Fundamendez - 2C 2024 - Clase 11

Vector Dinámico

Ejercicio 1

Como parte de su condena, Cruella de Vil tiene que estudiar veterinaria para comprender lo valioso que son los animales y en especial las mascotas.
Está haciendo un TP y necesita algunas estadísticas. Entre ellas la cantidad y edad promedio de las mascotas de una muestra considerable.
Aprovechó que Disney auspicia nuestra cátedra por lo que nos pidió hacer el censo en vivo.

Creó un struct que representa una muestra:

```c
typedef struct mascota {
    char animal;
    int edad;
} mascota_t;

typedef struct hogar {
    int cant_mascotas;
    mascota_t *mascotas;
} hogar_t;
```

Armar un programa que arme un vector de hogares con información sobre los alumnos y calculé las estadísticas necesarias.

Ejercicio 2

Implementar un vector dinámico de mascotas que pueda reemplazar al `hogar_t` con la siguiente interfaz:

```c
/*
      Interfaz de un vector dinámico de mascotas.
      Las posiciones del vector se numeran desde 0.
*/

struct vector_din;
typedef vector_din vector_din_t;

// Post: Crea un vector dinámico de mascotas. Devuelve un puntero a un
vector vacío.
```

```
//    Al terminar de usarlo este vector debe ser destruido con
vec_destruir().
//    Devuelve NULL si no se pudo crear el vector.
vector_din_t* vec_crear();


// Pre: `vec` fue creado usando vec_crear().
// Post: Dedinuye el vector.
void vec_destruir(vector_din_t* vec);

// Pre: `vec` fue creado usando vec_crear().
// Post: Agrega una mascota al final del vector.
//    Devuelve true si se pudo agregar, false en caso contrario.
bool vec_guardar(vector_din_t* vec, mascota_t mascota);

// Pre: `vec` fue creado usando vec_crear().
// Post: Devuelve un puntero elemento en la posición `pos` del vector.
//    Si `pos` es inválida, devuelve NULL.
//    Una posición es inválida si es mayor o igual al largo del vector.
mascota_t* vec_obtener(vector_din_t* vec, size_t pos);

// Pre: `vec` fue creado usando vec_crear().
// Post: Devuelve la cantidad de elementos en el vector.
size_t vec_largo(vector_din_t* vec);

// Pre: `vec` fue creado usando vec_crear().
// Post: Imprime los elementos del vector en orden.
void vec_imprimir(vector_din_t* vec);
```

Reemplazar el `hogar_t` en el ejercicio anterior con el nuevo vector dinámico.

Ejercicio Desafío

Implementar una nueva función para eliminar elementos del vector dinámico con la siguiente interfaz:

```
// Pre: `vec` fue creado usando vec_crear().
// Post: Elimina el elemento del vector en la posición deseada (`pos`)
reduciendo su largo.
//    Todos los elementos a la derecha de `pos` se desplazan una
posición a la izquierda.
//    Si `pos` es inválida, no hace nada.
void vec_eliminar(vector_din_t* vec, size_t pos);
```