Resuelva los siguientes ejercicios en lenguaje C.

Memoria Dinámica

1) Escribir un programa el cual reserve memoria dinámica para almacenar un número entero (int), le solicite al usuario el ingreso de un número y se asigna dicho valor en la memoria reservada, luego mostrar dicho valor por pantalla Liberar la memoria reservada al finalizar el programa.

2) Escribir un programa el cual reserve memoria dinámica para almacenar una cierta cantidad de números enteros (n * int), este valor **n** debe ser ingresado por el usuario. Luego solicitarle que ingresé n valores enteros ingresados de a uno y almacenarlos en la memoria previamente reservada. Mostrar luego todos los valores ingresados. Liberar la memoria reservada al finalizar el programa.

3) Asumiendo que ya existe un puntero que apuntan a un bloque de memoria previamente reservada con malloc de tamaño **(n * int)**, redimensionar dicha memoria a un tamaño de **(2 * n *int)**.

4) Escribir un programa el cual reserve memoria dinámica para almacenar un **struct** del tipo **t_alumno** (ver anexo). Luego solicitar al usuario que ingrese los datos del alumno y almacenarlos en la memoria previamente reservada. Mostrar luego todos los datos del alumno. Liberar la memoria reservada al finalizar el programa.

5) Escribir un programa el cual reserve memoria dinámica para almacenar una cantidad **n** de **struct** del tipo **t_alumno** (ver anexo). El usuario debe ingresar la cantidad **n.** Luego solicitar al usuario que ingrese los datos de los **n** alumnos y almacenarlos en la memoria previamente reservada. Mostrar luego todos los datos de todos los alumnos. Liberar la memoria reservada al finalizar el programa.

6) Implementar una función que retorna un puntero a un struct del tipo **t_alumno** (ver anexo), el cual apunta a un bloque de memoria dinámica, respetando la siguiente declaración. En caso de no poder reservar la memoria, retornar **NULL**.

```
t_alumno* crear_alumno();
```

7) Implementar una función que retorna un puntero a un vector de **n** struct del tipo **t_alumno** (ver anexo), el cual apunta a un bloque de memoria dinámica, respetando la siguiente declaración. En caso de no poder reservar la memoria, retornar **NULL**.

```
t_alumno* crear_alumnos(int n);
```

8) Implementar una función que retorna un puntero a un vector de **n** valores de tipo **int**, todos inicializados en cero, el cual apunta a un bloque de memoria dinámica, respetando la siguiente declaración. En caso de no poder reservar la memoria, retornar **NULL**.

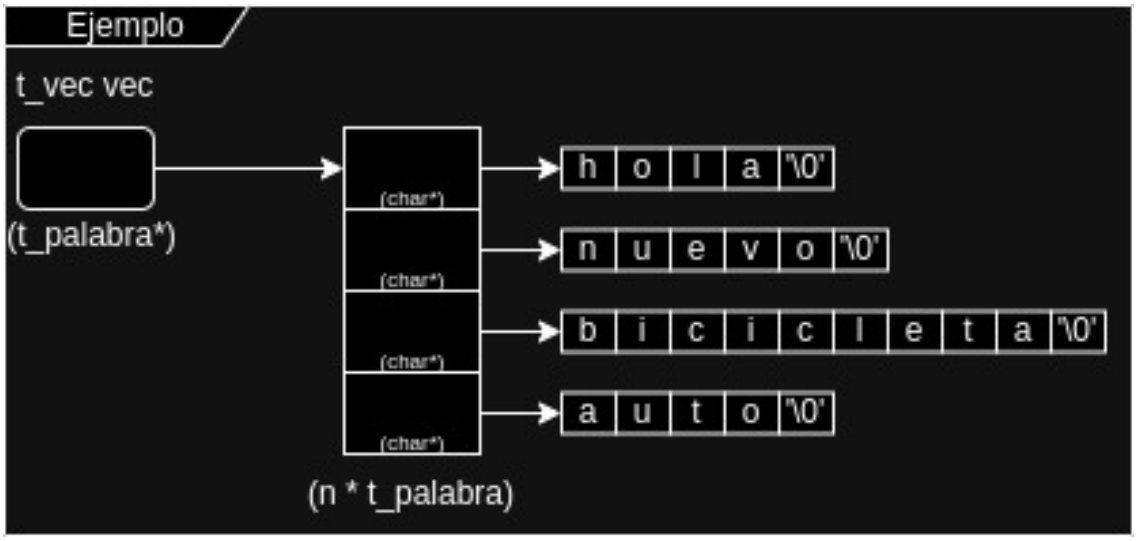
```
int* crear_vector_inicializado(int n);
```

9) Reservar memoria para almacenar una palabra ingresada por el usuario, es leída en una cadena auxiliar, luego copiada a memoria dinámica, recordando que se requiere adicionar un carácter de fin de cadena `'\n'`.
Ejemplo:
`"hola mundo" + '\0'` -> requiere espacio para 11 caracteres.

```
char* cadena_aux = "hola mundo";
char* cadena = malloc(strlen(cadena_aux) * sizeof(char) + 1);
```

10) Utilizar memoria dinámica para almacenar una cierta cantidad de **n** palabras ingresadas por el usuario. Crear una estructura de datos dinámica de dos dimensiones que permita almacenar esta información. Mostrar toda la información de la estructura luego de haber sido cargada. Liberar toda la memoria al finalizar el programa.

```c
typedef char* t_palabra;
typedef t_palabra* t_vec;
```



Anexo

```c
typedef struct {
    int padron;
    char nombre[30];
    char apellido[30];
} t_alumno;
```