

```

/**
 *
 */
void fijaValorAleatorio(int v[],int n,int max){
    srand(time(0)); //inicializa el generador de numeros aleatorios
    for(int i=0;i<n;i++){
        v[i]=rand()%(max+1); //genera un numero aleatorio entre 0 y 49
    }

}

/**
 *
 */
float max(float vector[],int n){
    float maximo=vector[0];
    for(int i=1;i<n;i++){
        if (maximo<vector[i]) maximo=vector[i];
    }
    return maximo;
}

/**
 *
 */
float min(float vector[],int n){
    float minimo=vector[0];
    for(int i=1;i<n;i++){
        if (minimo>vector[i]) minimo=vector[i];
    }
    return minimo;
}

/**
 *
 */
bool soniguales(float v1[],float v2[],int n){
    for(int i=0;i<n;i++){
        if (v1[i]!=v2[i]){
            return false;
        }
    }
    return true;
}

/**
 *
 */
void suma(float v1[],float v2[],float v3[],int n){
    for(int i=0;i<n;i++){
        v3[i]=v1[i]+v2[i];
    }
}

```

```

}
/**
 *
 */
void invierte(float v1[],float v2[],int n){

    for(int i=0;i<n;i++){
        v2[i]=v1[n-i-1];
    }
}

/**
 *
 */
int divisores(int val,int div[],int n){
    int ndiv=0;
    for(int i=1;i<val && ndiv<n;i++){
        if (val%i==0){
            div[ndiv]=i;
            ndiv++;
        }
    }
    return ndiv;
}

/**
 *
 */
void factoriales(int fact[],int n){
    fact[0]=1;
    for(int i=1;i<n;i++){
        fact[i]=i*fact[i-1];
    }
}

/**
 *
 */
int sum_odd(int v[],int n){
    int sum=0;
    for(int i=0;i<n;i++){
        if (v[i]%2==0){ sum=sum+v[i];}
    }
    return sum;
}

/**
 *
 */
int find(int val,int v[],int n){
    int nt=0;

```

```

    for(int i=0;i<n;i++){
        if (v[i]==val) {nt++;}
    }
    return nt;
}
/**
 *
 */
int find_indices(int val,int v[],int n,int indices[]){
    int nt=0;
    for(int i=0;i<n;i++){
        if (v[i]==val) {
            indices[nt]=i;
            nt++;
        }
    }
    return nt;
}
/**
 *
 */

void ordena(int v[],int n){
    for(int i=0;i<n-1;i++){
        for(int j=i+1;j<n;j++){
            if (v[i] < v[j]){
                int aux=v[i];
                v[i]=v[j];
                v[j]=aux;
            }
        }
    }
}

/**
 *
 */
bool find_in_sorted_vector(int val,int v[],int n){
    int inicio=0;
    int fin=n-1,central;
    while( inicio<=fin){
        central=(inicio+ fin)/2;
        if (v[central]==val) return true;
        else{
            if (v[central]<val){
                inicio=central+1;
            }
            else {
                fin=central-1;
            }
        }
    }
}

```

```
        central=inicio+ (fin-inicio)/2;  
    }  
    return false;  
}
```