UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Informática I

Ingeniería electrónica

Ejercicio de clase 6-8-2018

1. Implemente una función que le pida al usuario los datos necesarios para rellenar la estructura spersona_t (Abajo del ejercicio encontrara la estructura definida), el fin del ingreso se da cuando la cantidad de datos ingresados alcanza CANT_MAX_STRUCTS o cuando la edad ingresada vale cero.
   El último elemento del array debe tener el campo edad igual a 0 para indicar el fin de los datos.
   El prototipo de la función es el siguiente:

```
int ingreso (spersona_t *dataptr, int cant_max);
```

   Donde:

   dataptr: puntero al vector de estructuras
   cant_max: cantidad de elementos del vector apuntado por dataptr

   Devuelve: La cantidad de elementos del vector que fueron rellenados.

```
#define NOMBRE_CANT         32
#define APELLIDO_CANT       32
#define SEXO_CANT      32
#define CANT_MAX_STRUCTS    20

typedef struct {
   char nombre[NOMBRE_CANT];          //!< Nombre
   char apellido[APELLIDO_CANT]; //!< Apellido
} sfullname_t;

typedef struct {
   int edad;                     //!< Edad
   float altura;                 //!< Altura
   sfullname_t fullname;         //!< Struct con nombre y apellido
   char sexo[SEXO_CANT];                  //!< Sexo (M, F)
} spersona_t;
```

2. Implemente una función con el siguiente prototipo, que imprima los datos almacenados en las estructuras apuntadas por dataptr

```
void imprimir (spersona_t *dataptr);
```

Donde:

dataptr: puntero al vector de estructuras

Nota: Recuerde que los datos válidos terminan cuando se encuentra una estructura con el campo edad igual a 0.

El formato de salida será el siguiente:

Nombre: Facundo
Apellido: Bisso
Edad: 24
Altura: 1.70m
Sexo: M

3. Implemente una función con el siguiente prototipo, que cuente la cantidad de estructuras en un array, sabiendo que el mismo termina con una edad igual a 0.

int contar_personas (spersona_t * p);

Donde:

p: puntero al vector de estructuras

Devuelve:

la cantidad de estructuras

4. Implemente una función con el siguiente prototipo, que ordene el array de estructuras apuntado por dataptr, segun el nombre, en forma ascendente.

void bubble_sort_x_apellido (spersona_t *p, int (*compare)(spersona_t a, spersona_t b));

Donde:

p: puntero al vector de estructuras
compare: puntero a la función de comparación utilizada para el ordenamiento

La función de comparación tendrá el siguiente prototipo:

```
int comp_apellido_asc (spersona_t a, spersona_t b);
```

La función de swap será la siguiente:

```
void swap (spersona_t *a, spersona_t *b) {
        spersona_t aux;

        memcpy ( &aux, a, sizeof(spersona_t));
        memcpy ( a, b, sizeof(spersona_t));
        memcpy ( b, &aux, sizeof(spersona_t));

        return;
}
```

5. Ahora se pretende que, en aquellos casos donde los apellidos sean iguales, se ordene segun el campo nombre, se forma ascendente. Explique si es necesario modificar la función ‘bubble_sort_x_apellido’. Justifique.
(No se pide implementar la solución)

6. Desarrolle un programa, que pruebe todas las funciones desarrolladas anteriormente.