

## Práctica 2. Versión 2

Grado en Ingeniería Informática  
Grado en Ingeniería del Software  
Grado en Ingeniería de Computadores

Virginia Francisco Gilmartín  
Raquel Hervás Ballesteros  
Luis Hernández Yáñez  
Facultad de Informática  
Universidad Complutense



## Versión 2

---

Repartos de mazo del programa

- ✓ Separar en negras y rojas
- ✓ Separar en bajas y altas
- ✓ Repartir en tres montones

¿Dudas de funcionamiento?

- ✓ Demo ejecutable en el campus virtual



## Repartir negras y rojas

---

- ✓ `void repartirNegroRojo(const tMazo mazo, tMazo mazoNegro, tMazo mazoRojo)`
- 1. Usamos un índice diferente para señalar la posición en la que estamos de cada uno de los tres mazos. Todos deben de comenzar en 0
- 2. Recorremos el mazo original (mazo) de principio a fin:
  - ✓ Obtenemos el palo de la carta
  - ✓ Si el palo de la carta es negro, la añadimos al final de mazoNegro
  - ✓ Si el palo de la carta es rojo la añadimos al final de mazoRojo
- 3. Colocamos los centinelas al final de mazoNegro y mazoRojo



## Repartir bajas y altas

---

- ✓ `void repartirBajaAlta(const tMazo mazo, tMazo mazoBajas, tMazo mazoAltas)`
- ✓ Equivalente al anterior, pero separando las cartas según el número



## Repartir intercalando

- ✓ `void repartirIntercalando(const tMazo mazo, int enCuantosMazos, int queMazoPido, tMazo mazoNuevo)`
- ✓ Divide mazo en enCuantosMazos mazos y devuelve en mazoNuevo el mazo elegido (queMazoPido).
- ✓ Ejemplo de cartas elegidas en distintos ejemplos:



	11	26	41	27	25	14	0	45	50	36	52
	0	1	2	3	4	5	6	7	8	9	10
1º de 3	x			x			x			x	
3º de 4			x				x				
2º de 2		x		x		x		x		x	



## Repartir intercalando

- ✓ Sólo devuelve un mazo cada vez
- ✓ Si repartiendo el mazo en varias partes, queremos obtener todos los mazos resultantes, tendremos que llamar al procedimiento tantas veces como mazos haya.
- ✓ Ejemplo de repartir en tres mazos (mazo es el de la transparencia anterior):

mazoNuevo

`repartirIntercalando(mazo,3,1,mazoNuevo)`



`repartirIntercalando(mazo,3,2,mazoNuevo)`



`repartirIntercalando(mazo,3,3,mazoNuevo)`



## Repartir intercalando

---

- ✓ Usamos dos índices: uno para el mazo original (*mazo*) y otro para el mazo resultante del reparto que se ha pedido (*mazoNuevo*)
  - ✓ El del mazo original comenzará en *queMazoPido* - 1
  - ✓ El del mazo resultante en 0 (lo rellenamos desde el principio)
- ✓ Obtenemos cuantas cartas hay en *mazo*
- ✓ Recorremos *mazo* sin salirnos:
  - ✓ Copiamos la carta que corresponda de mazo en *mazoNuevo*
  - ✓ Incrementamos el índice del *mazo* original en tantas cartas como mazos se hayan pedido (*enCuantosMazos*)
  - ✓ Incrementamos en 1 el índice del *mazoNuevo*



## Programa principal

---

- ✓ Hay que añadir las siguientes opciones:
  - ✓ Separar entre cartas negras y rojas: Muestra el mazo, lo reparte en negras y rojas y muestra los mazos resultantes
    - Si el mazo del programa está vacío no hace nada e informa del error
  - ✓ Separar entre cartas bajas y altas: Muestra el mazo, lo reparte en bajas y altas y muestra los mazos resultantes
    - Si el mazo del programa está vacío no hace nada e informa del error
  - ✓ Repartir alternativamente en 3 montones: Muestra el mazo, lo reparte en 3 montones pidiendo el primero, muestra el primer montón, lo reparte en 3 montones pidiendo el segundo, muestra el segundo montón, lo reparte en 3 montones pidiendo el tercero y muestra el tercer montón
    - Si el mazo del programa está vacío no hace nada e informa del error



## Programa principal

---

- ✓ Para que el main sea más claro, hacemos un subprograma para cada nueva opción

```
switch (opcion){  
    ...  
    case 6: opcionRepartirNegroRojo(mazo);  
           break;  
    case 7: opcionRepartirBajaAlta(mazo);  
           break;  
    case 8: opcionRepartirMontones(mazo);  
           break;  
}
```



## Tratamiento de errores

---

- ✓ Los errores deben tratarse en los subprogramas de las opciones, no dentro de las funciones básicas
- ✓ Ninguno de los siguientes subprogramas debería usar cin o cout
  - ✓ void repartirNegroRojo(const tMazo mazo, tMazo mazoNegro, tMazo mazoRojo)
  - ✓ void repartirBajaAlta(const tMazo mazo, tMazo mazoBajas, tMazo mazoAltas)
  - ✓ void repartirIntercalando(const tMazo mazo, int enCuantosMazos, int queMazoPido, tMazo mazoNuevo)



# Tratamiento de errores. Ejemplo

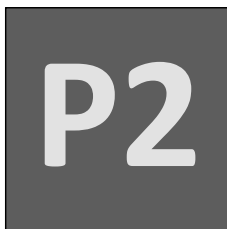
---

```
void opcionRepartirNegroRojo(const tMazo mazo) {  
    //El nombre del fichero se pide dentro de cargarMazo  
    if (mazoVacio(mazo))  
        cout << "El mazo está vacío, no hay nada que  
            repartir." << endl;  
    else{  
        ...  
    }  
}
```



## Fundamentos de Programación

---



## Práctica 2. Versión 2

Grado en Ingeniería Informática  
Grado en Ingeniería del Software  
Grado en Ingeniería de Computadores

Virginia Francisco Gilmartín  
Raquel Hervás Ballesteros  
Luis Hernández Yáñez

Facultad de Informática  
Universidad Complutense

