Aula		Fila		Col.	
Nombre		1er Ap.		2º Ap.	
					Firma:

NOTAS DE OBLIGATORIA LECTURA

1. Las soluciones de examen entregadas escritas a lápiz pueden ser anuladas
2. Sé ecológico: escribe por ambas caras de cada hoja de examen
3. Para la realización de este examen es obligatoria la presentación de identificación oficial actualizada con fotografía: carnet de estudiante, DNI, pasaporte o permiso de conducir formato tarjeta
4. La parte obligatoria durará 2h 30'
5. No se admite el uso de variables globales en todo el examen

Parte obligatoria. A resolver por TODOS los alumnos

1.- (**1 punto**) Completa los huecos del siguiente código (desde Línea 1 hasta Línea 5), para que dicho programa sea capaz de leer una matriz cuadrada de enteros y comprobar si el producto de los elementos de la diagonal es un número perfecto. Se dice que un número es perfecto cuando la suma de sus divisores propios es igual a dicho número. Los *divisores propios* de un número son los divisores menores que el número.
Ejemplo: 6 es un número perfecto ya que sus divisores propios son 1,2 y 3, y su suma da 1+2+3=6.

```
#include <stdio.h>
#define N 3
int divisible(int a, int b){  //función que devuelve 1 si a es divisible por b
    ____________________________________   //linea1
      return 1;
    else
      return 0;}

int perfecto(int n){ //función que devuelve 1 si n es perfecto y 0 si no lo es.
    int i, cont=0;
    for(i=1;i<n;i++)
    {
        if(divisible(n,i)==1)
          ____________________________________ //linea2
    }
    ____________________________________ //linea3
      return 1;
    else
      return 0;}

void lee(int m[N][N]){ //función que lee los elementos de una matriz NxN
    int i,j;
    printf("dime los elementos de la matriz:\n");
    for(i=0;i<N;i++)
       for(j=0;j<N;j++)
          ____________________________________  //linea4
}
int main(){
    int matriz[N][N];
    int producto=1, respuesta,i;
    lee(matriz);
    for(i=0;i<N;i++)
      producto=producto*matriz[i][i];
    ____________________________________   //linea5
    if(respuesta==1)
      printf("el producto de los elementos de la diagonal es un nº perfecto\n");
    else
      printf("el producto de los elementos de la diagonal no es un nº perfecto\n");
    getch(); return 0;}
```

2.- (**1.5 puntos**) Implementar un programa en C que pida al usuario un número entero (**n**) y calcule la suma del factorial de los **n** primeros números (desde el 1 hasta el **n**). Para ello, se debe implementar una función que calcule el factorial de un número y luego usarla.

$$\sum_{i=1}^{n} fact(i)$$

Por ejemplo, si el usuario da un valor de 5 para **n**, el programa debería sacar por pantalla 153, ya que el sumatorio sería fact(1)+fact(2)+fact(3)+fact(4)+fact(5) = 1+2+6+24+120 = 153.

3.- (**4 puntos**) La organización de la próxima vuelta ciclista de la UPV nos pide que desarrollemos un programa en lenguaje C que gestione los resultados obtenidos por los ciclistas en las diferentes etapas de la vuelta.

La vuelta consta de E etapas (identificadas por un número del 1 al E, ambos inclusive), y en ella compiten un total de C ciclistas (identificados por un número del 1 al C, ambos inclusive). Para cada etapa, se va a guardar **la posición** que cada ciclista ocupe en dicha etapa.

Las diferentes clasificaciones y premios de los que consta la vuelta son:

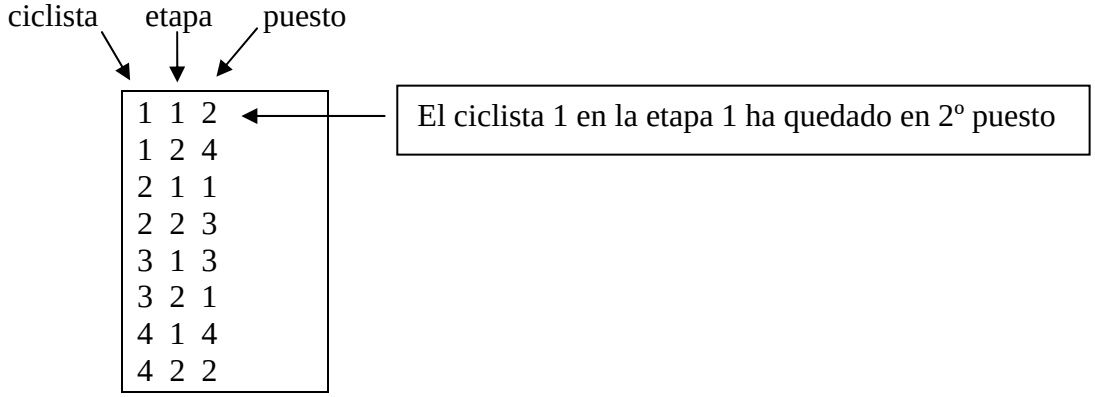
Ganador de Etapa: El que ha terminado primero en una etapa concreta.
Maillot verde de la Regularidad: El que ha finalizado con más puntos al final de todas las etapas. Los puntos otorgados en cada etapa son:

10 puntos al que finaliza primero una etapa.
5 puntos al que finaliza segundo una etapa
3 puntos al que finaliza tercero una etapa
1 punto a los que finalizan entre la posición cuarta y la décima
0 puntos a todos los demás

El programa mostrará por pantalla un menú compuesto por las diferentes opciones:

1. Mostrar el ganador de una etapa concreta.
2. Mostrar el ganador del maillot de la regularidad.
3. Salir.

En el momento de comenzar el programa y antes de mostrar el menú por pantalla, el programa leerá las posiciones de los ciclistas en todas las etapas desde el fichero *vueltaUPV.txt*. Cada línea del fichero consta de 3 números enteros: el primero indica el número de ciclista, el segundo indica la etapa y el tercero indica la posición de ese ciclista en esa etapa.
Ejemplo de fichero (suponiendo sólo 2 etapas y 4 ciclistas):



Se pide implementar las cuatro funciones indicadas en negrita para que el siguiente programa funcione correctamente:

```
#include <stdio.h>
#include <conio.h>

#define C 60   //Número de ciclistas
#define E 20   //Número de etapas

int main(){
  int op;
  int posiciones[C][E];

  leer_posiciones(posiciones);                    (1.5 puntos)

  do{
    op = menu();                                  (0.5 puntos)
    switch(op){
      case 1:
        mostrar_ganador_etapa(posiciones);        (0.5 puntos)
        break;

      case 2:
        mostrar_ganador_regularidad(posiciones);(1.5 puntos)
        break;

      case 3:
        printf("FIN DEL PROGRAMA\n");
        break;
    }
  }while(op != 3);
  getch();
  return 0;
}
```