

1. Tablas de multiplicar

Realice un programa que lea como dato un número entero "n", cuyo valor puede ser desde 1 al 9 y el programa realice lo siguiente:

- Validar el ingreso de n, solo debe aceptar valores desde 1 hasta 9.
- Si el número n es par, se imprimen todas las tablas de multiplicar desde el 2 hasta n, pero considerando solo los valores pares.
- Si el número n es impar, se imprimen todas las tablas de multiplicar desde 1 hasta n, pero considerando solo los valores impares.
- La tabla de multiplicar de cada número va desde 1 hasta 10.

Algunos ejemplos de diálogo de este programa serían:

Listing 1: Ejemplo 1

```
Numero desde 1 a 9 :-3
Numero desde 1 a 9 :14
Numero desde 1 a 9 :7

1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10

3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
```

```
3  x  7  =  21
3  x  8  =  24
3  x  9  =  27
3  x 10  =  30
```

```
5  x  1  =   5
5  x  2  =  10
5  x  3  =  15
5  x  4  =  20
5  x  5  =  25
5  x  6  =  30
5  x  7  =  35
5  x  8  =  40
5  x  9  =  45
5  x 10  =  50
```

```
7  x  1  =   7
7  x  2  =  14
7  x  3  =  21
7  x  4  =  28
7  x  5  =  35
7  x  6  =  42
7  x  7  =  49
7  x  8  =  56
7  x  9  =  63
7  x 10  =  70
```

Chao .

Listing 2: Ejemplo 2

```
Numero desde 1 a 9 :5
```

```
1  x  1  =   1
1  x  2  =   2
1  x  3  =   3
1  x  4  =   4
1  x  5  =   5
1  x  6  =   6
1  x  7  =   7
1  x  8  =   8
1  x  9  =   9
```

```
1  x  10  =  10

3  x  1   =  3
3  x  2   =  6
3  x  3   =  9
3  x  4   = 12
3  x  5   = 15
3  x  6   = 18
3  x  7   = 21
3  x  8   = 24
3  x  9   = 27
3  x 10   = 30
```

```
5  x  1   =  5
5  x  2   = 10
5  x  3   = 15
5  x  4   = 20
5  x  5   = 25
5  x  6   = 30
5  x  7   = 35
5  x  8   = 40
5  x  9   = 45
5  x 10   = 50
```

Chao .

Listing 3: Ejemplo 3

```
Numero desde 1 a 9 :6
```

```
2  x  1   =  2
2  x  2   =  4
2  x  3   =  6
2  x  4   =  8
2  x  5   = 10
2  x  6   = 12
2  x  7   = 14
2  x  8   = 16
2  x  9   = 18
2  x 10   = 20
```

```
4  x  1  =  4
4  x  2  =  8
4  x  3  = 12
4  x  4  = 16
4  x  5  = 20
4  x  6  = 24
4  x  7  = 28
4  x  8  = 32
4  x  9  = 36
4  x 10  = 40
```

```
6  x  1  =  6
6  x  2  = 12
6  x  3  = 18
6  x  4  = 24
6  x  5  = 30
6  x  6  = 36
6  x  7  = 42
6  x  8  = 48
6  x  9  = 54
6  x 10  = 60
```

Chao .

Listing 4: Ejemplo 4

```
Numero desde 1 a 9 :4
```

```
2  x  1  =  2
2  x  2  =  4
2  x  3  =  6
2  x  4  =  8
2  x  5  = 10
2  x  6  = 12
2  x  7  = 14
2  x  8  = 16
2  x  9  = 18
2  x 10  = 20
```

```
4  x  1  =  4
4  x  2  =  8
4  x  3  = 12
```

```
4   x   4   =   16
4   x   5   =   20
4   x   6   =   24
4   x   7   =   28
4   x   8   =   32
4   x   9   =   36
4   x  10   =   40
```

Chao .

2. Numeritos

Desarrolle un programa que permita leer como dato un número cuyo rango puede estar desde 1 a 9. Este dato representa la cantidad de filas que tendrá un rectángulo formado por números.

El programa:

- Debe verificar el ingreso del número de filas. Este dato deberá ser desde 1 a 9.
- Imprime un rectángulo formado por números tal y como se muestra en los ejemplos.

Algunos ejemplos de diálogo de este programa serían:

Listing 5: Ejemplo 1

```
Filas: 12
Filas: -3
Filas: 9

000000009
000000098
000000987
000009876
000098765
000987654
009876543
098765432
987654321

Chao .
```

Listing 6: Ejemplo 2

```
Filas: 6

000006
```

```
000065
000654
006543
065432
654321

Chao .
```

Listing 7: Ejemplo 3

```
Filas : 7

0000007
0000076
0000765
0007654
0076543
0765432
7654321

Chao .
```

Listing 8: Ejemplo 4

```
Filas : 3

003
032
321

Chao .
```

3. Arbolitos - bucles

Desarrolle un programa que permita leer como datos: un caracter, el **número de figuras**, y el número de filas que tendrá cada figura. Cada figura será un triángulo invertido según el número de filas.

El programa debe validar:

- Que los valores del número de figuras y el número de filas sea mayor a cero.

Algunos ejemplos de diálogo de este programa serían:

Listing 9: Ejemplo 1

```
Character : #
```

```
Numero de figuras: 5
Numero de filas de la figura: 3

###
##
#

###
##
#

###
##
#

###
##
#

###
##
#
```

Listing 10: Ejemplo 2

```
Character: &
Numero de figuras: 7
Numero de filas de la figura: 4

&&&&
&&&
&&
&

&&&&
&&&
&&
&

&&&&
&&&
&&
&

&&&&
```

```
&&&
&&
&

&&&&
&&&
&&
&

&&&&
&&&
&&
&

&&&&
&&&
&&
&
```

Listing 11: Ejemplo 2

```
Character: @
Numero de figuras: 2
Numero de filas de la figura: 9

@@@@@@@@@@
@@@@@@@@@
@@@@@@@@
@@@@@@@
@@@@@@
@@@@@
@@@@
@@@
@@
@

@@@@@@@@@@
@@@@@@@@@
@@@@@@@@
@@@@@@@
@@@@@@
@@@@@
@@@@
@@
@
```


Listing 12: Ejemplo 2

```
Caracter: *
Numero de figuras: 2
Numero de filas de la figura: 1

*

*
```

4. Bucles en el planeta Gorinki

En el planeta Gorinki, la presión sistólica de los gorinkianos oscila entre 90 y 130 como valores normales. Debido a la pandemia los gorinkianos han sufrido mucho y se ha reportado una subida en el caso de hipertensos. Para determinar si un gorinkiano es hipertenso, se le toma la presión sistólica 5 veces al día y si el promedio esta por encima de 130 se considera hipertenso.

Realice un programa que permita leer el número de gorinkiano a los cuales se hará el test para determinar si son o no hipertensos.

El programa:

- Debe verificar el ingreso del número de gorinkianos. Este dato deberá ser mayor a 3
- Luego leer para cada gorkiniano, los valores de las 5 pruebas de presión sistólica, hallar el promedio y determinar finalmente si es hipertenso o no.

Algunos ejemplos de diálogo de este programa serían:

Listing 13: Ejemplo 1

```
Numero de gorinkis : 2
Numero de gorinkis : -1
Numero de gorinkis : 4

Presion sist lica 1: 123
Presion sist lica 2: 130
Presion sist lica 3: 160
Presion sist lica 4: 155
Presion sist lica 5: 134
Ud. es hipertenso, el promedio de su presi n sist lica es
: 140.4
Presion sist lica 1: 120
Presion sist lica 2: 110
Presion sist lica 3: 90
Presion sist lica 4: 80
```

```
Presion sist lica 5: 111
Ud.No es hipertenso, el promedio de su presi n sist lica
  es : 102.2
Presion sist lica 1: 90
Presion sist lica 2: 95
Presion sist lica 3: 110
Presion sist lica 4: 120
Presion sist lica 5: 130
Ud.No es hipertenso, el promedio de su presi n sist lica
  es : 109.0
Presion sist lica 1: 70
Presion sist lica 2: 80
Presion sist lica 3: 81
Presion sist lica 4: 85
Presion sist lica 5: 99
Ud.No es hipertenso, el promedio de su presi n sist lica
  es : 83.0
Gracias por usar el programa!
```

Listing 14: Ejemplo 2

```
Numero de gorinkis : 3
Numero de gorinkis : 5

Presion sist lica 1: 120
Presion sist lica 2: 135
Presion sist lica 3: 125
Presion sist lica 4: 130
Presion sist lica 5: 130
Ud.No es hipertenso, el promedio de su presi n sist lica
  es : 128.0
Presion sist lica 1: 90
Presion sist lica 2: 95
Presion sist lica 3: 90
Presion sist lica 4: 123
Presion sist lica 5: 145
Ud.No es hipertenso, el promedio de su presi n sist lica
  es : 108.6
Presion sist lica 1: 145
Presion sist lica 2: 150
Presion sist lica 3: 160
Presion sist lica 4: 170
Presion sist lica 5: 145
Ud. es hipertenso, el promedio de su presi n sist lica es
  : 154.0
```

```
Presion sist lica 1: 70
Presion sist lica 2: 60
Presion sist lica 3: 70
Presion sist lica 4: 90
Presion sist lica 5: 80
Ud.No es hipertenso, el promedio de su presi n sist lica
  es : 74.0
Presion sist lica 1: 65
Presion sist lica 2: 70
Presion sist lica 3: 80
Presion sist lica 4: 90
Presion sist lica 5: 100
Ud.No es hipertenso, el promedio de su presi n sist lica
  es : 81.0
Gracias por usar el programa!
```

5. Números amigos - funciones

Desarrolle un programa, que permita leer dos números enteros positivos diferentes y el programa indique si los números son **números amigos**.

Dos números enteros positivos **son amigos**: si la **suma de los divisores propios de uno es igual al otro número y viceversa**. El 1 es considerado divisor propio, pero el número mismo no lo es.

Por ejemplo:

Los divisores propios de 220 son 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110, **que suman 284**

Los divisores propios de 284 son 1, 2, 4, 71 y 142, **que suman 220**.

Por tanto el 284 y el 220 son números amigos.

El programa:

- Debe verificar el ingreso de dos números enteros positivos diferentes.
- Codifique la **función sonAmigos** que reciba como datos los dos números y devuelva una valor bool que indique True si los números son amigos o False si no lo son.
- **Para que se pueda asignar el puntaje a esta pregunta es absolutamente necesario que se use la función sonAmigos. De lo contrario no se podrá asignar el puntaje.**

Algunos ejemplos de diálogo de este programa serían:

Listing 15: Ejemplo 1

```
Numero 1 : 220
```

```
Numero 2 : 284
220 y 284 son numeros amigos
```

Listing 16: Ejemplo 2

```
Numero 1 : 2324
Numero 2 : 4567
2324 y 4567 No, son numeros amigos
```

Listing 17: Ejemplo 3

```
Numero 1: 1184
Numero 2: 1184

Numero 1: 156
Numero 2: 156

Numero 1: 17296
Numero 2: 18416
17296 y 18416 son numeros amigos
```

Algunos ejemplos de diálogo de este programa serían:

Listing 18: Ejemplo 4

```
Numero 1 : 1184
Numero 2 : 1210
1184 y 1210 son numeros amigos
```

6. Día que faltan para año nuevo - Evalúa funciones.

Desarrolle un programa que permita leer como dato una fecha: día, mes y año, para que luego el programa indique:

- **La cantidad de días que faltan para que llegue año nuevo.**

Recuerde que la cantidad de días que tiene cada mes es como se indica en la tabla:

Número de mes	Número de días
1,3,5,7,8,10,12	31
4,6,9,11	30
2	28 si no es año bisiesto y 29 si es bisiesto

- Para efectos de este programa, considere que un año es bisiesto si es múltiplo de 4.

- No es necesario que valide la fecha.

Resuelva el programa utilizando al menos:

- **La función `hallaDias`**, que reciba como parámetro el día, mes y año y devuelva la cantidad de días que faltan para el año nuevo.

De no escribir esta función, no se asignará el puntaje a esta pregunta.

Algunos ejemplos de diálogo de este programa serían:

Listing 19: Ejemplo 1

```
Dia: 14
Mes: 10
Anio: 2020
Los dias que falta para anio nuevo : 78
```

Listing 20: Ejemplo 2

```
Dia: 27
Mes: 7
Anio: 2020
Los dias que falta para anio nuevo : 157
```

Listing 21: Ejemplo 3

```
Dia: 1
Mes: 1
Anio: 2020
Los dias que falta para anio nuevo : 365
```

Listing 22: Ejemplo 4

```
Dia: 31
Mes: 12
Anio: 2019
Los dias que falta para anio nuevo : 0
```

7. Posiciones - Evalúa str y funciones

Desarrolle un programa que solicite una frase y un caracter, el programa por medio de una función halle:

- La posición de la primera ocurrencia del caracter en la frase, si no hubiera ningun valor se devolverá -1
- La posición de la última ocurrencia del caracter en la frase, si no hubiera ningun valor se devolverá -1
- La cantidad de veces que el caracter aparece en la frase

- La coincidencia es exacta, es decir hay diferencia entre letras mayúsculas y letras minúsculas.
- Note que la función deberá devolver los 3 valores. Por lo que tenga cuidado con el orden en que se devuelven los parámetros.

```
Frase: anita lava la tina
Caracter: a
Posicion de la primera ocurrencia: 0
Posicion de la ultima ocurrencia : 17
Veces que aparece el caracter    : 6
```

```
Frase: anita lava la TINA
Caracter: a
Posicion de la primera ocurrencia: 0
Posicion de la ultima ocurrencia : 12
Veces que aparece el caracter    : 5
```

```
Frase: anita lava la tina
Caracter: n
Posicion de la primera ocurrencia: 1
Posicion de la ultima ocurrencia : 16
Veces que aparece el caracter    : 2
```

```
Frase: anita lava la tina
Caracter: p
Posicion de la primera ocurrencia: -1
Posicion de la ultima ocurrencia : -1
Veces que aparece el caracter    : 0
```

8. Siguiente Primo

Desarrolle un programa que permita leer un número mayor que 1 y el programa imprima el siguiente número primo. El programa debe:

- Validar el ingreso del numero mayor a uno
- Al realizar el algoritmo utilizar por lo menos una función propia. Para poder asignar el puntaje a esta pregunta **es necesario que por no lo menos de diseñe una función.**

Algunos ejemplos de diálogo de este programa serían:

Listing 23: Ejemplo 1

```
Numero : 5
Siguiente numero primo: 7
```

Listing 24: Ejemplo 2

```
Numero : 13
Siguiete numero primo: 17
```

Listing 25: Ejemplo 3

```
Numero : 45
Siguiete numero primo: 47
```

Listing 26: Ejemplo 4

```
Numero : 123
Siguiete numero primo: 127
```

9. Deletrea el numero - Str.

Diseñe e implemente un programa que permita leer un número entero mayor a cero y el programa permita **deletrear el número**, es decir indicar en palabras cada dígito que forma parte del número.

El programa deberá:

- Validar el ingreso del número.
- Para asignar el puntaje total a esta pregunta es necesario que forme **una cadena** en la que figuren los nombres de cada dígito e imprimir dicha cadena.

Algunos ejemplos de diálogo de este programa serían:

Listing 27: Ejemplo 1

```
Numero: 0
Numero: -4
Numero: -8
Numero: 67854
seis siete ocho cinco cuatro
```

Listing 28: Ejemplo 2

```
Numero: 670
seis siete cero
```

Listing 29: Ejemplo 2

```
Numero: 2346557802
dos tres cuatro seis cinco cinco siete ocho cero dos
```

10. Las cinco vocales - Evalúa str

Desarrolle un programa que permita leer una palabra y el programa indique si la palabra tiene **todas las 5 letras vocales o no**.

Al realizar el programa considere:

- Que la palabra podría tener letras en minúsculas o mayúsculas
- La palabra solo podrá contener letras del alfabeto inglés, esto significa que no se consideran letras con tilde.

Algunos ejemplos de diálogo de este programa serían:

Listing 30: Ejemplo 1

```
Input:
Palabra: Aurelio

Output:
La palabra tiene las 5 vocales!
```

Listing 31: Ejemplo 2

```
Input:
Palabra: archiTECTO

Output:
La palabra tiene las 5 vocales!
```

Listing 32: Ejemplo 3

```
Input:
Palabra: abuelito

Output:
La palabra tiene las 5 vocales!
```

Listing 33: Ejemplo 4

```
Input:
Palabra: Julio

Output:
La palabra no tiene las 5 vocales!
```