|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* |  |
| *Asignatura:* |  |
| *Grupo:* |  |
| *No de Práctica(s):* |  |
| *Integrante(s):* |  |
|  |  |
|  |  |
| *Semestre:* |  |
| *Fecha de entrega:* | 18 de marzo de 2018 |
| *Observaciones:* |  |
|  |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Práctica 4: Almacenamiento en tiempo de ejecución**

**Objetivo**:

Utilizaras funciones en lenguaje C que permiten reservar y almacenar información de manera dinámica (en tiempo de ejecución).

|  |
| --- |
| ***Ejercicio 1***  Concluir con el código de cada función del programa, con el objetivo de crear un bloque de datos de forma dinámica, asignar valores y procesar los datos. |
| [Substituir por el código ya terminado]  // EJ01P04.c  // Memoria dinamica  #include<stdio.h>  #include<stdlib.h>  int **\***crearConjunto**(**int tam**);**  void asignarValores**(**int **\***conjunto**,**int tam**);**  void imprimeValores**(**int **\***conjunto**,**int tam**);**  long promediarValores**(**int **\***conjunto**,**int tam**);**  void liberar**(**int **\***conjunto**);**  int main**()**  **{**  int tam **=** 0**;**  int **\***arr**=NULL;**  printf**(**"Escribe la longitud del conjunto: "**);**  scanf**(**"%d"**,&**tam**);**  arr **=** crearConjunto**(**tam**);**  asignarValores**(**arr**,**tam**);**  imprimeValores**(**arr**,**tam**);**  printf**(**"promedio= %ld \n"**,**promediarValores**(**arr**,**tam**));**  **return** 0**;**  **}**  int **\***crearConjunto**(**int tam**)**  **{**  // Crear el espacio necesario en memoria  // dinamica para almacenar los datos  **}**  void asignarValores**(**int **\***conjunto**,** int tam**)**  **{**  // Asignar un valor aleatorio en  // cada elemento del conjunto  **}**  void imprimeValores**(**int **\***conjunto**,**int tam**)**  **{**  int i**=**0**;**  **for(**i**=**0**;** i**<**tam**;** i**++)**  **{** // utilizando notación de arreglos  // printf("%d \n",conjunto[i]);  // usando el apuntador  printf("%d \n", \*(conjunto+i));  **}**  **}**  long promediarValores**(**int **\***conjunto**,**int tam**)**  **{**  // Calcular el valor promedio del conjunto de datos  **}**  void liberar**(**int **\***conjunto**)**  **{**  // Liberar el espacio en memoria del conjunto  **}** |

|  |
| --- |
| ***Ejercicio 2***  Concluir el siguiente código con el objetivo de leer todas las palabras de un archivo de texto y cargarlas dentro de memoria dinámica, sin hacer uso desmedido de la memoria.  Consideraciones :   * Crear un arreglo de apuntadores a carácter de 1000 elementos con nombre de **palabras** * Después de identificar cada palabra en el documento, cree un arreglo de caracteres en la memoria dinámica, para almacenarla la cadena. * Asignar cada dirección de memoria dinámica a un índice del arreglo **palabras** * Para terminar se deben imprimir todas las palabras leídas en forma de una matriz de 4 columnas y el número de líneas necesarias.   Ejemplo de la impresión:  Historia de C El  lenguaje de programación C  fue creado por Dennis  Ritchie entre 1969 y |
| [Substituir por el código ya terminado]  // EJ02P04.c  // Memoria dinamica  #include <stdio.h>  void cortarPalabras**();**  int main **(** int argc**,** char **\*\***argv **)**  **{**  cortarPalabras**();**  **return** 0**;**  **}**  void cortarPalabras**(){**  FILE **\***archivo**;**  char caracter**;**  int saltoPrev**=**0**;**  archivo **=** fopen **(** "historiaC.txt"**,** "r" **);**  **while** **(**feof**(**archivo**)** **==** 0**)**  **{**  caracter **=** fgetc**(**archivo**);**  **if(** caracter**==**' '**||**caracter**==**'.'**||**caracter**==**','**||**  caracter**==**';'**||**caracter**==**':'**||**caracter**==**'"'**||**  caracter**==**'('**||**caracter**==**')'**||**caracter**==**'['**||**  caracter**==**']'**||**caracter**==**'/'**||**caracter**==**'\\'**||**  caracter**==**'\n'**||**caracter**==**'\r'**||** caracter**==**'\_'**||**  caracter**==**EOF**)**  **{**  **if(**saltoPrev**!=**1**)**  **{**  // Agregar al buffer de caracteres el fin de cadena  buffer**[**indice**]=**'\0'**;**  printf**(**"%s"**,**buffer**);**  printf**(**"\n"**);**    // Activar bandera de salto de linea previo  saltoPrev**=**1**;**  // Dejar el buffer como una cadena vacia  indice**=**0**;**  buffer**[**indice**]=**'\0'**;** **}**  **}**  **else**  **{**  // Agregar al arreglo "buffer" caracter por caracter  // para formar la palabra sin los signos de puntuación,  // y el espacio  //printf("%c",caracter);  buffer**[** indice**++** **]=** caracter**;**  saltoPrev**=**0**;** **}**  **}**  fclose **(** archivo **);**  **}** |

## Conclusión