|  |
| --- |
| **Tarea 1: Tipos de datos en C** |
| **Entregar hasta** Mayo 26 2010 Expirado |
| **Investigar sobre los tipos de datos en C.**  La investigación debe incluir:   * Nombre del tipo de dato * Dominio * Operaciones * Declaración en C * Ejemplo     Por favor incluir la **bibliografía**de la investigación. **Es muy importante.**  El nobre del archivo deber ser su apellido y el nombre de la tarea.  Ejemplo: cedeñoTarea1  La fecha máxima de entrega es el 26 de mayo a las 15:30 pm. (Antes de la clase) |

|  |
| --- |
| **Tarea 2: Secuencias de Escape** |
| **Entregar hasta** Junio 02 2010 Expirado |
| **Tarea Secuencias de Escape** ¿Qué son las secuencias de escape? Formatos de Salida Ejemplo: d (entero decimal) Al imprimir entero, decimales y cadenas ¿qué significa la precisión? Ejemplos  **Nota:**Traten de incluir solamente lo que se pide, es una tarea rápida.  No olviden incluir la bibliografía siguiendo algún formato, este es el que les expliqué en clases:  Un estilo de bibliografía es el APA, el estilo de cita de la Asociación Psicológica Americana (APA). Información sobre esto la puede encontar en internet, por ejemplo ellos indican que para citar contenido de wikipedia se lo hace de la siguiente manera:  [1] Nombre del artículo. ([año], [día] de [mes] [fecha de última modificación]). En Wikipedia, la enciclopedia libre. Recuperado el [día] de [mes] de [año] a las [00:00] de .  No se olvide de enumerar las fuentes [1], [2]...... |

|  |
| --- |
| **Tarea 3: Ejercicio de laboratorio** |
| **Entregar hasta** Junio 04 2010 Expirado |
| Escriba un programa que muestre una tabla de multiplicar del 1 al 10, con el siguiente formato:  TABLAS DE MULTIPLICAR DEL 1 AL 10  1     2       3       4        5       6       7       8       9     10 2     4       6       8       10     12     14     16     18     20 3     6       9       12     15     18     21     24     27     30 4     8       12     16     20     24     28     32     36     40 5     10     15     20     25     30     35     40     45     50 6     12     18     24     30     36     42     48     54     60 7     14     21     28     35     42     49     56     63     70 8     16     24     32     40     48     56     64     72     80 9     18     27     36     45     54     63     72     81     90 10   20     30     40     50     60     70     80     90     100 |

|  |
| --- |
| **Tarea 4: Ejercicio de Laboratorio sobre funciones** |
| **Entregar hasta** Junio 11 2010 Expirado |
| Ejercicio para aplicar la declaración, implementación y uso de funciones: Completar el codigo fuente que se encuentra en el archivo adjunto.  Implementar las funciones:  int contarDigitos(int z); int esPar(int n); int sumarDigitos(int n); |

## **Tarea 5**

**I Término 2010 - 2011**

Fundamentos de Programación

1. Determine la salida por pantalla del siguiente código:

**int n=1, m=0; Salida por pantalla:**

**for(m=n;n<5;n++)**

**{**

**printf(“%d”,m);**

**printf(“\n\t”);**

**}**

1. Implementar la función **long reemplazarDigito(long n, int a, int b),** la cual retorna el número n habiendo reemplazado el dígito **a** por el dígito **b** enviados como parámetros.

*Por ejemplo:*

reemplazarDigito (24373,3,5) retorna 24575

reemplazarDigito (770,7,8) retorna 880

1. Escriba la función **bool esNarcisista (int n)**; la cual retorna verdadero si n es un número narcisista, caso contrario retorna falso. Un número narcisista es aquel en el cual la suma de los cubos de sus dígitos es igual al mismo número.

*Por ejemplo:*

153 = 13+53+33 , **esNarcisista (153)** retorna true.

25 ≠ 23+53 , **esNarcisista (25)** retorna false.

TAREA #6

/\*

\* Archivo: Arreglo.c

\* Autor:

\* Fecha de creación:

\* Descripción:

\* Programa que utiliza los procedimientos llenarArreglo y mostrarArreglo.

\* Los procedimientos reciben como parámetro un arreglo de enteros y el

\* tamaño de dichos arreglos.

\*/

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <time.h>

#define MAX 50

void llenarArreglo(int A[],int tam);

void mostrarArreglo(int A[],int tam);

void main()

{

int A[MAX];

srand(time(NULL));

llenarArreglo(A,MAX);

mostrarArreglo(A,MAX);

getch();

}

/\*

\* Procedimiento: llenarArreglo

\* Uso: llenarArreglo(A,MAX);

\* Descripción:

\* Este procedimiento llena el arreglo enviado como parámetro con números aleatorios

\* entre 1 y 6.

\*/

/\*

\* Procedimiento: mostrarArreglo

\* Uso: mostrarArreglo(A,MAX);

\* Descripción:

\* Este procedimiento muestra por pantalla el contenido del arreglo

\* enviado como parámetro. Muestra cada elemento en una línea diferente.

\*/

TAREA#7

**Fundamentos de Programación**

Ud. deberá entregar como tarea un archivo en c que contiene:

* La declaración e implementación de ciertas funciones y/o procedimientos que permitirán realizar avances del proyecto parcial “Serpientes y Escaleras”; y
* Un programa que permite probar (uso) dichas funciones y procedimientos implementados.

RECUERDE **incluir los comentarios** al programa y a las funciones o procedimientos de acuerdo a lo indicado en SIDWEB (ver contenido>recursos).

Las funciones y/o procedimientos a implementar son:

**IMPORTANTE**: Los nombres de las funciones o procedimientos y la lista de parámetros indicados a continuación no pueden ser cambiados; bajo ningún motivo.

**void menuPrincipal(void);**

Este procedimiento muestra el menú utilizado en el proyecto:

1. Definir número de jugadores
2. Sortear turnos
3. Iniciar juego
4. Salir

**void dibujaTablero (int n);**

Este procedimiento recibe el número de casilla que debe graficar en el tablero. *Nota*: Use formatos de salida y secuencias de escape para representar el tablero y su contenido de forma agradable.

**int definirJugadores(void);**

Retorna un entero que en este caso puede ser 1 o 2, que es el número de jugadores definidos por el usuario. Se debe validar que el usuario ingrese solamente el número 1 o 2.

**int lanzarDado(void);**

Retorna un número aleatorio del 1 al 6 que representa que el jugador lanzara un dado para jugar.

**int primerJugador(int a, int b);**

Recibe los dos valores de dado que cada jugador recibió al sortear los turnos. El entero a representa al jugador 1, el entero b representa al jugador 2. Retorna el entero 1 o 2 para indicar el jugador que inicia la partida. El jugador que inicia la partida es el que tuvo el número mayor Retorna 0 si los valores de a y b son iguales y hay un empate.

**bool sorteo(int a);**

Retorna true si el usuario realizó el sorteo. El entero a es el primer jugador, el cual puede ser 0, 1 ó 2. Retorna false si a es 0, ya que no se definido quién empieza primero, de lo contrario retorna true.

TAREA#8

1) Escriba una función recursiva que permita determinar si una cadena o no representa una cifra numérica.

2)El procedimiento para convertir un número en base 10 a base 2 consiste en divisiones sucesivas para dos

hasta que el cociente sea 0:

Ejemplo:

19 /2

1 9/2

1 4/2

0 2/2

0 1/2

1 0

19 -> 10011

Escriba un procedimiento recursivo que dado un número en base 10 imprima en consola el mismo número en base 2.

/\*

\* Procedimento: modifica

\* Uso: modifica(cad);

\* Descripcion: Cambia a mayusculas la cadena recibida como parámetro.

\*/

void modifica(char \*s)

{

}

/\*

\* Funcion: contarEspacios

\* Uso: a = contarEspacios(s);

\* Descripcion: Retorna el número de espacios en blancos contenidos en la cadena enviada como parámetro.

\*/

int contarEspacios(char \*s)

{

}

## **Tarea 9**

**I Término 2010 - 2011**

Fundamentos de Programación

1. **Escriba una función recursiva que cuente el número de veces que aparece un caracter en una cadena. Por ejemplo:**

El caracter ‘c’ en la cadena “cocada” retorna 2

El caracter ‘u’ en la cadena “perico” retorna 0

1. **Escriba el procedimiento recursivo cambiaMinuscula, la cual recibe una cadena y la convierte a minúscula.**

**Ejemplos:**

La cadena “SAludOs”, el procedimiento cambia la cadena a “saludos”.

La cadena “feLICidaD”, el procedimiento cambia la cadena a “felicidad”.

1. **Escriba una función que sirva para eliminar un carácter específico de una cadena de caracteres. La función recibirá como parámetros la cadena de caracteres y el carácter a eliminar y deberá retornar el número de veces que el carácter fue eliminado. Utilizar cadenas de caracteres como ARREGLOS de caracteres.**

**Ejemplo:**

Cadena Original

“ESTUDIANTE”

Cadena Resultante con el carácter ‘E’ eliminado

“STUDIANT”, la función deberá retornar 2

|  |
| --- |
| **Tarea 10: Laboratorio cadenas y archivos** |
| **Entregar hasta** Agosto 06 2010 Expirado |
| 1) Escriba la función char\* strnstr(char \*cadena1, char \*cadena2, int n), que permite localizar,  dentro de los primeros n caracteres de la cadena1,  la primera ocurrencia de cadena2.   Por ejemplo: strnstr(“cancion”, “can”,2) retorna null. strnstr(“cancionero”, “cion”,9) retorna un puntero al 4to carácter de cadena1. strnstr(“cancion”, “cion”, 5) retorna null. |

## **Tarea 11**

**I Término 2010 - 2011**

Fundamentos de Programación

**1. Implementar un programa que permita encriptar y desencriptar un archivo de acuerdo a la siguiente fórmula:**

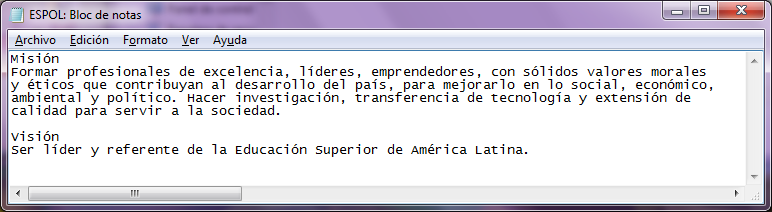
**asciiNuevoCaracter= (asciiCaracterOriginal+5)mod 256.**

--------------------------------------------------------------------------------------------------------------

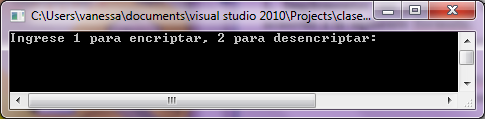
**Modificar el archivo encriptar.c llenando el main y las dos funciones Encriptar y Desencriptar. No olvide poner comentarios sobre el programa si es necesario.**

Por ejemplo:

Este es mi archivo ESPOL.txt el cual coloco al lado del archivo.c que ejecuto en C.



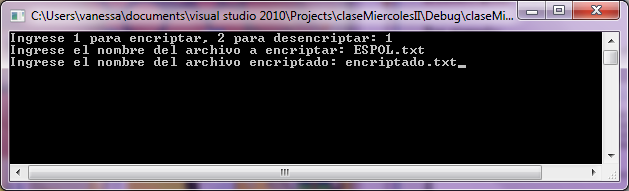
Al correr el programa, me pregunta:



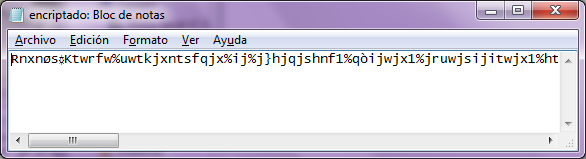
Escojo 1 para que mi archivo sea encriptado:

Indico el nombre del archivo a encriptar, en este caso ESPOL.txt.

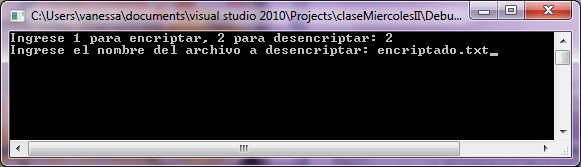
Luego me pregunta el nombre del archivo encriptado, lo llamare encriptado.txt.



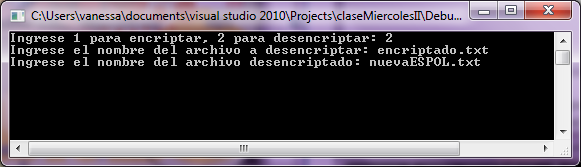
Así luce mi archivo encriptado con la formula indicada. La misión y visión de la ESPOL es un misterio para el que no conoce la fórmula.



Para desencriptarlo, correré mi programa por segunda vez. Pero escogeré la opción 2 de desencriptar. E ingresare el nombre del archivo a desencriptar. En este caso encriptado.txt.



El programa me pregunta un nombre para el nuevo archivo desencriptado, escogeré nuevaESPOL.txt.



Y mi archivo encriptado vuelve a la normalidad pero con nuevo nombre:

**Tarea#12**

**Fundamentos de Programación**

Ud. deberá entregar como tarea un archivo en c que contiene:

* La declaración e implementación de ciertas funciones y/o procedimientos que permitirán realizar avances del proyecto parcial “Serpientes y Escaleras Versión 2.0”; y
* Un programa que permite probar (uso) dichas funciones y procedimientos implementados.

RECUERDE **incluir los comentarios** al programa y a las funciones o procedimientos de acuerdo a lo indicado en SIDWEB (ver contenido>recursos).

Las funciones y/o procedimientos a implementar son:

**IMPORTANTE**: Los nombres de las funciones o procedimientos y la lista de parámetros indicados a continuación no pueden ser cambiados; bajo ningún motivo.

**void menuPrincipal(void);**

Este procedimiento muestra el menú utilizado en el proyecto:

1. Definir número de jugadores
2. Sortear turnos
3. Nueva partida
4. Continuar Partida
5. Salir

**void declararGanador (char \* nombre);**

Este procedimiento recibe el nombre del ganador e imprime en pantalla una felicitación por ganar el juego.

**int cargarDatos(FILE \*archivo);**

Esta función lee un archivo y retorna un entero que en este caso puede ser -1 que indica que no hay ningún juego grabado ya que el archivo llamado juegoSyE.ptd contendrá el valor de -1. La función retorna 1 si hay un juego grabado y en este caso se procede a obtener el turno en juego, nombre del jugador1, casilla del jugador 1, nombre del jugador2 y casilla del jugador 2.

**void guardarPartida(FILE \* archivo,int turno,**

**char \*nombre1,int casilla1, char \*nombre2,int casilla2);**

Graba en un archivo llamado juegoSyE.ptd los datos que permitirán dar continuidad a una partida iniciada. A parte de recibir el archivo en el cual se guardará, recibe turno en juego, nombre del jugador1, casilla del jugador 1, nombre del jugador2 y casilla del jugador 2. El archivo se guarda con el formato especificado en la descripción del proyecto.

**int cargarJugadores(FILE \*top, jugadorT jugadores[],**

**char jugador[30]);**

Recibe el archivo que contiene los jugadores top, el arreglo de estructuras de jugador y el nombre del nuevo jugador a ingresar. Devuelve el número de jugadores en el archivo top.

**void guardarJugadores(FILE \*top, jugadorT jugadores [],int tamanio);**

Recibe el archivo donde se guardarán jugadores top, el arreglo de estructuras de nombre y jugador y su tamaño.

typedef struct{

char nombre[30];

int veces;

}jugadorT;