



Programación - Guía de ejercicios de memoria dinámica

1. Escribir un prog. en el que el usuario informa la cantidad de elementos que va a ingresar y luego los valores de los elementos (enteros).

Los elementos deben almacenarse en un vector y luego en el archivo **vector.dat**. No se debe limitar la cantidad de elementos.

2. Escribir un programa para leer el archivo **vector .dat** del ejercicio anterior y almacenarlo en un vector. No se conoce la cantidad de elementos y no se puede limitar.
3. Escriba la función `duplicar` que reserve memoria dinámicamente para duplicar un string que se le pasa como argumento. Por ejemplo, la llamada:

```
p = duplicar (str);
```

debe reservar memoria para un string de la misma longitud que `str`, copiar los datos y retornar un puntero a la nueva cadena de caracteres. Si la reserva de memoria falla, entonces `duplicar` debe retornar un puntero nulo.

4. Escriba la siguiente función:

```
int *crear_vector (int n, int valor_inicial);
```

La función debe retornar un puntero a un vector de enteros de tamaño `n` creado dinámicamente cuyas posiciones son inicializadas con `valor_inicial`. La función debe retornar `NULL` si no se puede reservar la memoria.

5. Suponga que se ejecutan las siguientes instrucciones:

```
struct punto {  
    int x;  
    int y;  
};  
struct rectangulo {  
    struct punto superior_izq;  
    struct punto inferior_der;  
};  
struct rectangulo *p;
```

Escriba las líneas necesarias para que `p` apunte a una estructura `rectangulo` **válida** que tenga (10,25) como esquina superior izquierda y (20,15) como esquina inferior derecha.

6. Investigue y explique el uso de las funciones `memset ()` y `memcpy ()`.