreciente de cantidad de votos obtenidos por lista. Dos variables globales con los votos obtenidos por cada lista (no es de aplicación para este momento)

Dispone

Funcion OrdenarVector(TipoDato Nonbre[], int tamaño). Si la utiliza debe mencionar criterio de ordenamiento.

Se pide

1. Defina y declare todas estructuras de datos que considere necesarias para la resolución del problema
2. Declare el prototipo de la función GenerarListaConsejerosDocentes, que con los datos disponibles permita generar un array con los 5 consejeros electos, determinando posición, nombre y lista que representa.
3. Codifique o diagrame la función GenerarListaConsejerosDocentes descrita en el punto anterior

Disponiendo de la lista de candidatos y los votos totales obtenidos lo mas simple es cargar los datos en un vector de 15 posiciones con el nombre del candidato, la lista que representa y un campo para calcular por cada candidato, y en orden el total de votos dividido por números sucesivos desde 1 hasta la cantidad de cargos, 5 en este caso, luego ordenar el vector por este campo y seleccionar los 5 mejores.

Punto 1

```
struct RegistroDato{
    char Candidato[20 + 1]; //supongo ambos campos con un máximo de 20 caracteres
    char Lista[20 + 1]
};
struct RegistroResultado{
    char Candidato[20 + 1];
    char Lista[20 + 1];
    int Posicion
};
```

Punto 2

La función recibe el vector de 15 posiciones con los datos y devuelve el de 5 posiciones con los resultados. Los votos son variables globales como indica el enunciado.

```
void GenerarListaConsejerosDocentes(RegistroDato VectorDatos[], RegistroResultado VectorResultado[])
```

Punto 3

```
void GenerarListaConsejerosDocentes(RegistroDato VectorDatos[] RegistroResultado VectorResultado[]){
    RegistroResultado VectorAuxiliar[15]; // el int de la posición lo usa para el porcentual de votos
    int k=0, Votos[ ] = {Votos1,Votos2,Votos3}, VotosLista;
    //char* strcpy(char* s, const char* t)
    // carga los datos en un vector auxiliar
    for(int i = 0; i < 3; i++) {
        VotosLista = Votos[i];
        for(int j = 0; j < 5; j++,k++){
            strcpy(VectorAuxiliar[k].Candidato, VectorDato[k].Candidato);
            strcpy(VectorAuxiliar[k].Lista, VectorDato[k].Lista);
            VectorAuxiliar[k].Posicion = VotosLista/(j + 1); //en este entero coloca el porcentual de votos
        } // fin ciclo interno
    } // fin ciclo externo

    OrdenarVector(VectorAuxiliar, 15) //ordena por PorcentualVotos en forma decreciente

    for (int i = 0 ; i < 5; i++){ // carga del vector resultado
        VectorResultado[i].posicion = i+1;
        strcpy(VectorResultado[i].candidato, VectorAuxiliar[i].candidato);
        strcpy(VectorResultado[i].lista, VectorAuxiliar[i].lista);
    }
    return;
}
```