



Página 1 de 14



1º Ejercicio	3
2º Ejercicio	4
3º Ejercicio	5
4º Ejercicio	6
5º Ejercicio	8
6º Ejercicio	9
7º Ejercicio	12
8º Ejercicio	13



## 1. Ejercicios

De acuerdo con las técnicas de testeo de caja negra, genera las clases de equivalencia, casos de prueba y valores límite para estos ejercicios.

1. Dado un programa que se utiliza para calcular el tramo del IRPF dado que se aplica a cada individuo ([@Xokas](#)), crea las clases de equivalencia, casos de prueba y valores límite. (1 p)

Base Liquidable Hasta	Cuota íntegra	Resto base liquidable	Tipo estatal	Tipo autonómico	Tipo total
0,00 €	0,00 €	12.450,00 €	9,50%	9,50%	19,00%
12.450,00 €	1.182,75 €	7.750,00 €	12,00%	12,00%	24,00%
20.200,00 €	2.112,75 €	15.000,00 €	15,00%	15,00%	30,00%
35.200,00 €	4.362,75 €	24.800,00 €	18,50%	18,50%	37,00%
60.000,00 €	8.950,75 €	En adelante	22,50%	22,50%	45,00%

Nota: Dicho programa sólo recibe un entero, que será el dinero que ingresa un trabajador, y devolverá el porcentaje que se retiene.

Condición de Entrada	Tipo	Clase Equivalencia Válida	Clase Equivalencia No Válida
Salario	Rango	Salario = 0	Salario <= 0 Salario >= 0
Salario	Rango	Salario > 1 Salario <= 12.450	Salario < 1 Salario > 12.450
Salario	Rango	Salario > 12.451 Salario <= 20.200	Salario < 12.451 Salario > 20.200
Salario	Rango	Salario > 20.201 Salario <= 35.200	Salario < 20.201 Salario > 35.200
Salario	Rango	Salario > 35.201 Salario < 60.000	Salario < 35.201 Salario > 60.000



Nº de casos	Limites	Valores de Prueba	Resultados Validos	Resultados No Validos
1	0€	100€	19%	≠19%
2	12.450€	15.000€	24%	≠24%
3	20.200€	25.000€	30%	≠30%
4	35.200€	55.000€	37%	≠37%
5	+60.000€	70.000€	45%	≠45%
6	0€	-100€	Ninguno	No es un número entero

2. Se os ha encomendado la función ***int encontrarValorArray(int[] arrayBuscar, int valor)***, que devolverá la posición del elemento ***valor*** en ***arrayBuscar***, si existe, y -1 en caso contrario. Usando la técnica de las clases de equivalencia y de los valores límite, generar los casos de pruebas asociado para asegurarnos de su correcto funcionamiento. (1 p)

Condición de entrada	Tipo	Clase de equivalencia valida	Clase de equivalencia no valida
Vacío	Número entero	-1	Números enteros
Con elementos que no se llaman valor	Número entero	-1	Números enteros
Con elemento llamado valor	Número entero	Posición de valor: 0 - ∞	-1



3. Una empresa de BBDD os pide que hagáis un test de caja negra para comprobar el registro de un usuario, en caso concreto la introducción de una contraseña. Según lo que cuenta la empresa, el password es de tipo String, y tiene que tener entre 5 y 10 caracteres. Además, tiene que incluir por lo menos una mayúscula y un dígito, si no el sistema tiene que devolver un error 500, al igual que si se introduce una contraseña que no respete la longitud.

Como es una aplicación seria, si incluye la palabra 'hitler' en mayúsculas o en minúsculas en alguna parte de la cadena, deberá devolver un error 800.

De acuerdo a este enunciado, crea las clases de equivalencia, casos de prueba y valores límite.

(1.5 p)

Condición de entrada	Tipo	Clase de equivalencia valida	Clase de equivalencia no valida
Contraseña correcta	String	Contraseña valida	Error 500 Error 800
Contraseña con el numero de caracteres incorrecto	String	Error 500	Contraseña < 5 Contraseña > 10
Contraseña no tiene mayúscula y un dígito	String	Error 500	Contraseña no tiene mayúsculas Contraseña no tiene dígitos
Contraseña contiene la palabra "hitler" en mayúsculas o minusculas	String	Error 800	Contraseña valida



Nº caso	Valores	Resultados válidos	Resultados no válidos
1	Pep4	Error 500	Error 800 Contraseña valida
2	Hitler	Error 800	Error 500 Contraseña valida
3	MyPassw0rd	Contraseña valida	Error 500 Error 800
4	MiContras3ñ4EsS3gur4	Error 500	Error 800 Contraseña valida
5	Contraseña	Error 500	Error 800 Contraseña valida
6	123456789	Error 500	Error 800 Contraseña valida

4. Tinder quiere que hagas un test de una nueva interfaz gráfica de búsqueda que quieren introducir en una nueva actualización para hombres. Se podrán introducir 3 campos: un campo género (que tendrá un check para elegir hombres y mujeres), la edad del solicitante y la edad de la gente a la que busca.

Se pueden elegir hombres, mujeres o ambos, pero se debe elegir un valor siempre.

La edad del que solicita y solicitante debe ser SIEMPRE mayor que 18 y menor que 100, en el caso en el que alguno de estos valores sea menor o mayor que dicho rango la aplicación debe devolver un error.

Si la edad de a quien busca es menor a 18 la aplicación debe devolver un ERROR con amenaza de llamar a la policía.

Crea las clases de equivalencia, casos de prueba y valores límite, teniendo en cuenta cuáles podrían ser las clases inválidas en este ejercicio. (1,5 p)



Condición de entrada	Tipo	Clase de equivalencia valida	Clase de equivalencia no valida
Todos los datos validos	Formulario	Bien	Error
Genero vacío	Formulario	Error	Hombre, mujer, ambos
Edad mayor de 100	Formulario	Error	Bien
Edad menor de 18	Formulario	Error con amenaza de llamar a la policía	Bien
Edad vacía	Formulario	Error	Bien

Nº caso	Valores	Resultados válidos	Resultados no válidos
1	Validos	Bien	Error
2	Genero vacío	Error	Bien
3	Edad < 18 Edad > 100	Error Error con amenaza de llamar a la policía	Bien



5. Sois alguien que trabaja para McFit, y os encomiendan que hagáis tests para probar una nueva máquina elíptica con parte digital, la cual modifica la velocidad de acuerdo a estos dos parámetros. La máquina toma dos enteros como entrada: la altura y el peso, si mide menos de 160 cm la velocidad será de 5 Km/h, y si mide más de 200 cm la velocidad será de 15 Km/h. En el caso en el que la altura se encuentre en el rango de 160 y 200 cm, entrará en juego el peso. Si pesa menos de 60kg, la velocidad será de 8 km/h, si pesa entre 60 y 100 kg la velocidad será de 10 KM/h, y si pesa más de 100 kg la velocidad será de 13 km/h. Crea las clases de equivalencia, casos de prueba y valores límite, teniendo en cuenta cuáles podrían ser las clases inválidas en este ejercicio. (2 p)

Condición de entrada	Tipo	Clase de equivalencia valida	Clase de equivalencia no valida
Datos validos	Numero entero	Velocidad correspondiente	
Altura menos de 160cm	Numero entero	5 km/h	Otras velocidades
Altura mas de 200cm	Numero entero	15 km/h	Otras velocidades
Altura entre 160cm y 200cm	Numero entero	Menos de 60kg = 8 km/h Entre 60kg y 100kg = 10 km/h Mas de 100kg = 13 km/h	Otras velocidades
Vacio	Vacio	No valido	
Texto en lugar de números	String	No valido	
Números negativos	Numero entero	No valido	





Nº caso	Valores	Resultados válidos	Resultados no válidos
1	Altura 150cm	5 km/h	Otras velocidades
2	Altura 180cm	Menos de 60kg = 8 km/h Entre 60kg y 100kg = 10 km/h Mas de 100kg = 13 km/h	Otras velocidades
3	Altura 210cm	15 km/h	Otras velocidades
4	Vacio	No valido	
5	Altura Pepe	No valido	
6	Altura -150cm	No valido	

6. Eclipse necesita a algún programador que le testee la web que tiene preparado para poder realizar donaciones: una vez hayas descargado el IDE, te da la posibilidad de realizar una donación, incluyendo tu código postal y tu número de teléfono. La donación deberá ser un número de dos dígitos, pero no puede ser 00. Si intentas ir de gracioso y poner un número negativo de dos dígitos, devolverá un mensaje muy agresivo.

El código postal es un número de 5 dígitos, y el teléfono irá precedido de un código (+34, + 31) y 9 dígitos.



Condición de entrada	Tipo	Clase de equivalencia valida	Clase de equivalencia no valida
Datos validos	Números enteros	Bien	Error
Donación es 00	Números enteros	Error	Bien
Donación vacia	Vacia	Error	Bien
Donación negativa	Numero negativo	Error con mensaje agresivo	Bien
Codigo postal vacio	Vacio	Error	Bien
Codigo postal no valido	Numero entero	Error	Bien
Codigo de teléfono vacio	Numero entero	Error	Bien
Codigo de teléfono no valido	Numero entero	Error	Bien
Numero de teléfono vacio	Vacio	Error	Bien
Numero de teléfono no valido	Numero entero	Error	Bien



Nº caso	Valores	Resultados válidos	Resultados no válidos
1	Donacion 10 Codigo postal 38000 Telefono +34 123456789	Bien	Error
2	Donacion 00	Error	Bien
3	Donacion -10	Error con mensaje agresivo	Bien
4	Codigo postal 111	Error	Bien
5	Codigo postal 989898989	Error	Bien
6	Codigo de teléfono +1	Error	Bien
7	Telefono 12345	Error	Bien



7. Un programa se encarga de calcular la suma de los primeros **num** enteros siempre que dicha suma sea menor que **maxint**. Los parámetros que recibe son **num** y **maxint**, y en el caso el que la suma sea mayor que **maxint** devolverá un error. Si **num** es negativo entonces se tomará su valor absoluto.

Crea las clases de equivalencia, casos de prueba y valores límite, teniendo en cuenta cuáles podrían ser las clases inválidas en este ejercicio (2 p)

Condición de entrada	Tipo	Clase de equivalencia valida	Clase de equivalencia no valida
num es un número entero y la suma de los primeros num enteros es menor que maxint	Numero entero	Suma de los primeros num enteros	Error
num es un número entero y la suma de los primeros num enteros es mayor que maxint	Numero entero	Error	Suma de los primeros num enteros
num es un número negativo y la suma de los primeros	Numero	num = su valor absoluto	Error



Nº caso	Valores	Resultados válidos	Resultados no válidos
1	Num = 1 Maxint = 100	Suma de los primeros num enteros	Error
2	Num = 1000 Maxint = 500	Error	Suma de los primeros num enteros
3	Num = -1	Num es su valor absoluto. Suma de los primeros num enteros	Error

8. Un sistema lee tres valores positivos de la entrada estándar. Los tres valores A, B y C se interpretan como cada una de las longitudes de los lados de un triángulo. El sistema luego imprime un mensaje en la salida estándar diciendo si el triángulo es escaleno, isósceles o equilátero, [si es que se puede formar un triángulo](#). De acuerdo a este enunciado, crea las clases de equivalencia, casos de prueba y valores límite. (1 p)

Condición de entrada	Tipo	Clase de equivalencia valida	Clase de equivalencia no valida
Los tres valores son iguales	Int	Triangulo Equilatero	No se puede formar un triangulo
Dos valores son iguales	Int	Triangulo Isosceles	No se puede formar un triangulo
Los tres valores son diferentes	Int	Triangulo Escaleno	No se puede formar un triangulo
Al menos un valor es negativo	Int	No se puede formar un triangulo	-



Nº caso	Valores	Resultados válidos	Resultados no válidos
1	5, 5, 5	Triangulo Equilatero	No se puede formar un triangulo Triangulo Isosceles ni Escaleno
2	5, 5, 7	Triangulo Isosceles	No se puede formar un triangulo Triangulo Equilatero ni Escaleno
3	3, 4, 5	Triangulo Escaleno	No se puede formar un triangulo Triangulo Isosceles ni Equilatero
4	-1, 4, 5	No se puede formar un triangulo	-