

Teoría de las prácticas AAED

Cronometro:

```
cronometro(); //Constructor de la clase, crea un objeto tipo contador inactivo

void activar(); //activa el cronometro

void parar(); //para el cronometro

double tiempo(); // devuelve un double con el tiempo que ha pasado desde que se activo el cronometro.
```

Función Rand() incluida en la librería `cstdlib`, devuelve un entero entre 0 y `RAND_MAX` (número mayor aleatorio), devuelve una secuencia de números aleatorios, el cual está determinado por un valor inicial llamada semilla.

Función `srand()` recibe un único parámetro y se encarga de dar valor a la semilla para `rand()`.

ejemplo:

```
srand(unsigned time(NULL)); //Utilizamos time para que sean valores realmente aleatorio

a+rand()%(b-a)+1 //numero entero aleatorio entre los valores a y b

a+(rand()*((b-a)/RAND_MAX)); //Número float aleatorio entre a y b
```

Permutaciones Pseudoaleatorios, necesitaremos de la librería `<algorithm>`, utilizar `std`

funcion `random_shuffle(v,v+n)` //recibe el inicio y el final del vector, devuelve una permutación pseudoaleatoria de dicho vector.

función `next_permutation(v,v+n)` //recibe el inicio y el final del vector, crea la siguiente permutación del vector, devuelve true cuando la ultima permutaciones está ordenada

función `memcpy(aux,v,sizeof(v))` //Copia a un vector a otro vector

Medidas de tiempo de ejecución:

Medida Directa: El valor de la magnitud a medir se obtiene directamente, se observa directamente el valor numérico en un instrumento de medida adecuado

Medida Indirecta: Se obtiene mediante la medición de otras magnitudes relacionadas con ella, por lo que debe manipularse matemáticamente una o varias medidas directas

Ejemplo: Medimos n elementos y dividimos la medición entre n , para saber cuánto mide un solo elemento. Se utiliza cuando el sistema de medición no es el adecuado y para disminuir la posibilidad de error.

Medida adaptativa: Dentro de la medida indirecta, con la medida relativa se mide un numero finito de veces hasta que el error dado es el mínimo posible.

Medida directa:

```
Cronometro c;
c.activar();
//programa
c.parar()
double t = c.tiempo()
```

Medida Indirecta:

```
Cronometro c;
c.activar();
for(int i = 0; i < 100; i++){
    //programa
}
c.parar()
double t = c.tiempo()/100;
```

Medida adaptativa

```
Cronometro c;
c.activar();
int r = 0;
const double e_abs=0.01;
const double e_rel=0.001;
c.activar();
do{
    fragmento//
    r++;
}while(c.tiempo() < e_abs/e_rel + e_abs);
c.parar()
double t=c.tiempo()/r;
```

Donde $e_abs/e_rel + e_abs$ marcan el tiempo el cual debe ejecutarse para que el error sea el minimo posible, en este caso son $0.01/0.011 = 10$ segs