

Fundamentos de Informática
Examen Febrero 2007

06/02/07

Parte 1: Arquitectura de Computadores

PROBLEMA 1 (1.5 puntos)

- a) Realizar la operación $36 - 44$ en complemento a uno y en complemento a dos. (0.5 p)
- b) Realizar las siguientes conversiones de base. (0.5 p)
 - $125_{10} \rightarrow X_{16}$
 - $125_8 \rightarrow X_{16}$
 - $617_8 \rightarrow X_{10}$
 - $4AB_{16} \rightarrow X_{10}$
- c) Calcular el tiempo (en minutos) de sonido en calidad CD estéreo sin comprimir que se puede almacenar en un CD-ROM de capacidad 650 MB. (0.5 p)

PROBLEMA 2 (1 punto)

Un programa, requiere la ejecución de una instrucción **LOAD (n),A** y a continuación una instrucción **JUMP n**, Dibujar un cronograma donde se muestren las señales necesarias para la ejecución de estas instrucciones, incluyendo la fase de búsqueda de ambas y teniendo en cuenta el posible solapamiento entre las mismas.

PROBLEMA 3 (1 punto)

Dividir 7 entre 3 utilizando el método de división sin restauración. Indicar claramente dónde se encuentra el resultado de la operación (cociente y resto).

CUESTIONES (1.5 punto)

- a) ¿Qué elementos componen la unidad de control del computador diseñado en clase? ¿Cuál es la función de cada componente? (0.5 p)
- b) ¿Qué son las bifurcaciones con retorno? ¿Dónde se puede almacenar la dirección de retorno? Explicar brevemente las ventajas e inconvenientes de las diferentes opciones. (0.5 p)
- c) ¿Qué es la memoria virtual? ¿Qué métodos existen para el trasvase de bloques entre la memoria virtual y la física? Explica brevemente cómo se realiza la conversión de direcciones virtuales a físicas. (0.5 p)

	Nº de bits/muestra	Frecuencia de muestro (F, KHz)	Periodo de muestreo (T _s , µsegundos)
PCM teléfono	8	8	125
Calidad telefónica	8	11,025	90,7
Radio	8	22,05	45,4
CD	16 ⁽¹⁾	44,1	22,7

⁽¹⁾ Numero de bits/muestra por canal, con sonido estereofónico hay que multiplicar por 2

Parte 2: Programación

PROBLEMA 1 (2.5 puntos)

Escribir un programa que pida al usuario que introduzca el nombre de un trabajador y a continuación el número de horas que ha trabajado cada día de la semana (de lunes a viernes). Después de recibir esta información, el programa debe mostrar las iniciales del trabajador, el número medio de horas trabajadas y el sueldo. Si el número medio de horas es menor a 6 o mayor a 10, en lugar del sueldo se debe mostrar un mensaje de alerta.

Las primeras 40 horas se pagarán a razón de 12 € por hora. Por cada hora que se pase de 40 se cobrará 1 € adicional. (La hora 41 se cobra a 13 €, la 42 a 14 €, etc.). Al sueldo bruto se le aplicará una retención del 15% si es menor o igual a 400 euros y del 18% si es mayor.

A continuación se muestran dos ejemplos de funcionamiento. Los datos introducidos por el usuario aparecen en negrita. Todos los resultados se deben mostrar con dos decimales de precisión.

Introduzca nombre:
Juan Perez Garcia
Introduzca horas día 1: **8**
Introduzca horas día 2: **9**
Introduzca horas día 3: **7**
Introduzca horas día 4: **10**
Introduzca horas día 5: **7**

JPG
Num. medio de horas: 8.20
Sueldo bruto: 493.00 euros
Impuestos: 88.74 euros
Sueldo final: 404.26 euros

Introduzca nombre:
Antonio Javier Martinez Gonzalez
Introduzca horas día 1: **6**
Introduzca horas día 2: **5**
Introduzca horas día 3: **7**
Introduzca horas día 4: **5**
Introduzca horas día 5: **6**

AJMG
Num. medio de horas: 5.80
NUMERO DE HORAS MENOR A 6

Para resolver el problema, se deben crear las siguientes funciones:

- CalculaSueldo: Recibe como parámetro el número total de horas trabajadas. Se encarga de calcular y sacar por pantalla el sueldo bruto y los impuestos. Finalmente calcula el sueldo neto y lo devuelve como resultado.
- CalculaMedia: Toma como parámetro un vector de números y su tamaño y devuelve el valor medio.
- ExtraIniciales: Toma como parámetros una cadena origen y una cadena resultado. Debe almacenar en la cadena resultado las iniciales de las palabras de la cadena origen. Se debe tener en cuenta que entre dos palabras consecutivas puede haber más de un espacio en blanco.

Por último, se debe crear una función main desde la que se vayan pidiendo los datos al usuario, llamando a las funciones necesarias y mostrando los resultados.

CUESTIÓN 1 (0.5 puntos)

Escribir la salida por pantalla del siguiente programa:

```
char *cad = "Fundamentos de Informatica";  
int i;  
  
for(i=0; cad[i] != '\0'; i++)  
    printf("%c", *cad+i);  
printf("\n");
```

CUESTIÓN 2 (0.5 puntos)

Explicar brevemente cuáles son las aplicaciones principales de los punteros.

CUESTIÓN 3 (0.5 puntos)

¿Qué son las funciones recursivas? ¿Cuándo se pueden utilizar? Pon un ejemplo.

PROBLEMA 2 (1 punto)

Con el siguiente programa en C, se pretende pedir al usuario que introduzca varios números (reservando memoria dinámicamente para almacenarlos) y a continuación calcular su media, varianza y desviación típica. Para ello, se ha creado una función llamada **Funcion1** que debe calcular y devolver estos tres valores. Se pide comprobar si existe algún error en el programa y en caso afirmativo, reescribir correctamente las sentencias erróneas.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

void Funcion1(float *media, float *varianza, float *desviacion, float *vector, int n);

void main(void)
{
    int n, i;
    float *datos, med, var, desv;

    printf("Cuantos numeros va a introducir? ");
    scanf("%d", &n);
    datos = malloc(n*sizeof(float));
    if(datos == NULL)
        printf("\nError durante la reserva de memoria\n");
    else
    {
        for(i=0; i<n; i++)
        {
            printf("Introduzca el dato %d: ", i+1);
            scanf("%f", &(datos+i));
        }
        Funcion1(&med, &var, &desv, datos, n);
        printf("\nLa media es %f\nLa varianza es %f\nLa desviacion es %f\n", med, var, desv);
        free(datos);
    }

    system("PAUSE");
}

void Funcion1(float *media, float *varianza, float *desviacion, float *vector, int n)
{
    int i;

    for(i=0; i<=n; i++)
        *media += *(vector+i);
    *media /= n;

    for(i=0; i<n; i++)
        *varianza += (vector[i] - *media) * (vector[i] - *media);
    *varianza /= n;
    *desviacion = sqrt(*varianza);
}
```