Máster en Ingeniería Mecatrónica
Computadores y Programación

Examen Ordinario – Enero 2021

1) *1.5 puntos*

Realizar una función que, dados como parámetros una medida (número real), dos tablas (del mismo tamaño) de reales (medida nominal y valor nominal), y un valor real de tolerancia de medida, devuelva el valor nominal para el cual se cumple que la medida recibida está en el entorno medida nominal ± tolerancia de medida, o cero si no encuentra ninguna.

Ejemplo:

DIAMETROS MONEDAS EURO (Tole diam = ±0.05)

Diam (mm)	Valor nominal €
21.25	0.05
19.75	0.10
22.25	0.20
24.25	0.50
23.25	1.00
25.75	2.00

EJEMPLOS:

FnEjercicio1(22.22, tablas diam y valor, 0.05) → devuelve el valor nominal 0.20 porque la medida pertenece al intervalo 22.25±0.05

2) *1.5 puntos*

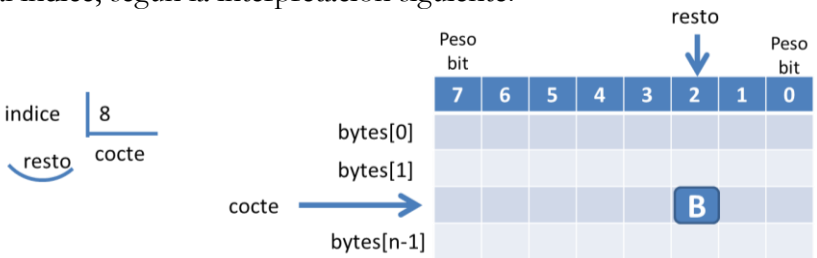
Realizar una función que, dado como parámetros un índice entero y una cadena de caracteres donde copiar el nombre de un artículo, busque en una variable global `_tabla_sel` la entrada que tenga el índice buscado, y devuelva el valor real y el texto descriptivo. La variable global es una tabla de 40 cadenas de caracteres, cada una de las cuales contiene un texto con un índice (número entero), un importe (número real), y una descripción (texto), separados por comas y con posibles espacios en blanco entre ellos.
Ejemplo:

```
char* _tabla_sel[40]={
                      " 1 , 0.50 , AGUA" ,          // _tabla_sel[0]
                      " 24 , 1.25   , PATATAS" ,    // _tabla_sel[1]
                      " 15   , 0.60 ,   COLA" ,     // _tabla_sel[2]
                      ...
                     };

char articulo[20];
float valor;

valor=FuncionEj2(24,articulo);// valor=1.25, articulo → "PATATAS"
```

La función debe devolver valor 0 y artículo vacío si no encuentra ninguna entrada correcta.

3) *1 punto*

Realizar una función que, dados como parámetros un índice entero, una tabla de **char** y un carácter A (activar) o D (desactivar), active o desactive en dicha tabla el bit B correspondiente al índice, según la interpretación siguiente:



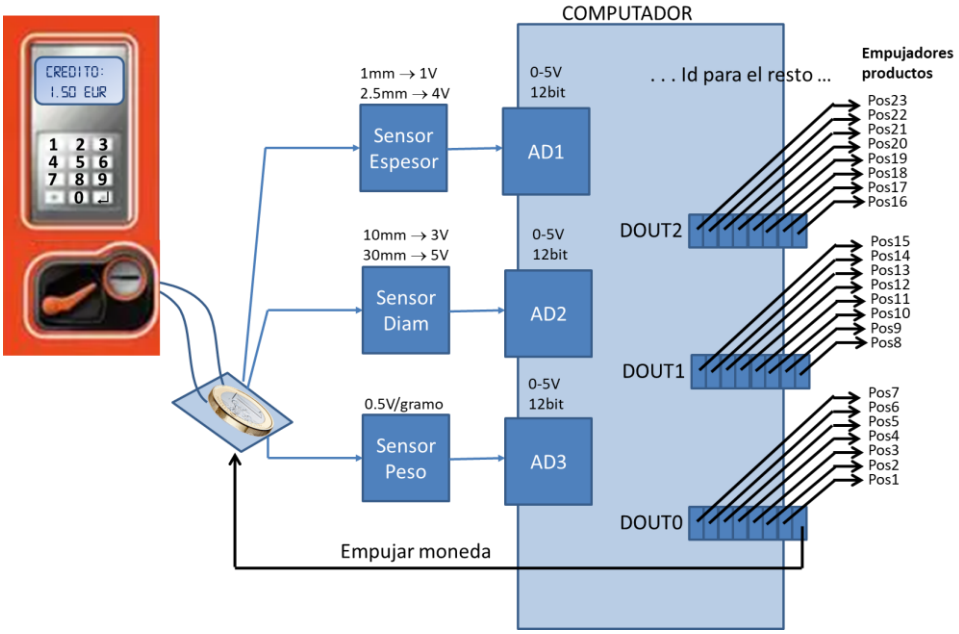
El resto de bits de la tabla tienen que mantener su valor anterior.

Máster en Ingeniería Mecatrónica
Computadores y Programación

Examen Ordinario – Enero 2021

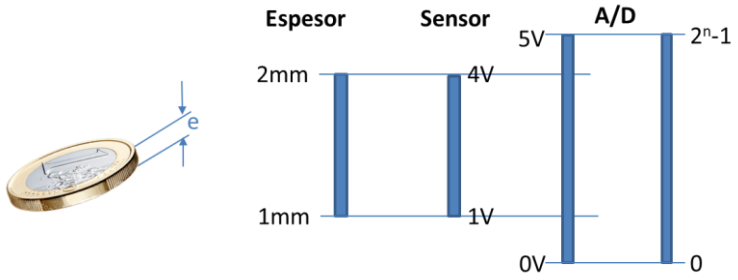
4) Se desea realizar un sistema de control para la venta automática utilizando monedas.

5 puntos



Como interfaz con el usuario, la máquina dispone de un teclado básico para la selección del artículo, y de un display LCD para la visualización.

Las monedas son detectadas por un sistema redundante para asegurar su validez, que utiliza tres sensores analógicos: espesor, diámetro y peso. Ejemplo sensor espesor:



Para detectar cada moneda se toman medidas de los tres sensores de forma permanente con periodo 2ms, y se obtiene el promedio de las 20 últimas medidas. Si los valores medios de espesor, diámetro y peso se corresponden con el mismo valor nominal de moneda (usar función ejercicio 1), se considera que es válida.

Cada vez que se detecta una moneda válida, se añade su valor al crédito total, se escribe el crédito en el display LCD, se empuja la moneda para que vaya al cajón de la recaudación, y se vacían las tablas de medidas para evitar detectar de nuevo la misma moneda.

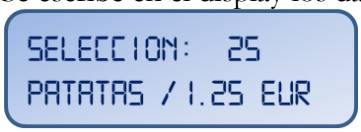
Cuando el cliente pulsa una cadena de selección seguida de INTRO, se obtiene el importe del artículo y su nombre (utilizando la función del ejercicio 2) y, en caso de que el artículo exista y el saldo acumulado sea suficiente, se abre durante 1 seg el dispensador correspondiente. Se utilizará la función del ejercicio 3 para determinar qué dispensador se debe abrir.

Máster en Ingeniería Mecatrónica
Computadores y Programación

Examen Ordinario – Enero 2021

Organización del programa:

- El programa principal realiza las inicializaciones necesarias (variable que contiene valores y medidas de monedas usando función `GetValoresMonedasEuro( )` especificada en el anexo, tablas temporales, etc.) y lanza una interrupción temporizada cada 10 ms. A continuación, realiza un bucle infinito en que espera la selección (índice) del usuario como un valor entero de teclado (scanf); cada vez que se introduce el índice, se determina el nombre e importe del artículo asociado utilizando la función del ejercicio 2 y, si el artículo existe yel crédito disponible es suficiente:
  - Se escribe en el display los datos del artículo seleccionado con el formato siguiente:

  - Se evita que la función de interrupción siga acumulando crédito temporalmente
  - Se ordena la activación del dispensador correspondiente durante 1 seg (usar Sleep). Se utiliza la función del ejercicio 3 para establecer el contenido de la variable global `_dout` (especificada en el anexo).
  - Se reduce el importe del artículo del crédito disponible
  - Se vuelve a permitir que la función de interrupción siga acumulando crédito

- En la interrupción temporizada se realizarán, por este orden:
  - Desactivar bit empujar moneda (bit de peso cero de `_dout[0]`)
  - Comprobar que está permitida la acumulación de crédito. En caso afirmativo:
    - Desplazamiento de las tablas utilizadas `para` valores temporales de medidas (peso, espesor y diámetro).
    - Leer canales AD 1, 2 y 3 para obtener medidas actuales, añadir como datos más recientes de sus tablas y calcular media de últimos 20 datos.
    - Llamar a función ejercicio 1 para las tres magnitudes disponibles (espesor, diámetro, peso); si el valor nominal de la moneda que devuelven las tres es el mismo (y no cero), hay moneda nueva y es válida.
    - Si hay moneda nueva y es válida:
      - Añadir su valor nominal al crédito
      - Activar bit empujar moneda (bit de peso cero de `_dout[0]`)
      - Borrar tablas de evolución temporal que contienen las medidas
    - Escribir en display LCD el saldo actual con el formato siguiente:

Máster en Ingeniería Mecatrónica
Computadores y Programación

Examen Ordinario – Enero 2021

ANEXO

Librería auxiliar:

Se dispone de las siguientes funciones y declaraciones dentro de una librería FnAux (archivos FnAux.h y FnAux.lib):

```
// Inicialización de un temporizador que llame a la función indicada en
// FnCallback cada cierto intervalo (en ms)
void InitTemporizador(int tiempo_ms,void (*FnCallback)();

// Obtención del valor de conversion A/D del canal deseado utilizando
// nBits : resultado de 0 a 2^nBits-1, para tensión de entrada 0 a 5V.
int ConversionAD(int nCanal,int nBits);

// Estructura para los valores mínimo y máximo de medidas de monedas
struct monedas
{
   int n_monedas;
   float valor_nominal_eur[N_MONEDAS_MAX];
   float diam_nominal_mm[N_MONEDAS_MAX],tole_diam_mm;
   float espesor_nominal_mm[N_MONEDAS_MAX],tole_espesor_mm;
   float peso_nominal_g[N_MONEDAS_MAX],tole_peso_g;
} ;

// Función que asigna memoria para las tablas y rellena los datos de las
// monedas de euro (NO HAY QUE HACERLA, YA EXISTE)
struct monedas GetValoresMonedasEuro();

// Función para escribir en display LCD
void LCD_Printf(int fila,const char* texto,...otros params como printf...);

// Variable global que almacena los contenidos de las salidas digitales
char _dout[5];  // Permite hasta 40 salidas diferentes

// Variable global que almacena los datos de selecciones
char* _tabla_sel[40];   // _tabla_sel[0]="3, COLA , 0.75"
                        // _tabla_sel[1]="  24, AGUA , 0.60"
                        // _tabla_sel[2]="12 , PATATAS , 1.25"
                        // ...
```

Algunas funciones de C:

```
int atoi(const char* cad);              // Devuelve entero equivalente a cadena
double atof(const char* cad);           // Devuelve real equivalente a cadena
double strtod(const char* cad,char** next);   // Id. a atof() y guarda en next puntero
                                              // a final de conversión
int strlen(const char* cadena);         // Devuelve longitud de cadena
char* strcpy(char* dst,const char* src);      // Copia cadena fuente en destino
char* strncpy(char* dst,const char* src,int n);      // Id. Máximo n caracteres
char* strcat(char* dst,const char* src);      // Concatena cadena Fuente a destino
char* strncat(char* dst,const char* src,int n);      // Id. Máximo n caracteres
char* strchr(const char* cad,char c);  // Busca caracter en cadena, devuelve puntero
                                       // a la primera ocurrencia o NULL si no está
char* strstr(const char* cad,const char* busca);     // Id. buscando cadena
int strcmp(const char* c1,const char* c2);    // Compara cadenas, devuelve 0 si iguales
char* gets(char* destino);              // Lee cadena de consola, almacena en destino
void* malloc(int n_bytes);              // Asigna memoria para n bytes
void free(void* ptr);                   // Libera memoria asignada
```

Máster en Ingeniería Mecatrónica
Computadores y Programación

Examen Ordinario – Enero 2021

APELLIDOS Y NOMBRE: _____

Ejercicio 5 (responder aquí):
1 punto

a) ¿Cuánto vale la variable *x* tras ejecutar el código siguiente?

```
int Suma(const int* t,int n);                /* Suma los valores de una tabla
...                                          de n enteros */
int datos[4]={0,2,4,6};
int x;
x=Suma(datos+1,2);
```

- ☐ *x* vale 12 si la función Suma( ) está correctamente realizada
- ☐ *x* vale 2 si la función Suma( ) está correctamente realizada
- ☐ El valor de *x* depende del valor del puntero *datos*, es indeterminado
- ☐ *x* vale 6 si la función Suma( ) está correctamente realizada

b) ¿ Cuánto vale la variable *x* tras ejecutar el código siguiente ?

```
void Suma(const int* t,int n,float result);  /* Suma los valores de una tabla
...                                          de n enteros */
int datos[4]={0,2,4,6};
float x ;
Suma(datos,4,x);
```

- ☐ *x* vale 12 si la función Suma( ) está correctamente realizada
- ☐ No se puede compilar, la tabla debería ser de float
- ☐ No se puede saber el valor de *x*, independientemente de lo que haga Suma( )
- ☐ Debería declararse `float* x` y pasarle a la función `*x`. En ese caso, *x* vale 12 si la función Suma( ) está correctamente realizada.

c) ¿ Cómo se sabe si el bit de peso P de la variable entera V está inactivo (a cero) ?

- ☐ `if (!(v & (1 <<p)) )`
- ☐ `if ~(v =& 1<<p)`
- ☐ `if (p[v] << ~1)`
- ☐ `if (v & (0 <<p) )`

d) ¿ Cómo se asigna memoria dinámica para una tabla de n caracteres, donde n es un entero ?

- ☐ `char cadena[n];` `n=(float*) malloc(n*sizeof(float));`
- ☐ `char *cadena;` `cadena=(char*) malloc(n*sizeof(char));`
- ☐ `char *cadena;` `cadena=(char*) malloc(n*char);`
- ☐ `char cadena[];` `cadena[n]=malloc();`