**Trabajo práctico nro. 1**

|  |  |  |
| --- | --- | --- |
|  | **Asignatura:** | |
|  | |
| **Cursado:** | **Horas** **semanales**: |
|  | **Horas semestrales:** |
| **Carrera**: *Tecnicatura Universitaria en Programación* | **Nivel (Año):** |
| **Ciclo Lectivo: 2023** |

**Integrantes de la Cátedra:**

* **DOCENTES:**
* **Tema:**
* **Enunciados:**
* **Ejercicio sobre identificadores: indica qué identificadores Java son válidos.**

Indica cuáles de los siguientes identificadores son válidos en Java. Si el identificador no es válido explica porqué no lo es.



* Valido
* No valido porque inicia con un digito
* Valido
* No valido porque es una palabra reservada
* Valido
* Valido
* No valido porque contiene un espacio
* Valido
* No valido porque contiene el carácter “-“
* Valido
* No valido porque contiene el carácter “ ‘ “
* Valido
* No valido porque contiene el carácter especial “#”
* Valido
* No valido porque inicia con el carácter “%”
* Valido
* Valido
* Valido
* Valido
* No valido porque contiene el carácter “.”
* Valido
* No valido porque contiene el carácter “?”
* Valido
* Valido
* Valido
* Valido
* Valido
* Valido
* No valido porque inicia con digito
* No valido porque contiene el carácter “&”
* No valido porque inicia con digito
* No valido porque es una palabra reservada
* **Ejercicio sobre literales: indica qué literales Java son válidos**.

Indica cuáles de los siguientes literales son válidos en Java. Si el literal es válido indica además de qué tipo es (int, double, long, etc) y el sistema de numeración en el que está escrito (decimal, binario, octal, hexadecimal). Si el literal no es válido explica porqué no lo es.



* Valido. Tipo Double. Base decimal
* Valido. Tipo Double. Base decimal
* Valido. Tipo Double. Base decimal
* Valido. Tipo Double. Base decimal
* Valido. Tipo Entero. Base decimal
* No valido porque contiene el carácter “\_”
* No valido porque el exponente no puede tener decimales
* No valido porque contiene un espacio
* Valido. Tipo Entero. Base octal
* No valido porque contiene las letras CDF y no son validas en el sistema octal
* Valido. Tipo Entero. Base hexadecimal
* Valido. Tipo Entero. Base hexadecimal
* Vaido. Tipo Long. Base decimal
* No valido porque contiene el carácter “\_” y no se puede separar los caracteres 0B
* Valido. Tipo Entero. Base octal
* Valido. Tipo Entero. Base octal
* No valido porque solo puede contener un “.”
* Valido. Tipo Float. Base decimal
* Valido. Tipo entero. Binario
* Valido. Tipo Long. Base decimal
* Valido. Tipo Entero. Base hexadecimal
* Valido. Tipo Long. Base hexadecimal
* No valido porque inicia con el carácter “\_”
* No valido porque si fuese binario debería empezar con 0B
* Valido. Tipo Entero. Base hexadecimal
* Valido. Tipo Double. Base decimal
* No valido porque falta el valor del exponente entre los caracteres E y F
* Valido. Tipo Float. Base decimal
* No valido porque en un tipo float o double no puede contener el carácter “\_”
* No valido porque comienza por 0B
* Valido. Tipo Entero. Base hexadecimal
* Valido. Tipo Long. Base hexadecimal
* No valido porque si fuese hexadecimal empezario por 0X
* Valido. Tipo Entero. Base octal
* Valido. Tipo Double. Base decimal
* Valido. Tipo Double. Base decimal
* No valido porque en un tipo float o double no puede contener el carácter “\_”
* Valido. Tipo Double. Base decimal
* No valido porque falta el valor del exponente
* No valido porque un numero binario solo puede contener 0 y 1
* Valido. Tipo Entero. Base decimal
* Valido. Tipo Entero. Base hexadecimal
* Valido. Tipo Entero. Base hexadecimal
* Valido. Tipo Double. Base decimal
* No valido porque si fuese binario debería empezar por 0B
* No valido porque los decimales solo pueden contener un “.”
* Valido. Tipo Float. Base decimal
* No valido porque G no es un carácter valido para un numero hexadecimal
* **Ejercicio sobre literales de tipo char.**

Indica cuáles de los siguientes literales de tipo char son válidos en Java. Si el literal no es válido explica el motivo.



* Valido
* Valido
* Valido
* No valido porque un literal tipo char debe estar formado por un solo carácter a menos que sea una secuencia de escape
* Valido
* No valido porque un literal tipo char debe estar formado por un solo carácter a menos que sea una secuencia de escape
* No valido porque el literal tipo char debe ir entre comillas simples
* Valido
* No valido porque un literal tipo char debe estar formado por un solo carácter a menos que sea una secuencia de escape
* Valido
* No valido porque un literal tipo char debe estar formado por un solo carácter a menos que sea una secuencia de escape
* Valido
* Valido
* Valido
* No valido porque el literal tipo char debe ir entre comillas simples
* No valido porque el literal tipo char debe ir entre comillas simples
* Valido
* Valido
* Valido
* Valido
* **Ejercicio sobre literales de tipo String.**

Indica cuáles de los siguientes literales de tipo String son válidos en Java. Si el literal no es válido explica el motivo.



* No valido porque los Strings van entre comillas dobles
* Valido
* No valido porque faltan las comillas de cierre
* Valido
* Valido
* Valido
* Valido
* Valido
* No valido porque ambas comillas deben ser dobles
* Valido
* No valido porque el string debería contener las comilas interiores como secuencia de escape
* No valido porque los Strings van entre comillas dobles
* No valido porque los Strings van entre comillas dobles
* Valido
* Valido
* Ejercicio sobre declaraciones de variables

**Ejercicio 1**: Escribe la declaración para cada una de las variables y asígnales un valor inicial en la propia declaración de variable.



* int p = 5, q = 3 ;
* float x = 26.01F, y = 45F, z = 3.4F ;
* char a = ‘j’, b = ‘q’, c = ‘k’ ;
* double raiz1 = 45.32, raiz2 = 32.6 ;
* long contador = 0 ;
* shor indicador = 0 ;
* int índice = 3 ;
* double precio = 4.3, precioFinal = 9.2 ;
* char car1 = ‘l’, car2 = ‘m’ ;
* byte valor = 3 ;
* boolean primero = true, ultimo = false ;
* string nombre = “Valentin” ;

**Ejercicio 2**: Escribe la declaración más apropiada para cada una de las siguientes variables. El nombre de cada una indica el tipo de dato que contendrá y servirá para determinar el tipo de dato más adecuado en la declaración de la variable.



* int edad;
* String códigoPostal;
* double altura;
* char genero;
* string nombre;
* int númeroDeHijos;
* double iva;
* int tallaCamisa;
* double peso;
* double precio;
* string alumnoRepetidor;
* string mensaje;
* char letra;
* int mayorEdad;
* int minutos;
* int días;
* string matriculaCoche;
* int contador;
* boolean mayorDeEdad;
* string tallaCamiseta;

6. Convertir expresiones algebráicas en expresiones algorítmicas

**Ejercicio:** Convierte en expresiones algorítmicas las siguientes expresiones algebráicas. Coloca paréntesis solamente donde sean necesarios.



* 3/2 + 4/3
* 1/(x-5) – (3\*x\*y)/4
* 1/2 + 7
* 7 + 1/2
* a\*a/(b-c) + (d-e)/(f - g \* h/j)
* m/n + p
* m + n/(p – q)
* a\*a/(b\*b) – c\*c/(d\*d)
* (m + n/p)/(q – r/s)
* (3\*a + b)/(c – (d + 5\*e)/(f + g/(2\*h)))
* (a\*a + 2\*a\*b + b\*b)/(1/x\*x )) + 2

7. Convertir expresiones algorítmicas en expresiones algebráicas

**Ejercicio:** Convierte en expresiones algebráicas las siguientes expresiones algorítmicas.





8. Ejercicio sobre operadores aritméticos

**Ejercicio 1:** Si a, b y c son variables enteras con valores a=8, b=3, c=-5, determina el valor de las siguientes expresiones aritméticas:



* 6
* 45
* 2
* 2
* -1
* 3
* -4
* 0
* -1
* -16
* 18
* 0
* 0
* 15

**Ejercicio 2:**Si x, y, z son variables de tipo double con valores x= 88, y = 3.5, z = -5.2, determina el valor de las siguientes expresiones aritméticas. Obtén el resultado de cada expresión con un máximo de cuatro decimales.



* 86.3
* 286.6
* 25.1428
* 0.5
* 51.7647
* 19.9428
* 205.3333
* 16.7619
* 1.1999
* 308.0
* 93.1999
* 0.2000
* 86.0
* 94.9

**Ejercicio 3.**Si c1, c2 y c3 son variables de tipo char con valores c1=’E’, c2=’5’, c3=’?’, determina el valor numérico de las siguientes expresiones aritméticas. Para resolverlo necesitas saber el valor numérico correspondiente a esos caracteres según la tabla ASCII:

’E’      69

’5’      53

’?’      63

Consulta en la tabla ASCII los valores numéricos del resto de caracteres que aparecen en las operaciones.



* 70
* 79
* 51
* 3
* 98
* 3
* 100
* 63
* 159
* 2703

9. Determinar el tipo del resultado en operaciones aritméticas

En aquellas expresiones aritméticas en las que intervienen operandos de distinto tipo (int, double, char, etc.) java convierte el tipo de los operandos que intervienen al tipo del operando de mayor precisión y este será el tipo del resultado obtenido en la operación.

Esta conversión de tipos se realiza de forma temporal, solamente para el cálculo de la operación. Los tipos originales de los operandos que intervienen siguen siendo los mismos después de la operación.

Debemos tener en cuenta que cuando se realiza la conversión temporal de tipos, los operandos de tipo short, byte y char se convierten a int.

Teniendo en cuenta todo esto, realiza el siguiente ejercicio.

**Ejercicio**: A partir de las siguientes declaraciones de variables:



Determina cuál es el tipo de dato del resultado de las siguientes expresiones aritméticas:



* Int
* Float
* double
* double
* float
* int
* double
* int
* double
* int
* int
* int
* double
* long
* int
* double
* double
* float
* long
* float
* double
* int
* int
* int
* long
* double

10. Ejercicio sobre operadores lógicos

**Ejercicio 1:** Si X, Y y Z son variables de tipo boolean con valores X = true, Y = false, Z = true, determina el valor de las siguientes expresiones lógicas:



* True
* True
* True
* False
* True
* True

**Ejercicio 2:**Si W, X, Y y Z son variables de tipo boolean con valores W = false, X = true, Y = true, Z = false, determina el valor de las siguientes expresiones lógicas:



* False
* True
* True
* True
* True
* False

Para resolver ejercicios con operadores lógicos tienes que conocer las tablas de verdad de estos operadores:

|  |  |  |
| --- | --- | --- |
| Operador AND | Operador OR | Operador NOT |
|  |  |  |

Como puedes observar:

* El operador lógico && (AND) da como resultado **true** cuando ambos operandos son true, en cualquier otro caso el resultado es false.
* El operador || (OR) da como resultado **false** cuando ambn cualquier otro caso el resultado es true.
* El operador ! (NOT) cambia el valor lógico del operando.

También debemos tener en cuenta para resolver estos ejercicios cuál es la precedencia de los operadores lógicos en Java:

|  |
| --- |
| Precedencia de los operadores lógicos |
|  |

11. Ejercicios sobre operadores relacionales

En estos ejercicios practicaremos con los operadores Java relacionales. En algunos de los ejercicios, los operadores relacionales aparecerán en expresiones en los que también intervienen el resto de operadores Java: operadores aritméticos, lógicos y los operadores incremento y decremento.

**Ejercicio:**

A partir de las siguientes declaraciones de variables:

c = 99 d = 100

Determina el valor de las siguientes expresiones:



* False
* False
* True
* False
* True
* True
* True
* False
* False
* True
* True
* True
* True
* False
* True
* 0.01 + False
* True
* True
* False
* False
* True

12. Ejercicio con los operadores incremento y decremento: ++ y --.

**Ejercicio 1:** Si x es una variable de tipo int con valor 10, determina qué se muestra por pantalla cuando se ejecutan las siguientes instrucciones:



|  |  |  |
| --- | --- | --- |
|  | Valor de X | Salida por pantalla |
| int x = 10; | 10 |  |
| System.out.println(x); |  | 10 |
| x++; | 11 |  |
| System.out.println(x); |  | 11 |
| System.out.println(++x); | 12 | 12 |
| System.out.println(x++); | 13 | 12 (primero se muestra el valor, luego se hace la suma) |
| System.out.println(x); |  | 13 |
| System.out.println(x++); | 14 | 14 |
| System.out.println(++x); | 15 | 14 |
| System.out.println(++x); | 16 | 16 |
| ++x; | 17 |  |
| x++; | 18 |  |
| System.out.println(++x); | 19 | 19 |
| System.out.println(x++); | 20 | 19 |
| System.out.println(++x); | 21 | 21 |

**Ejercicio 2:**Dadas las variables A y B de tipo char,calcula qué se muestra por pantalla cuando se ejecutan las siguientes instrucciones**:**



|  |  |  |  |
| --- | --- | --- | --- |
|  | Valor de A | Valor de B | Salida por pantalla |
| char A = 'c'; | ‘c’ |  |  |
| char B; |  |  |  |
| System.out.println(A++); | ‘d’ |  | ‘c’ |
| System.out.println(A++); | ‘e’ |  | ‘d’ |
| System.out.println(++A); | ‘f’ |  | ‘f’ |
| B = --A; | ‘e’ | ‘e’ |  |
| System.out.println(++A); | ‘f’ |  | ‘f’ |
| A++; | ‘g’ |  |  |
| --B; |  | ‘d’ |  |
| System.out.println(B++); |  | ‘e’ | ‘d’ |
| System.out.println(++B); |  | ‘f’ | ‘f’ |
| System.out.println(++A); | ‘h’ |  | ‘h’ |
| System.out.println(B--); |  | ‘e’ | ‘f’ |
| System.out.println(A); |  |  | ‘h’ |
| System.out.println(B); |  |  | ‘e’ |

**Ejercicio 3:**Un programa Java contiene las siguientes declaraciones y asignaciones iniciales:

**int i = 1, j = 1, k = 1;**

Determina el valor final de las tres variables en cada una de las siguientes instrucciones. Las instrucciones son independientes unas de otras, es decir, el valor inicial de las variables en cada instrucción es i = 1, j = 1, k = 1.



|  |  |  |  |
| --- | --- | --- | --- |
|  | Valor de i | Valor de j | Valor de k |
| int i = 1, j = 1, k = 1; | 1 | 1 | 1 |
| a) i = ++j; | 2 | 2 | 1 |
| b) i = k++; | 1 | 1 | 2 |
| c) i = k + ++j; | 3 | 2 | 1 |
| d) i = i + j++; | 2 | 2 | 1 |
| e) i = j + ++k; | 3 | 1 | 2 |
| f) i = ++j + k++; | 3 | 2 | 2 |
| g) j = k-- + --i; | 0 | 1 | 0 |
| h) i = k + 1 + ++j; | 4 | 2 | 1 |
| i) i = ++i + --j + k--; | 3 | 0 | 0 |
| j) k = j-- + ++k; | 1 | 0 | 3 |

13. Ejercicios para practicar la salida de datos por pantalla: System.out.print y System.out.println – I

Relación Nº 1: Ejercicios 1 y 2  
  
Empezamos con una serie de ejercicios para aquellos que están comenzando desde cero con el lenguaje Java y tampoco tienen experiencia con ningún otro lenguaje de programación.

En estos ejercicios básicos inicales se realizan las siguientes instrucciones: declarar variables, asignarles un valor, operar con ellas y mostrar resultados por pantalla.

Estos ejercicios son todos de estructura secuencial, es decir, no hay condiciones ni bucles. Tampoco se lee nada por teclado. El objetivo es familiarizarse con la declaración de variables y practicar la salida por consola utilizando los métodos print y println.

**Ejercicio básico inicial 1**:

Escribe un programa Java que realice lo siguiente: declarar una variable N de tipo int, una variable A de tipo double y una variable C de tipo char y asigna a cada una un valor. A continuación muestra por pantalla:

* El valor de cada variable.
* La suma de N + A
* La diferencia de A – N
* El valor numérico correspondiente al carácter que contiene la variable C.

Si por ejemplo le hemos dado a N el valor 5, a A el valor 4.56 y a C el valor ‘a’, se debe mostrar por pantalla:



**Ejercicio básico inicial 2**:

Escribe un programa Java que realice lo siguiente: declarar dos variables X e Y de tipo int, dos variables N y M de tipo double y asigna a cada una un valor. A continuación muestra por pantalla:



Si por ejemplo le hemos dado a X el valor 1, a Y el valor 2, a M el valor 3.2 y a N el valor 4.7 se debe mostrar por pantalla:



public class Main {  
 public static void main(String[] args) {  
  
 int X = 6;  
 int Y = 5;  
 double M = 2.5;  
 double N = 3.5;  
  
 System.*out*.println(X + " + " + Y + " = " + (X+Y));  
 System.*out*.println(X + " - " + Y + " = " + (X-Y));  
 System.*out*.println(X + " x " + Y + " = " + (X\*Y));  
 System.*out*.println(X + " / " + Y + " = " + (X/Y));  
 System.*out*.println(X + " % " + Y + " = " + (X%Y));  
 System.*out*.println(N + " + " + M + " = " + (N+M));  
 System.*out*.println(N + " - " + M + " = " + (N-M));  
 System.*out*.println(N + " \* " + M + " = " + (N\*M));  
 System.*out*.println(N + " / " + M + " = " + (N/M));  
 System.*out*.println(N + " % " + M + " = " + (N%M));  
 System.*out*.println(X + " + " + N + " = " + (X+N));  
 System.*out*.println(Y + " / " + M + " = " + (Y/M));  
 System.*out*.println(Y + " % " + M + " = " + (Y%M));  
 System.*out*.println(X + " x 2 = " + (X\*2));  
 System.*out*.println(Y + " x 2 = " + (Y\*2));  
 System.*out*.println(N + " x 2 = " + (N\*2));  
 System.*out*.println(M + " x 2 = " + (M\*2));  
 System.*out*.println(X + " + " + Y + " + " + M + " + " + N + " = " + (X+Y+N+M));  
 System.*out*.println(X + " x " + Y + " x " + M + " x " + N + " = " + (X\*Y\*N\*M));  
  
 }  
 }

14. Ejercicios para practicar la salida de datos por pantalla: System.out.print y System.out.println – II

**Ejercicio básico inicial 3**

Escribe un programa Java que declare una variable entera N y asígnale un valor. A continuación escribe las instrucciones que realicen los siguientes:



Si por ejemplo N = 1 la salida del programa será:



public class Main {  
 public static void main(String[] args) {  
  
 int N = 10;  
  
 System.*out*.println(N + " + 77 = " + (N+=77));  
 System.*out*.println(N + " - 3 = " + (N-=3));  
 System.*out*.println(N + " x 2 = " + (N\*=2));  
  
 }  
 }

**Ejercicio básico inicial 4**

Programa java que declare cuatro variables enteras A, B, C y D y asígnale un valor a cada una. A continuación realiza las instrucciones necesarias para que:



Si por ejemplo A = 1, B = 2, C = 3 y D = 4 el programa debe mostrar:



public class Main {  
 public static void main(String[] args) {  
  
 int A = 2;  
 int B = 4;  
 int C = 6;  
 int D = 8;  
  
 System.*out*.println("Valores iniciales:");  
 System.*out*.println("A = " + A);  
 System.*out*.println("B = " + B);  
 System.*out*.println("C = " + C);  
 System.*out*.println("D = " + D);  
  
 int AUX = B;  
 B = C;  
 C = A;  
 A = D;  
 D = AUX;  
  
 System.*out*.println("Valores cambiados:");  
 System.*out*.println("A = " + A);  
 System.*out*.println("B = " + B);  
 System.*out*.println("C = " + C);  
 System.*out*.println("D = " + D);  
  
 }  
 }

15. Ejercicios con el operador condicional ( ? : ) - I

En esta entrada vamos a ver tres ejemplos de utilización del operador condicional ? :  
Se trata de usar el operador condicional en lugar de la instrucción condicional if para mostrar por pantalla un mensaje u otro dependiendo de una condición.  
  
**Ejercicio básico inicial 5**

Escribe un programa java que declare una variable A de tipo entero y asígnale un valor. A continuación muestra un mensaje indicando si A es par o impar. Utiliza el operador condicional ( ? : ) dentro del println para resolverlo.

Si por ejemplo A = 14 la salida será



Si fuese por ejemplo A = 15 la salida será:



public class Main {  
 public static void main(String[] args) {  
  
 int A = 5;  
  
 System.*out*.println(A % 2 == 0 ? A+" es par" : A+" es impar" );  
  
 }  
 }

**Ejercicio básico inicial 6**

Escribe un programa java que declare una variable B de tipo entero y asígnale un valor. A continuación muestra un mensaje indicando si el valor de B es positivo o negativo. Consideraremos el 0 como positivo. Utiliza el operador condicional ( ? : ) dentro del println para resolverlo.

Si por ejemplo B = 1 la salida será



Si fuese por ejemplo B = -1 la salida será:



public class Main {  
 public static void main(String[] args) {  
  
 int B = -10;  
  
 System.*out*.println(B >= 0 ? B+" es positivo" : B+" es negativo" );  
  
 }  
 }

**Ejercicio básico inicial 7**

Escribe un programa java que declare una variable C de tipo entero y asígnale un valor. A continuación muestra un mensaje indicando si el valor de C es positivo o negativo, si es par o impar, si es múltiplo de 5, si es múltiplo de 10 y si es mayor o menor que 100. Consideraremos el 0 como positivo. Utiliza el operador condicional ( ? : ) dentro del println para resolverlo.

Si por ejemplo C = 55 la salida será



public class Main {  
 public static void main(String[] args) {  
  
 int C = 100;  
  
 System.*out*.println(C >= 0 ? C+" es positivo" : C+" es negativo" );  
 System.*out*.println(C % 2 == 0 ? C+" es par" : C+" es impar" );  
 System.*out*.println(C % 5 == 0 ? C+" es multiplo de 5" : C+" no es multiplo de 5");  
 System.*out*.println(C % 10 == 0 ? C+" es multiplo de 10" : C+" no es multiplo de 10");  
 System.*out*.println(C >= 100 ? C+" es mayor o igual a 100" : C+" es menor a 100");  
  
 }  
 }

16. Ejercicios con el operador condicional ( ? : ) – II

**Ejercicio 1:**Un programa contiene las siguientes declaraciones y asignaciones iniciales de variables:



Determina el valor de cada una de las siguientes expresiones:



* k = 8
* k = 5
* z = 0.005
* z = 0
* a = ‘p’
* k = 5
* k = 0
* k = 9
* a = ‘q’
* a = ‘r’

**Ejercicio 2:**Un programa contiene las siguientes declaraciones y asignaciones iniciales de variables:



Determina el valor que toma la variable x al ejecutarse cada una de las siguientes expresiones. Las instrucciones son independientes unas de otras, es decir, el valor inicial para las variables i, j, k en cada instrucción es i = 1, j = 1, k = 1.



* x = 2
* x = 4
* x = 10
* x = 4
* x = 1
* x = 2
* x = -1
* x = 0
* x = 5
* x = 0

17. Ejercicios sobre expresiones algorítmicas





* 2 \* N
* 3 \* N
* 6 \* (A-B)
* (A \* B) – (C + D)
* A % 10 / 2
* N/100 + N%10 + N/10%10
* N % 10 + M % 10
* N % 2 == 0 && N % 3 == 0
* N % 10 % 2 == 0
* N/100 % 2 == 1
* N/1000 % 2 == 0
* A >= ‘A’ && A <= ‘Z’
* A >= ‘A’ && A <= ‘Z’ || A >= ‘a’ && A <= ‘z’
* A < ‘A’ || A > ‘Z’
* (A < ‘a’ || A > ‘z’) && (A < ‘A’ || A > ‘Z’)
* N % 10 == 0 || N % 10 == 7
* Precio >= 10 && precio < 50
* N += 77
* M -= 3
* X \*= 2
* N+=N%10
* N/1000 == N%10 && N/100%10 == N/10%10
* M == 4 || M == 6 || M == 9 || M == 11
* N = N/10
* N = N % 10000
* C >= ‘0’ && C <= ‘9’
* N = 1000

M = 2000

X = N/1000 + N/100%10 + N/10%10 + N%4

Y = M/1000 + M/100%10 + M/10%10 + M%4

Z = X + Y

* AA – AN >= 18 && MA > MN || AA – AN == 18 && MA == MN && DA >= DN

18. Ejercicios sobre expresiones algorítmicas – II

Escribe las expresiones algorítmicas equivalentes para cada uno de los siguientes enunciados:

* N / 1000 % 2 == 1
* N / 10000 == N/1000 % 10
* N /10000 == N % 10 && N/1000 % 10 == N/10%10
* C == ‘A’ || C == ‘E’ || C == ‘I’ || C == ‘O’ || C == ‘U’
* C != ‘A’ && C != ‘E’ && C != ‘I’ && C != ‘O’ && C != ‘U’
* A == ‘a’ || A == ‘e’ || A == ‘i’ || A == ‘o’ || A == ‘u’ &&

B == ‘a’ || B == ‘e’ || B == ‘i’ || B == ‘o’ || B == ‘u’)

* A % 4 == 0 && A % 100 != 0 || A % 400 == 0
* N/10 == N % 10
* (N/10 + N % 10) % 2 == 0
* N/10 % 10 > N/100 && N/10 % 10 > N % 10
* N/1000 == 4 || N/100 % 10 == 4 || N/10 % 10 == 4 || N % 10 == 4
* X = N/100
* N -=N%10
* M -=N%10
* N += N > 0 ? 5 : 10