

# Computación I

## Preparación Control Recuperativo

Profesores: Nelson Baloian T.

Auxiliares: Sebastián Kreft C., Pedro Valenzuela S.

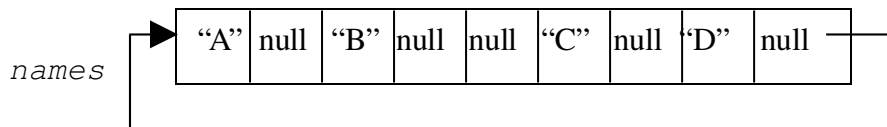
### Pregunta 1.

La leyenda cuenta que un grupo de guerreros fueron atrapados en una cueva por soldados Romanos. Ellos decidieron matarse entre ellos antes que ser capturados. Entonces formaron un círculo, se eligió un primer hombre para matarlo y a partir de él se iban a contar 3 lugares para elegir el siguiente, luego otros 3 lugares y así sucesivamente hasta que quedase uno solo, quien se iba a entregar a los enemigos. En esta pregunta vamos a desarrollar una serie de funciones para finalmente obtener el nombre del guerrero que sobrevive y debe entregarse.

Parte a) **3.0ptos.** Implemente una función con el siguiente encabezado:

```
static public int getNext(String[] x, int i)
```

Esta función debe retornar la primera posición en el arreglo *x* a partir de la posición *i* que contine un valor distinto de null. El arreglo debe recorrerse de forma circular. Esto quiere decir que si se está al final de arreglo y se quiere avanzar una posición hacia adelante se debe volver al comienzo. Esto puede hacerse si se incrementa el índice como  $i = (i+1) \% x.length$ . En el arreglo *names* que se muestra `getNext(names, 0)` retorna 2, `getNext(names, 2)` retorna 5 y `getNext(names, 7)` retorna 0.



Parte b) **3.0ptos.** Implemente una función con el siguiente encabezado:

```
static public String getSobreviviente(String[] x, int m)
```

Que entrega el nombre del último guerrero sobreviviente suponiendo que *x* contiene el nombre de todos los guerreros atrapados. El primero que se mata está en la primera posición. '*m*' corresponde al número de personas que se debe avanzar en el círculo para elegir al próximo que se debe matar (3 según la leyenda, pero en este caso lo queremos dejar variable). Para programar este último método le recomendamos seguir la siguiente estrategia:

- 1.- Para reflejar que el primer guerrero se suicida asignele 'null' al primer elemento del arreglo.
- 2.- avance '*m*' posiciones, sin contar los que ya han muerto (para eso use la función de la parte a) y elimínelo
- 3.- Volver al punto 2 hasta que quede un elemento en el arreglo distinto de null y retornar dicho valor.

### Pregunta 2.

Considere la siguiente clase:

Ejemplos	Resultado	Encabezamiento
Alumno a=new Alumno("Nelson Zilleruelo", 123456)	Crea un objeto de clase Alumno que contiene el nombre y el número de matrícula de un alumno	Alumno (String x, int y)
a.setPromedio(6.3 )	Registra el promedio de un alumno	void setPromedio (double x)

double i = a.getPromedio()	Retorna el promedio que estaba registrado para ese alumno. El valor retornado será -1 si no se había registrado nada aún	double getPromedio()
----------------------------	--	----------------------

Cree la clase AlumnoParciales que extienda la clase anterior para que pueda contener notas parciales. Estas notas se guardarán en un arreglo. En particular se le pide agregar reescribir los siguientes métodos:

Ejemplos	Resultado	Encabezamiento
AlumnoParciales a = new AlumnoParciales("Ricardo Baloian", 123456, 10)	crea un objeto de clase Alumno que contiene el nombre, el número de matrícula de un alumno y una capacidad para almacenar 10 notas parciales	Alumno (String x, int y, int z)
boolean t = a.setNota(6.3 )	agrega la nota que viene como parámetro a la lista de notas del alumno y retorna verdadero si aún había lugar. Si no había lugar se retorna false y no se agrega	boolean setNota (double x)
double i = a.getPromedio()	retorna el promedio para ese alumno que es el promedio de las notas parciales que se han colocado hasta ese momento. El valor retornado será -1 si no se había registrado nada aún	double getPromedio()

### Pregunta 3.

Se le pide que construya una aplicación con una interfaz gráfica que grafique las fuerzas aplicadas sobre un bloque. El tamaño del área de dibujo es de 450 pixels de ancho y 400 pixels de alto.

Para dibujar el bloque el usuario debe presionar un botón de etiqueta “Bloque”, con lo cual el programa dibujará un bloque de color negro cuyo vértice superior derecho está en el punto (145,130) y tiene 160 pixeles de ancho y 80 de alto. Para dibujar las fuerzas, el usuario debe ingresar la componente de la fuerza en x y la componente de la fuerza en y en pixeles. Para esto la aplicación tiene 2 elementos TextField. Después de ingresar los valores el usuario debe presionar el botón con etiqueta “Fuerza” para que la aplicación dibuje una línea de color rojo representando a la resultante de las fuerzas x e y ingresadas. Esto se logra dibujando una línea desde el punto (255,170) que corresponde al centro del bloque, hasta el punto (255+fx,170+fy), en donde fx y fy corresponden a los valores ingresados por el usuario para las fuerzas x e y. Además la aplicación deberá imprimir en un Label la magnitud de la fuerza en el área que se muestra en la figura. La magnitud de la fuerza se calcula como:  $\sqrt{f_x^2 + f_y^2}$ . Los elementos deben distribuirse como se muestra en la figura.

