UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA	**Fundamentos de Informática 1º IIND** **Examen Práctico Febrero 2009**	13 - febrero - 2009

Nombre:	**Apellidos:**		
Grupo:	**Aula:**	**PC Nº:**	**Nº alumno:**

INSTRUCCIONES:

- Se pondrá el nombre y el número de PC en esta hoja, así como en la ficha (cabecera) del programa.
- Crear el proyecto del programa en **D:\temp\exam**
- El fichero del código fuente debe ser el número de matrícula, por ejemplo: **200801234.cpp**

- Las funciones sólo pueden tener un `return`.
- Sólo puede realizar operaciones de entrada/salida (`printf`, `scanf`, `gets`, etc,) en la función `main`().
- No puede utilizar `exit`, `continue`, ni `break` (salvo en la instrucción `switch`).
- No puede utilizar variables globales.

DURACIÓN: 1 hora 30 minutos

Problema:

Se quiere disponer de un programa para detectar "**Números Amigos**".

Dos números enteros son amigos, si cada uno de ellos es igual a la suma de los divisores del otro. Por ejemplo, 220 y 284 son amigos, ya que:

Suma de los divisores de 284 es 220: `1+2+4+71+142 = 220`
Suma de los divisores de 220 es 284: `1+2+4+5+10+11+20+22+44+55+110 = 284`

Escribir un programa que pida al usuario un conjunto de parejas de número y diga si son amigos o no. El programa debe realizar las siguientes operaciones:

- Declarar dos vectores de enteros para almacenar los números a analizar.
- Preguntar cuántas parejas de números se quieren analizar (comprobando que el tamaño sea válido por el tamaño máximo con el que se han declarado los vectores).
- Solicitar los vectores de enteros utilizando la función:
  `void SolicitarVector(int v[], int n);`
- llamar a la función `AnalizarVectores()` que reciba los dos vectores de enteros y un tercer vector para el resultado. El vector de resultados contendrá 1 ó 0.
- Para cada pareja de valores(`vec1[i]` y `vec2[i]`), la función `AnalizarVector()` llamará a la función `Amigos()`, que para determinar si son amigos o no. La función devolverá el valor 1 si los números son amigos y devolverá 0 si no son amigos.
- Finalmente el programa principal mostrará una tabla donde aparece la pareja de números y si son amigos o no. Ver ejemplo:

```
vec1 vec2 amigos
---------------
 220  284  si
  12  123  no
 284  220  si
 100  200  no
```

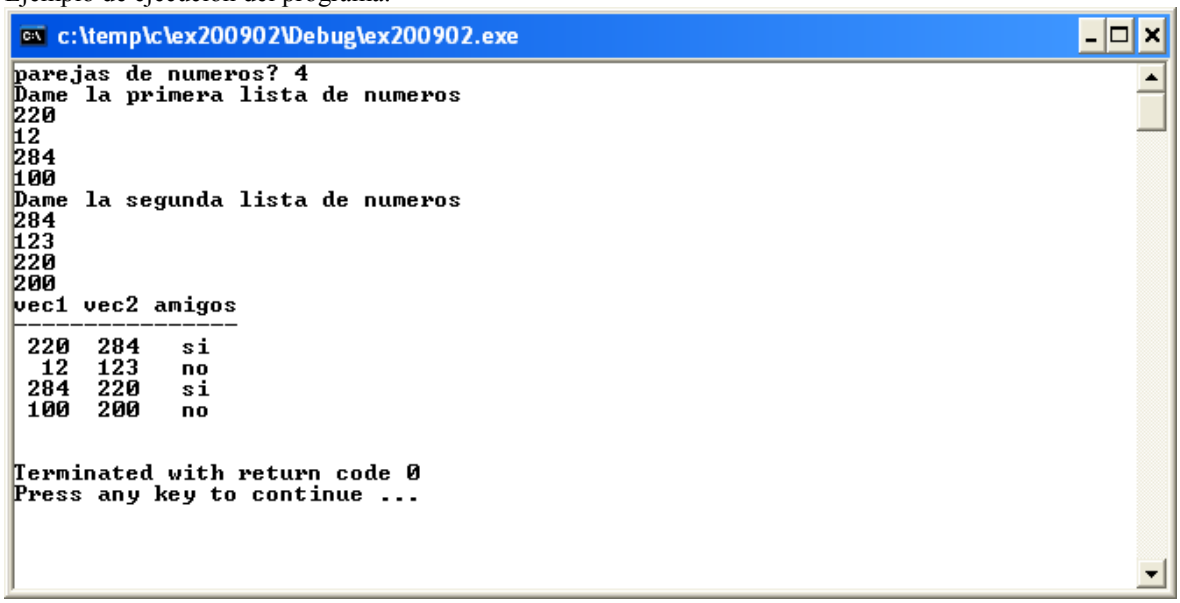
Se recomienda desarrollar el programa de la siguiente manera:

1º Desarrollar la función `Amigos()`, que es la que más puntos vale.
2º Desarrollar un pequeño programa principal que permita llamar a `Amigos()` para comprobar su correcto funcionamiento.
3º Implementar la función `SolicitarVector()`, la lectura de los vectores y mostrar la tabla.
4º Implementar la función `AnalizarVectores()`.

Calificación:

0,5 puntos - Estructura general: Ficha, includes, **prototipos**, declaración de los vectores, comentarios.
4 puntos - Función `Amigos()`.
2,5 puntos - Función `SolicitarVector()`, y la lectura de los dos vectores. Se valoran los controles para no desbordar el vector. Hay que ajustarse al prototipo de `SolicitarVector` que se propone.
1 puntos - Código para generar la tabla de resultados. Se valora que la tabla quede bien alineada.
2 puntos - Función `AnalizarVectores()`.

Ejemplo de ejecución del programa:



```c
/* Examen de programación 1ºIIND FEB/2009
   Rafael Palacios
*/
#include <stdio.h>
#define N 100
int Amigos(int a, int b);
void SolicitarVector(int v[], int n);
void AnalizarVector(int vec1[], int vec2[], int res[], int n);

int main(void)
{
    int v1[N],v2[N],v3[N];
    int n;
    int i;

    //Leer n
    do {
        printf("parejas de numeros? ");
        scanf("%d",&n);
        if (n>N || n<0) printf("Error, fuera de rango\n");
    } while(n>N || n<0);

    //Leer vectores
    printf("Dame la primera lista de numeros\n");
    SolicitarVector(v1,n);
    printf("Dame la segunda lista de numeros\n");
    SolicitarVector(v2,n);

    //Calculos
    AnalizarVector(v1,v2,v3,n);

    //Mostrar resultados
    printf("vec1 vec2 amigos\n");
    printf("----------------\n");
    for(i=0; i<n; i++) {
        printf("%4d %4d",v1[i],v2[i]);
        if (v3[i]==1) printf("   si\n");
        else printf("   no\n");
    }

    return 0;
}

int Amigos(int a, int b)
{
    int res; //resultado
    int i; //contador
    int suma_a;  //suma de los divisores de a
    int suma_b;  //suma de los divisores de b

    suma_a=0;
    for(i=1; i<a; i++) {
        if (a%i==0) suma_a+=i;
    }

    suma_b=0;
    for(i=1; i<b; i++) {
        if (b%i==0) suma_b+=i;
    }

    if (a==suma_b && b==suma_a) res=1;
    else res=0;

    return res;
}

void SolicitarVector(int v[], int n)
{
    int i; //contador

    for(i=0; i<n; i++) {
        scanf("%d",&v[i]);
    }
}

void AnalizarVector(int vec1[], int vec2[], int res[], int n)
{
    int i;

    for(i=0; i<n; i++) {
        res[i]=Amigos(vec1[i],vec2[i]);
    }
}
```