

Proyecto 1

Arquitectura de Computadores

Primer Semestre 2022, Prof. Cecilia Hernández

Fecha Inicio: Lunes 12 de Sept. 2022.

Fecha Entrega: Lunes 3 de Octubre 2022 (23:59 hrs).

Trabajo en grupo: Integrado con 3 o 4 estudiantes.

Entrega: Informe con resolución de problemas. Para código, incluya un readme.txt con compilación y modo de ejecución.

1. [0.5 puntos] Convierta los siguientes tipos de datos a binario. Proporcione sus desarrollos. (0.5 punto)
 - a) `int x = 379`
 - b) `int x = -872`
 - c) `unsigned int x = 0xFF9029AE`
 - d) `unsigned char x = 0xAB`
 - e) `float x = -72.45`
2. [0.5 puntos] Convierta las siguientes secuencias de bits de acuerdo a la representación solicitada a su representación decimal. Proporcione sus desarrollos. (0.5 punto)
 - a) Entero con signo de 8 bits. 1001 0010
 - b) Entero sin signo de 8 bits. 1111 0110
 - c) Punto flotante precision simple. 1011 1110 1011 1110 1011 1110 1011 1000
3. [1.0 puntos] Compute las siguientes operaciones con las siguientes variables dadas y proporcione su respuesta en hexadecimal. Incluya el desarrollo, no solo los resultados.

`int x = 0xFA430021; int y = 0x3F; int k=10;`

- a) $x + y$
 - b) $x|(y \ll 16)$
 - c) $x \& (y \ll 24)$
 - d) $(x \gg 6) \& \sim y$
 - e) $(x \wedge y)|(y \ll 12)$
 - f) $x = (y \ll k)|(x \gg (32 - k))$
 - g) $x \wedge = (x \ll 8)$
4. [0.5 puntos] Escriba una función utilizando el siguiente prototipo:

```
// Asuma  $0 \leq n < w$ , donde  $w$  es el largo de  $x$ 
unsigned rotate_left(unsigned x, int n);
```

Esta función se encarga realizar un desplazamiento a la izquierda rotativo, es decir, los bits que salen por la izquierda, deberán aparecer a la derecha del valor desplazado.

Por ejemplo, asuma $x = 0x12345678$. Si $n = 4$, entonces el resultado de la función debe ser $0x23456781$. Si $n = 20$, entonces el resultado de la función debe ser $0x67812345$.

5. [1.0 puntos] Se define el tipo de dato **datetime** que maneja la fecha y la hora almacenados en 32 bits. Su declaración es la siguiente:

```
typedef unsigned datetime;
```

Para ello, utiliza cada bit con la siguiente distribución (numerados desde el bit menos significativo al más significativo):

Rango	Cantidad de bits	Dato
[0 – 5]	6	Minute
[6 – 10]	5	Hour
[11 – 15]	5	Day
[16 – 19]	4	Month
[20 – 31]	12	Year

Implemente una función que permita almacenar los datos de la estructura, y una función que le permita mostrar por pantalla el contenido de la estructura. Utilice los siguientes prototipos:

```
// Asuma que todos los valores son válidos
void save_datetime(unsigned minute, unsigned hour, unsigned day, unsigned month,
unsigned year);

// Utilice el formato yyyy-MM-DD;HH:mm
void print_datetime(unsigned datetime);
```

6. [2.5 puntos] Implemente las siguientes funciones usando solo operaciones bitwise, es decir, los operadores lógicos en C and &, or |, xor ^ y not ~; y los operadores de desplazamiento >> y <<.

- int andAB(int A, int B): $A \& B$ usando solo not y or.
- int xorAB(int A, int B): $A \wedge B$ usando solo and y not.
- int notA(int A): $\sim A$ usando xor.
- int64_t numeroRango(int64_t A, unsigned char s, unsigned char e): Retorna el número en decimal que se obtiene entre los rangos de bits $[s, e]$. Proporcione además la verificación requerida para que el resultado tenga sentido.
- int numeroDe1s(int A): retorna número de 1s en representación binaria de A. Para esta implementación puede usar un solo for o while, pero cero ifs.