

EXAMEN ENERO 2022/2023

Los **include** necesarios y otros requisitos ya viene dados en el examen. Además, ninguno de estas soluciones tiene por qué ser la más óptima, hay otras formas de solucionarlas.

```
6 //1. Comprueba que si dos vectores son o no iguales
7
8 bool soniguales (float v1[], float v2[], int n){
9     for (int i=0; i<n; i++){
10         if(v1[i]!=v2[i]){ //Si son distintos terminamos el bucle
11             return false;
12         }
13     }
14     return true;
15 }
```

```
6 //2. Rellena un vector con los factoriales de sus posiciones correspondientes
7
8 void factoriales(int fact[], int n){
9     fact[0]=1;
10    for (int i=1; i<n; i++){
11        fact[i]=fact[i-1]*i;
12    }
13 }
```

```

6 //3. Calcula la suma de los n primeros números primos
7
8 bool esprimo(int n){ //Creamos esta función para comprobar si un número es primo o no
9     if(n==0){return false;} //Especificamos que el 0 no es primo
10    else{
11        for(int i=2; i<=(n/2); i++){ //Hacemos que revise sus divisores desde el 2 hasta la
12            if(n%i==0){
13                return false;
14            }
15        }
16    }
17    return true;
18 }
19
20 int sumaPrimo(int n){
21     int num=0;
22     int cont=0;
23     int suma=0;
24     while(cont<n){ //Hacemos la suma de todos los valores que sean primos
25         num++;
26         if(esprimo(num)==true){
27             suma=suma+num;
28             cont++;
29         }
30     }
31     return suma;
32 }

```

```

6 //4. Comprueba si una cadena de caracteres es palíndroma o no. *Se lee igual del derecho y del
7     revés*
8 bool palindroma(string cad){
9     string cad2="";
10    for(size_t i=0; i<cad.size(); i++){
11        if(cad[i]!=' '){
12            cad2.push_back(cad[i]); //Creamos otra cadena igual a la primera
13        }
14    }
15    for(size_t i=0; i<cad2.size(); i++){
16        if(cad2[i]!=cad2[cad2.size()-1-i]){ //Comprobamos si el inicio y el final de la cadena es igual
17            return false;
18        }
19    }
20    return true;
21 }

```

```

6 // 5. Comprueba si dos matrices son iguales
7
8 bool matrizIgual (m1[10][10], m2[10][10]){
9     for(int i=0; i<10; i++){
10         for(int j=0; j<10; j++){
11             if(m1[i][j]!=m2[i][j]){ //Si algún valor no es igual, entonces no son iguales
12                 return false;
13             }
14         }
15     }
16     return true;
17 }
18

```

```

6 //6. Rellena un vector con el número más pequeño de cada columna
7
8 int mincolumna(m[10][10]){
9     int min[10]; //Creamos el vector en el que guardaremos los valores
10    for(int j=0; j<10; j++){ //Como son columnas, avanzaremos en función de las columnas
11        min[j]=m[0][j]; //Suponemos que el menor valor de cada columna es el primero
12        for(int i=0; i<10; i++){
13            if(min[j]>m[i][j]){ //Si algún valor es menor que el mínimo, lo cambiamos
14                min[j]=m[i][j];
15            }
16        }
17    }
18    return min;
19 }
20

```

```

6 //7. Cree una función que devuelva la posición del rectángulo de mayor área en un vector
  (obligatorio usar la estructura Rectangulo y la función mayorArea).
7
8 struct Rectangulo{
9     float lado1;
10    float lado2;
11 };
12
13 int mayorArea(Rectangulo v[], int n){ //v es el vector en el que se guardan los rectangulos
14     float area[n];
15     for(int i=0; i<n; i++){ //Con este bucle rellenos un vector con las áreas de los rectángulos
16         area[i]=v[i].lado1*v[i].lado2;
17     }
18     float maxarea=area[0]; //Suponemos que el rectángulo de mayor área se encuentra en la
    posición 0
19     int posicion=0;
20     for(int i=1; i<n; i++){ //Como ya tenemos como máxima la posición 0, empezamos por la 1
21         if(area[i]>maxarea){
22             maxarea=area[i];
23             posicion=i; //Si el rectángulo en esa posición tiene mayor área que el anterior, lo ponemos
    como mayor y cambiamos la posición
24         }
25     }
26     return posicion;
27 }
28

```

```

6 //8. En una tienda se coloca un sensor que calcula el tiempo que pasa cada persona en una
  tienda con 5 departamentos. Dada una matriz de 10x5 (siendo 5 el número de departamentos
  y 10 el número de personas que entran en cada departamento), en la que consta el tiempo de
  cada persona en cada departamento, devuelve el departamento en el que la gente pasa más
  tiempo.
7
8 int tiempomax(int m[10][5]){
9     int tiempo[5]; //En este vector guardaremos el tiempo pasado en cada departamento
10    int max=0; //En esta variable guardamos el tiempo mayor
11    int departamento; //En esta variable guardamos el departamento
12    for(int j=0; j<5; j++){
13        tiempo[j]=0; //Inicializamos la variable
14        for(int i=0; i<10; i++){
15            tiempo[j]=tiempo[j]+m[i][j]; //Sumamos el tiempo de las 10 personas en los
    departamentos
16        }
17        if(tiempo[j]>max){ //Comprobamos cuál es el mayor y le asignamos a la variable
    departamento su valor
18            max=tiempo[j];
19            departamento=j;
20        }
21    }
22    return departamento;
23 }

```

```
6 // 9. Realiza el histograma de un vector. *El histograma de un vector es otro vector en el que,
  en cada posición 0, 1, 2, etc. aparece el número de veces que aparece dicho número en el
  primer vector*
```

```
8 void histograma(int v[], int n, int histo[], int hist){ //vector v, tamaño n. Vector histo, tamaño
  hist
9   for(int i=0; i<hist; i++){ //Rellenaremos el vector histo hasta su tamaño
10    for (int j=0; j<n; j++){ //Buscamos los valores en v, de tamaño n
11      if(v[j]==i){ //Si el valor de v coincide con la posición de histo, le sumamos uno
12        histo[i]++;
13      }
14    }
15  }
16 }
17 }
```

```
6 //Dado un vector con puntos, calcular el punto medio de todos ellos (obligatorio usar la
  estructura Punto)
```

```
8 struct Punto{
9   int x=0;
10  int y=0;
11 };
12
13 Punto puntoMedio(Punto v[], int n){
14   Punto medio; //Creamos la variable en la que guardaremos el punto medio
15   for(int i=0; i<n; i++){
16     medio.x=medio.x+v[i].x; //Sumamos los puntos x
17     medio.y=medio.y+v[i].y; //Sumamos los puntos y
18   }
19   medio.x=medio.x/n; //Calculamos la media de todos ellos
20   medio.y=medio.y/n;
21   return medio; //Devolvemos el punto medio
22 }
23 }
```