Nombre\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ID\_\_\_\_\_\_\_\_Bonificación\_\_\_\_\_

* El archivo q cree se debe llamar parcial1\_ seguido de sus iniciales. Envíelo al correo: [**cesar.lopezg@upb.edu.co**](mailto:cesar.lopezg@upb.edu.co)
* Use para calcular los aleatorios: aleatorio\*(mayor-menor+1)+menor

1. **Valor: 30pts.** Diseñe un algoritmo y codifique una función en Java que simule un juego simple de cartas donde gana el jugador que saque la mayor carta. Las cartas van del 1 al 11. La función se llamará: Simular\_juego, recibirá como argumento de entrada el número de jugadores que deberá estar entre 2 y 6. Si el número de jugadores no está dentro del rango, deberá retornar en un texto -1 y así mismo si hay un error atrapado en el catch. Si el número de jugadores si está en el rango, generará una carta aleatoria para cada jugador y devolverá en un texto, el número de jugador que ganó y la puntuación de la carta que ganó.
2. **Valor: 20 pts.** Diseñe un algoritmo y codifique una función en Java que simule un cifrador para un carácter (solo minúsculas). La función se llamará cifrar no tiene datos de entrada y retorna un String de la forma caracter -> caracter cifrado.

La función calculará aleatoriamente un carácter en minúscula y lo devolverá cifrado siguiendo esta regla:

si es una a, la va a cambiar por una z, y va a retornar la cadena “a->z”

si es una b, la va a cambiar por una y, y va a retornar la cadena “b->y”

si es una m, la va a cambiar por una n, y va a retornar la cadena “m->n”

si es una n, la va a cambiar por una m, y va a retornar la cadena “n->m”

si es una z, la va a cambiar por una a, y va a retornar la cadena “z->a”

**NO puede usar switch-case,** por lo que debe pensar, usar la lógica y crear un patrón.

1. **Bonus Track.** Recargue la función anterior y va a tener como dato de entrada, el número de caracteres y va a retornar un String. Controlará que solamente va a trabajar con cadenas de 7 caracteres en minúscula. A diferencia de la anterior, acá se debe retornar un String de la siguiente forma:

Cadena Original: azasulx

Cadena Cifrada : zazhfoc

Por lo tanto, deberá usar la operación concatenar.

Rubrica:

Planteamiento de los elementos de la función: 20%

El algoritmo de la función correcto: 50%

Uso correcto de los tipos de dato: 10%

Escritura ordenada del programa: 10%

Manejo de errores 10%

Nombre\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ID\_\_\_\_\_\_\_\_Bonificación\_\_\_\_\_

* El archivo q cree se debe llamar parcial1\_ seguido de sus iniciales. Envíelo al correo: [**cesar.lopezg@upb.edu.co**](mailto:cesar.lopezg@upb.edu.co)
* Use para calcular los aleatorios: aleatorio\*(mayor-menor+1)+menor

1. Diseñe un algoritmo y codifique una función en Java que simule la liquidación de la nómina de una empresa. La función se llamará Liquidar\_nomina y recibirá como parámetro el número de empleados que debe ser mayor a 10. La empresa paga un salario entre 1.500.000 y 9.000.000 de pesos. A cada empleado se le debe calcular aleatoriamente, el salario en este rango. A cada empleado se le deduce el 14.5% de su salario para la seguridad social y el 1% para su fondo de ahorro. Al final la función debe retornar la siguiente información:

Valor total de los salarios de todos los empleados $########

Valor total del ahorro de todos los empleados: $########

Valor total de la seguridad social para todos los empleados: $########

Valor total a pagar a todos los empleados $########

Si ocurre un error, retorna un -1 en texto.

1. Diseñe un algoritmo y codifique una función en Java llamada Comparar\_cadena, que genere aleatoriamente, dos cadenas de texto con 6 caracteres concatenados en mayúscula para cada una, luego las compare y retorne booleano true si las dos cadenas son iguales y false si no lo son o si hay errores.

Nombre\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ID\_\_\_\_\_\_\_\_Bonificación\_\_\_\_\_

1. Diseñe un algoritmo y codifique una función en Java que simule la programación para la aplicación de una encuesta de salud en un sector de una ciudad. La función se llamará Programar\_encuesta y recibirá como argumento de entrada la cantidad de personas a las que se le aplican la encuesta que debe ser mayor a 50 personas y no va a pasar de las 300. A cada persona se le debe asignar aleatorimente, un grupo entre 1 y 6. Por una razón cualquiera, si el número es 6 se deberá volver a calcular el grupo aleatorio hasta que sea válido. (No se puede reducir el rango siempre será entre 1 y 6)

Si todo sale bien, la función retornará un mensaje como el siguiente:

Grupo 1: ### personas

Grupo 2: ### personas

Grupo 3: ### personas

Grupo 4: ### personas

Grupo 5: ### personas

Si ocurre un error debe retornar un –1 en un texto

1. Diseñe un algoritmo y codifique una función en Java que simule una serie de producción de una empresa y se llama Ejecutar\_produccion. Cada producto se imprime así: /0/. Cada 10 productos generados, se adiciona una caja con su respectivo número y salta la línea para generar otros 10 productos con su respectiva caja y así sigue el ciclo de producción. La función recibe como dato de entrada la cantidad de productos a generar que debe ser múltiplo de 10. La función retorna el mensaje de los 10 productos con su respectiva caja en una línea. Y tantas líneas como productos deba generar.

Por ejemplo, si la función recibe un 30, la salida se vería así:

/0//0//0//0//0//0//0//0//0//0/Caja1

/0//0//0//0//0//0//0//0//0//0/Caja2

/0//0//0//0//0//0//0//0//0//0/Caja3

Fin de la producción

Si ocurre un error debe retornar un –1 en un texto