

Universidad Mayor de San Andrés
Facultad de Ciencias Puras y Naturales
Carrera de Informática



INTRODUCCIÓN A LA INFORMÁTICA INF - 111
PROGRAMACIÓN I INF-111
LABORATORIO DE INF - 111

PRÁCTICA GENERAL

Auxiliares:

Sarai Blanco Salgado
Sergio Daniel Ramirez Castro
Kelvin Bohn Davisson Vargas
Garcia
Alex Ramos Jimenez
Marcelo Nelson De La Quintanilla Illanes

Marcelo Luis Cusi Fernandez
Yessid Gaston Miranda Villca
Javier Menacho Paca
Carlos Manuel Jahuira Sullca

Gestión I/2023

Secuenciales

1. Conversión de moneda

Escribe un programa que convierta una cantidad de dólares a euros. El programa debe pedir al usuario que ingrese la cantidad en dólares y mostrar la cantidad equivalente en euros. Se considera que el cambio de divisa es de 1 dólar = 0,83 euros.

2. Tiempo de viaje

Escribe un programa que calcule el tiempo que tardará un vehículo en recorrer una distancia a una velocidad constante. El programa debe pedir al usuario que ingrese la distancia en kilómetros y la velocidad en kilómetros por hora, y mostrar el tiempo que tardará en recorrer la distancia en horas.

3. Área y circunferencia de un círculo

Escribe un programa que lea el radio de un círculo y muestre su área y circunferencia en la pantalla. Utiliza las fórmulas siguientes:

$$\text{Área} = \pi * \text{radio}^2$$

$$\text{Circunferencia} = 2 * \pi * \text{radio}$$

4. Conversión de unidades

Escribe un programa que convierta una longitud en pies y pulgadas a centímetros. El programa debe pedir al usuario que ingrese la cantidad de pies y pulgadas y mostrar la longitud equivalente en centímetros. Se considera que 1 pie = 12 pulgadas y 1 pulgada = 2.54 centímetros.

5. Tiempo de descarga de un archivo

Escribe un programa que calcule el tiempo de descarga de un archivo en función del tamaño del archivo y la velocidad de descarga en megabits por segundo. El programa debe pedir al usuario que ingrese el tamaño del archivo en megabytes y la velocidad de descarga en megabits por segundo, y mostrar el tiempo de descarga en minutos y segundos.

6. Cálculo de la hipotenusa de un triángulo

Escribe un programa que calcule la longitud de la hipotenusa de un triángulo rectángulo en función de las longitudes de los catetos. El programa debe pedir al usuario que ingrese la longitud de los catetos y mostrar la longitud de la hipotenusa.

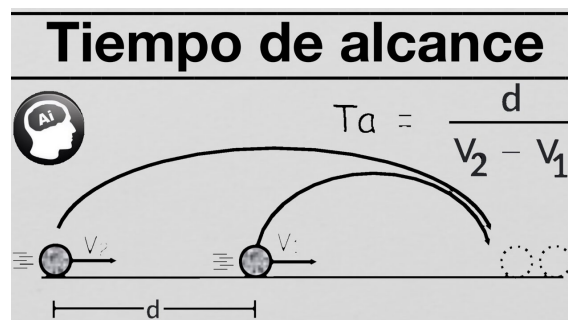
7. Conversión.

Escribir un programa que convierta un valor dado en grados Fahrenheit a grados Celsius. Recordar que la fórmula para la conversión es:

$$C = (F-32)*5/9$$

8. Tiempo de alcance.

Dos vehículos viajan a diferentes velocidades (v_1 y v_2) y están distanciados por una distancia d . El que está detrás (v_1) viaja a una velocidad mayor. Se pide hacer un algoritmo para ingresar la distancia entre los dos vehículos (km) y sus respectivas velocidades (km/h) y con esto determinar y mostrar en qué tiempo (minutos) alcanzará el vehículo más rápido al otro.



Condicionales

1. Escribe un programa que pida al usuario una letra y determine si es una vocal o una consonante. Imprime "Es una vocal" o "Es una consonante" según corresponda.
2. Escribe un programa que solicite al usuario ingresar tres números enteros. El programa deberá imprimir en pantalla el mensaje "Hay un número par" si al menos uno de los números ingresados es par. De lo contrario, deberá imprimir en pantalla el mensaje "No hay números pares".

3. Calculadora de Índice de Masa Corporal (IMC)

Escribe un programa que pida al usuario su peso en kilogramos y su altura en metros, y calcule su Índice de Masa Corporal (IMC) utilizando la siguiente fórmula:

$$\text{IMC} = \text{peso} / \text{altura}^2$$

Luego, el programa debe imprimir el resultado del IMC y una categoría que indique si el usuario está en su peso ideal, por debajo o por encima de él. Las categorías son las siguientes:

- Bajo peso: IMC menor a 18.5
- Peso normal: IMC entre 18.5 y 24.9
- Sobrepeso: IMC entre 25 y 29.9
- Obesidad: IMC mayor a 30

4. Escribe un programa que solicite al usuario ingresar una letra. Si la letra ingresada es una vocal (a, e, i, o, u), el programa deberá imprimir en pantalla el mensaje "La letra ingresada es una vocal". De lo contrario, el programa deberá imprimir en pantalla el mensaje "La letra ingresada es una consonante".
5. Escribe un programa que pida al usuario un número entero y determine si es múltiplo de 3 y/o de 5. Imprime "Es múltiplo de 3", "Es múltiplo de 5" o "No es múltiplo de 3 ni de 5" según corresponda.
6. Escribe un programa que solicite al usuario ingresar dos números enteros. El programa deberá imprimir en pantalla el resultado de la suma de los dos números si ambos números son mayores que 10. Si alguno de los números es menor o igual que 10, el programa deberá imprimir en pantalla el mensaje "Los números ingresados deben ser mayores que 10".
7. Escribe un programa que solicite al usuario ingresar su edad. Si el usuario tiene entre 18 y 25 años (ambos inclusive), el programa deberá imprimir en pantalla el mensaje "¡Felicidades! Eres elegible para nuestra oferta especial para jóvenes". Si el usuario tiene más de 25 años, el programa deberá imprimir en

pantalla el mensaje "¡Gracias por tu interés en nuestra oferta especial para jóvenes, pero esta oferta es solo para personas entre 18 y 25 años!".

8. Escribe un programa que solicite al usuario ingresar el lenguaje de programación que está aprendiendo. Si el usuario ingresa "Python", el programa deberá imprimir en pantalla el mensaje "¡Python es un excelente lenguaje de programación! Sigue adelante". Si el usuario ingresa "JavaScript", el programa deberá imprimir en pantalla el mensaje "¡JavaScript es un lenguaje de programación popular y útil! Sigue adelante". Si el usuario ingresa cualquier otro lenguaje de programación, el programa deberá imprimir en pantalla el mensaje "¡Ese es un buen lenguaje de programación! Sigue adelante".

Repetitivos

1. Con el uso de fors, resolver:

Entrada:

6

Salida:

123456

23456

3456

456

56

6

65

654

6543

65432

654321

2. Con el uso de while, resolver:

Entrada:

5

Salida:

12345
23454
34543
45432
54321

3. Con el uso de do-while verifique que el ingreso sea un número n mayor a 0 y muestre el factorial de n :

Entrada:

0
-2
5

Salida:

120

Explicación:

El último número ingresado es un número mayor a 0, entonces se ejecuta el algoritmo del factorial.

4. Con el uso de for y while resolver:

Entrada:

7

Salida:

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1

5. Resuelva el siguiente problema:

Entrada:

5

Salida:

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

6. Resuelva el siguiente problema

Entrada:

5

Salida:

1 2 3 4 5

6 7 8 9

10 11 12

13 14

15

7. Resuelva el siguiente problema

Entrada:

4

Salida:

0 0 0 0

0 0 0 1

0 0 0 0

0 0 2 1

0 0 0 0

0 3 2 1

0 0 0 0

4 3 2 1

8. Resuelva el siguiente problema

Caso 1:

Entrada:

5

Salida:

1 0 0 0 2

1 1 0 2 2

1 1 1 2 2

1 1 1 1 2

1 1 1 1 1

Caso 2:

Entrada:

6

Salida:

1 0 0 0 0 2

1 1 0 0 2 2

1 1 1 2 2 2

1 1 1 1 2 2

1 1 1 1 1 2

1 1 1 1 1 1

9. Resuelva el siguiente problema:

Entrada:

5

Salida:

123454321

1234 4321

123 321

12 21

1 1

10.Diseñe un menú que pueda resolver:

MENU

Ingrese una opción entre 1 y 4:

Mostrar el factorial del número n.

Mostrar si el número n es un número primo

Mostrar todos los números pares desde 0 a n inclusive

Mostrar todos los números impares desde 1 a n inclusive

Observaciones. Si la opción elegida es diferente de 1 a 4 mostrar un mensaje de error.

Caso 1

Entrada:

2

7

Salida:

“El número 7 es primo”

Caso 2

Entrada:

1

3

Salida:

“El factorial de 3 es: 6”

Caso 3

Entrada:

3

6

Salida:

0 2 4 6

Caso 4

Entrada:

4

3

Salida:

1 3

Caso 5

Entrada:

8

Salida:

“La opción 8 no existe”

Series

1. 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...
2. 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, ...
3. 2, 0, 3, 1, 5, 1, 7, 2, 11, 3, 13, 5, 17, 8, 19, 13, ...
4. 1, 2, 1, 2, 3, 2, 1, 2, 3, 4, 5, 4, 3, 2, 1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 10, 9...
5. 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6, ...
6. 1, -1, -1, 0, -1, 1, -1, 0, 0, 1, -1, 0, -1, 1, 1, 0, -1, 0, ...
7. 2, 1, 2, 4, 5, 4, 6, 5, 6, 8, 7, 8, 10, 9, 10, ...
8. 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0 ...
9. 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, ...
10. 1, 2, 3, 4, 5, 4, 3, 2, 1, 2, 3, 4, 5, 4, 3, 2, 1, 2, 3, ...

Sumatorias

Resolver las siguientes sumatorias para **n** términos, en caso de tener operaciones con fracciones, debe mostrar la salida con formato de 2 decimales.

1.

$$\frac{1}{3} + \frac{2}{2} + \frac{1}{5} + \frac{4}{4} + \frac{3}{7} + \frac{6}{6} + \frac{5}{9} + \frac{8}{8} + \dots$$

Entrada:

5

Salida:

2.96

Entrada:

8

Salida:

5.52

2.

$$0 + 1 - 1 + 2 - 3 + 5 - 8 + 13 - \dots$$

Entrada:

5

Salida:

-1

Entrada:

10

Salida:

22

3.

$$0 + 1 + 1 + 2 + 4 + 8 + 16 + 31 + 61 + 120 + 236 + \dots$$

Entrada:

10

Salida:

244

Entrada:

17

Salida:

27728

4.

$$\frac{1+2}{3} + \frac{3+4}{5} + \frac{5+6}{7} + \frac{7+8}{9} + \frac{9+10}{11} + \dots$$

Entrada:

1

Salida:

1.00

Entrada:

5

Salida:

7.37

5.

$$0 + 0 + \frac{4}{5} + \frac{6}{7} + \frac{8 + 9 + 10}{11} + \frac{12}{13} + \frac{14 + 15 + 16}{17} + \frac{18}{\dots} + \dots$$

Entrada:

3

Salida:

0.80

Entrada:

8

Salida:

8.63

Lotes

1. Dado un lote de n números, encontrar el menor y mayor, encontrar el MCD de ambos valores.
2. Dado n números enteros A_i , ($1 \leq i \leq n \leq 10000000$) donde $0 \leq A_i \leq 100$ imprimirlos de manera ordenada ascendente.
Si $N=6$ y $A=\{3,4,2,5,1,6\}$
La secuencia sería 1,2,3,4,5,6
3. Dado n números enteros A_i , ($1 \leq i \leq n \leq 1000$) donde $0 \leq A_i \leq 100$ encontrar el número que se repite más veces si existen varios imprimir el menor.
Si $N=6$ y $A=\{1,4,2,2,1,6\}$ se imprime el 1 ya que el 1 2 veces y 1 es menor que 2
4. Sumar los números impares de un lote hasta que se ingrese un número negativo
Si el lote es: 1 4 8 6 5 3 -1
R. 9
5. Sumar los números primos de un lote hasta que se ingrese un número negativo
Si el lote es: 1 13 6 5 3 12 -7
R. 21

6. Dado n números enteros A_i , ($1 \leq i \leq n \leq 1000$) donde $0 \leq A_i \leq 100000$ imprimir la cantidad de números primos sin repeticiones
Si $N=6$ y $A=\{1,4,2,2,1,6\}$ se imprime 1 ya que el único número primo presente es el 2
7. Sumar todos los dígitos de los números ingresados hasta que se ingrese -1 sin adicionar este mismo
Para la entrada: 1,2,3,4,11,12,-1 Calculamos $1+2+3+4+1+1+1+2=15$
8. Dado un número ($1 \leq i \leq n \leq 1000000$) imprime si hay alguna forma de mover la posición de sus dígitos de tal manera que este número sea capicúa
Por ejemplo 1122334 se puede mover de las siguientes formas 1234321,21412,3124213,1324231 imprimir “si se puede”, caso contrario, ”No se puede”
9. Leer A_i números $0 \leq A_i \leq 9$ hasta que la composición de esos dígitos sea múltiplo de 3.
Ejemplo: 3, 2, 1, 5 parar, porque 3215 es múltiplo de 5
10. Dado n números enteros A_i , ($1 \leq i \leq n \leq 100000$) donde $0 \leq A_i \leq 1$, Imprimir el tamaño de la secuencia más grandes de unos y ceros.
Si $N = 9$ y los $A_i = \{1,0,1,1,1,1,0,0,1\}$
Tamaño de la secuencia más grande de 0 es 2.
Tamaño de la secuencia más grande de 1 es 4.

Descomposición

1. Escribe un programa que le pida al usuario un número entero y luego descomponga el número en sus dígitos y los muestre en orden inverso.
Ejemplo:

Entrada:

1234

Salida:

4 3 2 1

2. Escribe un programa que le pida al usuario un número entero y luego descomponga el número en sus dígitos y los muestre en orden inverso. Además, el programa debe mostrar cada dígito junto con su posición en el número original.

Ejemplo:

Entrada:

1234

Salida:

4 está en la posición 1

3 está en la posición 2

2 está en la posición 3

1 está en la posición 4

3. Escribe un programa que le pida al usuario un número entero y luego descomponga el número en sus dígitos y los muestre en orden ascendente. Además, el programa debe mostrar la suma de los dígitos.

Ejemplo:

Entrada:

547

Salida:

4 5 7

La suma de los dígitos es 16

4. Escribe un programa que le pida al usuario un número entero y luego descomponga el número en sus dígitos y los muestre en orden inverso, pero solo los dígitos impares.

Ejemplo:

Entrada:

123456789

Salida:

9 7 5 3 1

5. Escribe un programa que le pida al usuario un número entero y luego descomponga el número en sus dígitos y los muestre en orden ascendente, pero solo los dígitos pares.

Ejemplo:

Entrada:

987654321

Salida:

2 4 6 8

6. Escribe un programa que le pida al usuario un número entero y luego descomponga el número en sus dígitos y muestre los dígitos separados por un espacio, pero solo si el dígito es divisible por 3.

Ejemplo:

Entrada:

135798642

Salida:

1 3 6 9

7. Escribe un programa que le pida al usuario un número entero y luego descomponga el número en sus dígitos y muestre los dígitos separados por un espacio, pero solo si el dígito es un número primo.

Ejemplo:

Entrada:

123456789

Salida:

2 3 5 7

Modularidad

1. Verificar si un número dado es número primo, si este es primo imprimir un mensaje que diga “es primo” si no lo es, imprimir un mensaje que diga “no es primo”

Ejemplo Entrada

3

8

Ejemplo Salida

es primo

no es primo

2. Obtener el mayor entre dos números mayores que cero en caso de que alguno sea negativo volverlo positivo y encontrar el mayor.

Ejemplo Entrada	Ejemplo Salida
1 2	1
20 19	20
1234 5000	5000
-5 1	5

3. Dado un número mayor que cero, verificar si es capicua o no, imprimir "SI" si es así, caso contrario imprima "NO".

Ejemplo Entrada	Ejemplo Salida
121	SI
2019	NO
12321	SI

4. Realizar una función que realice el cálculo de a^x . Para a y x números enteros no debe usar ** o pow o alguna función ya creada.

Ejemplo Entrada	Ejemplo Salida
2 3	8
3 2	6

5. Realizar un algoritmo que muestre el n-ésimo número Fibonacci.

Ejemplo Entrada	Ejemplo Salida
1	0
3	1
5	3

6. Determinar la cantidad de dígitos pares e impares que tiene un número x.

Ejemplo de entrada	Ejemplo de Salida
123	1 2
1234	2 2

7. Realizar una función que convierta un número decimal en un número binario.

Ejemplo de entrada	Ejemplo de Salida
7	111
10	1010

8. Crear un función que reciba un número y devuelva el menor y el mayor dígito.

Ejemplo de entrada	Ejemplo de Salida
123	1 3
5671	1 7

9. Crear un procedimiento o función que calcule la ocurrencia de un número y determine el número de apariciones de un dígito x.

Ejemplo de entrada	Ejemplo de Salida
123 1	1
23342 2	2

10. Crear una función que dado un número n devuelve un número x este número será igual a la suma de todos n primeros números fibonacci y con otra función verifique si este es un número para o impar.

Ejemplo de entrada	Ejemplo de Salida
1	0 es par
2	1 es impar
3	2 es par
4	4 es par
5	7 es impar

Cadenas

1. Dada una cadena (sin contar números, ni caracteres especiales), ordenarlo lexicográficamente, nótese que los caracteres serán convertidos a minúscula.

Entrada:

Tu puedes con este ejercicio

Salida:

cccdeeeeeiijnooprsttuu

2. CONTANDO CARACTERES: Leer una cadena, posteriormente leer un número, que indicará las veces que se leerá por teclado un carácter (este carácter debe ser leído en minúsculas) y contar cuantas veces se repite ese carácter.

Entrada:

Este ejercicio esta regalado

2

Ingrese caracter: e

Ingrese caracter: t

Salida:

6

2

3. Dada una frase mostrar **las palabras** que tengan más consonantes.

Entrada:

Si Puedes Imaginarlo Puedes Programarlo

Salida:

Programarlo

4. VERIFICANDO LA CONTRASEÑA: Se dice que una contraseña es segura, si es que es mínimamente tiene un tamaño mayor o igual a 8, tambien si es que por lo menos tiene 1 número, 1 letra mayúscula y 1 carácter especial (@,#,\$,_etc...), realizar un programa que dado una verifique si una contraseña es segura, en el caso de que sea segura imprimir “es segura”, en el caso de que no lo sea imprimir “te van a hackear”.

Entrada:

contraseña

Salida:

te van a hackear

5. RUNG-LENGTH: Dada una cadena, imprimir la cantidad de veces que se repite una letra, al lado de la letra correspondiente.

Entrada:

AAARRAAATTBBPPOrreeewe

Salida:

3A2R3A2T2B2P1O3r3e1w1e

6. SEGURIDAD OBSESIVA: Dada una cadena, encriptarlo de la siguiente manera.

Entrada:

hello

Salida:

abccd

Entrada:

i like it

Salida:

abcadebaf

7. REDUCIENDO LA CADENA: Se tiene una cadena de tamaño compuesta por ceros y unos, la cadena se reduce siempre y cuando exista un par de 1 cero y 1 uno, si es que existe podemos remover esos dos dígitos de la cadena, imprima el tamaño de la nueva cadena.

Entrada:

11011101011

Salida:

5

Entrada:

0

Salida:

1

8. GENERAR CÓDIGO: Dada una cadena, que represente el nombre de completo de una persona, generar el código de esa persona, el código se genera tomando los primeros caracteres del nombre y uniendolos alfabéticamente, concatenando con un guión (-) seguido del código ascii de cada carácter.

Entrada:

Carlos Antonio Limachi Nina

Salida:

ACLN-65677678

Entrada:

Juan Jose Soliz Juares

Salida:

JJJS-74747483

Vectores

1. Primero se ingresará un número n que es el tamaño del vector, posteriormente los datos que posee el vector. Del vector dado mostrar el número mayor, el número menor y el promedio del vector. (OJO, es posible que ingresen números negativos)

Entrada:

5

0 -5 8 9 13

Salida:

Mayor: 10

Menor: -5

Promedio: 5

2. Leer por teclado un número n , que indica el tamaño de un vector, luego los datos del vector. Llevar los números pares al principio y los números impares al final.

Entrada:

8

3 6 12 45 1 0 8 21

Salida:

6 12 0 8 3 45 1 21

3. Dado un número n , que indica el tamaño de un vector, luego los datos del vector introducidos por teclado (números enteros), calcular el promedio de sus datos, luego contar y mostrar cuántos elementos son mayores y cuántos son menores o iguales.

Entrada:

6
3 6 13 28 18 4

Salida:

mayores: 3
menores: 3

4. Dado un vector de números naturales, de tamaño n , y luego un número x , determinar cuántos divisores y cuántos múltiplos de x hay en el vector. (si hay un número igual a x dentro del vector cuenta tanto como divisor y como múltiplo).

Entrada:

8
32 6 84 23 36 12 3 1
12

Salida:

divisores: 4
múltiplos: 3

5. Dado dos vectores de enteros positivos (u y v), ambos de tamaño n e introducidos por teclado, generar un nuevo vector donde se encuentre la diferencia entre cada par de elementos u_i y v_i .

Entrada:

5
8 1 29 17 4
11 46 15 5 4

Salida:

3 45 14 12 0

6. Dado un número $n > 0$, generar el siguiente vector:

1	1	2	1	2	3	1	2	3	4	1	2	...
---	---	---	---	---	---	---	---	---	---	---	---	-----

Entrada:

15

Salida:

1 1 2 1 2 3 1 2 3 4 1 2 3 4 5

7. Dado un número $n > 0$, generar el siguiente vector:

si $n = 4$:

4	1	3	2	2	3	1	4
---	---	---	---	---	---	---	---

Entrada:

5

Salida:

5 1 4 2 3 3 2 4 1 5

8. Dado un arreglo A de tamaño N imprimir su arreglo adjunto acumulado C. Se define como vector acumulado de A a un vector C del mismo tamaño tal que todo elemento i-esimo en C es igual a la suma de los i elementos de A.

Entrada:

$N = 5$

$A = \{ 7, 5, 0, 1, 5 \}$

Salida:

$C = \{ 7, 12, 12, 13, 18 \}$

9. Llene un vector A de N (será ingresado por teclado) elementos **aleatorios** en el rango de 1 al 100. Imprimir el vector y luego el promedio de todos los primos y verificar si el **valor entero (sin redondear)** de ese número promedio existe en el vector A, en caso de existir se debe imprimir “si existe” y en caso de no existir se debe imprimir “no existe”.

Entrada:

10

Salida:

Vector A:

1 23 42 2 3 10 9 7 30 8

8 -> si existe

10. Del ejercicio anterior rotar los elementos k veces a la izquierda del vector A, imprimir el vector generado y luego el vector con las rotaciones respectivas.

Entrada:

9

k: 2

Salida:

Vector A:

99 10 76 95 68 49 69 79 2

Vector con Rotaciones:

79 2 99 10 76 95 68 49 69

11. Del ejercicio anterior tome el vector A y rote los elementos **pares** k veces a la derecha y los elementos **impares** k veces a la izquierda.

Entrada:

13

k: 2

Salida:

Vector A:

96 61 40 9 2 50 42 3 63 91 91 33 84

Vector con Rotaciones a la Derecha:

61 9 2 50 42 3 63 91 91 33 84 96 40

Vector con Rotaciones a la Izquierda:

91 33 96 61 40 9 2 50 42 3 63 91 84

12. Leer un vector de tamaño n, buscar el numero k dentro del vector y eliminar los múltiplos de k, además de ordenar el vector con cualquier método de ordenación que usted conozca.

Entrada:

10

8 6 13 1 3 5 7 2

k: 2

Salida:

1 3 5 7 13

- 13.** Leer un vector con N números positivos y mostrar los números primos que había en el vector conjuntamente con la posición en la que se encontraba.

Ejemplo de entrada

3 5 88 202 7 5

Ejemplo de Salida

3 0

5 1

7 4

5 5

- 14.** Dado un vector con N elementos mostrar el elemento que más se repite.

Ejemplo de entrada

1 2 3 4 5 6 7 4

Ejemplo de Salida

4

- 15.** Dado un vector de N elementos contar cuántos números primos hay en el vector.

Ejemplo de entrada

1 2 3 4 5 6 7 4

Ejemplo de Salida

4

- 16. Alex Jhon Y Las Monedas:** Alex Jhon tiene varios conjuntos de monedas de 1.Bs en forma de escalera con cantidades de “1, 2, 3, 4,...,n” colados con yurex, que quiere llevar en su mochila para ir a jugar en el Cyber Café con sus mejores amigos que son Cristofer y Juan Jose, pero se le hace imposible guardar todos los conjuntos de monedas de 1.Bs en su mochila.

Ya que su mochila tiene “n” bolsillos, donde cada bolsillo tiene una capacidad diferente para guardar.

Ahora Alex Jhon se pregunta cuantos conjuntos máximos de monedas de 1.Bs podrá guardar en su mochila.

Ejemplo:

Si Alex Jhon tiene 6 Bolsillos con capacidades de “5, 4, 4, 4, 4, 5”, también tendrá 6 conjuntos de monedas con cantidades de “1, 2, 3, 4, 5, 6”.

- En el Bolsillo que tiene una capacidad de 4 guardará un conjunto de 1 moneda de 1Bs.

- En el otro Bolsillo que tiene una capacidad de 4 guardará un conjunto de 2 monedas de 1Bs.

- En el otro Bolsillo que tiene una capacidad de 4 guardará un conjunto de 3 monedas de 1Bs.

- En el otro Bolsillo que tiene una capacidad de 4 guardará un conjunto de 4 monedas de 1Bs.

- En el otro Bolsillo que tiene una capacidad de 5 guardará un conjunto de 5 monedas de 1Bs.

- En el otro Bolsillo que tiene una capacidad de 5 ya no podrá guardar un conjunto de monedas ya que cada conjunto tendrá mayor a 5 monedas.

Entonces Alex Jhon solo puede guardar 4 conjuntos de monedas de 1Bs.

Ayuda a Alex Jhon a resolver este problema para que pueda ir al Cyber café con sus amigos.

Ejemplo de entrada

4
6
5 4 4 4 4 5
4
7 11 13 9
10
10 9 6 6 7 4 4 5 6 4
1
11

Ejemplo de Salida

5
4
9
1

17. Dados dos vectores de números A y B, se pide intercalar el mínimo con el máximo de ambos vectores A y B, en un solo vector además de poner los elementos que sean primos en nuevo vector en el orden requerido, se debe seguir buscando el mínimo o máximo, según se requiera, es decir si estamos buscando el máximo y este resulta ser primo, ya no se considera en la siguiente búsqueda y seguimos buscando el máximo de los elementos que aún se tengan.

Ejemplo: Sean $\{1,7,9,1\}$ y $\{5,5,8\}$ los vectores A y B respectivamente.

El mínimo es 1 en A

El máximo es 9 en A

El nuevo mínimo es 1 en A

El nuevo máximo es 8 en B

El nuevo mínimo es 5(primo, se lleva a otro vector) en B

El nuevo mínimo es 5(primo, se lleva a otro vector) en B

El nuevo mínimo es 7(primo, se lleva a otro vector) en A

Siendo los vectores de salida $\{1,9,1,8\}$ y $\{5,5,7\}$, en caso de no tener ningún elemento en alguno de los vectores resultantes imprimir -1

Entrada:

1 7 9 1
5 5 8

Salida:

1 9 1 8
5 5 7

Entrada:

1 3 4 2 5 1

1 7 13

Salida:

1 4 1 1

13 7 5 3 2 2

Matrices

1. Para un número entero $n > 0$, generar la siguiente matriz:

Si $n=5$:

0	0	0	0	1
0	0	0	1	1
0	0	1	1	1
0	1	1	1	1
1	1	1	1	1

Entrada:

4

Salida:

0 0 0 1

0 0 1 1

0 1 1 1

1 1 1 1

2. Dado un número entero $n > 0$, generar la siguiente matriz:

Si $n=6$:

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
0	0	1	1	0	0
0	1	0	0	1	0
1	0	0	0	0	1

Entrada:

5

Salida:

1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
0 1 0 1 0
1 0 0 0 1

3. Dada una cadena (palabra) generar la siguiente matriz:

Ejemplo cadena: "Datos"

D	a	t	o	s
a	a	-	-	-
t	-	t	-	-
o	-	-	o	-
s	-	-	-	s

Entrada:

Java

Salida:

J a v a
a a - -
v - v -
a - - a

4. Dada un número $n > 0$, generar la siguiente matriz:

si $n=6$:

A	B	C	D	E	F
L	K	J	I	H	G
M	N	O	P	Q	R
X	W	V	U	T	S
Y	Z	A	B	C	D
J	I	H	G	F	E

Entrada:

4

Salida:

A B C D
H G F E
I J K L
P O N M

5. Dados un entero positivo n , generar la siguiente matriz:
si $n=7$:

7	7	7	7	7	7	7
7	5	5	5	5	5	7
7	5	3	3	3	5	7
7	5	3	1	3	5	7
7	5	3	3	3	5	7
7	5	5	5	5	5	7
7	7	7	7	7	7	7

Entrada:

6

Salida:

6 6 6 6 6 6
6 4 4 4 4 6
6 4 2 2 4 6
6 4 2 2 4 6
6 4 4 4 4 6
6 6 6 6 6 6

6. Dada una matriz (M) cuadrada de tamaño n , generar una nueva matriz que sea el resultado de intercambiar las filas y las columnas de M, e invertir la diagonal principal.

Entrada:

4

1 3 4 2
4 5 2 8
2 3 0 1
2 0 3 7

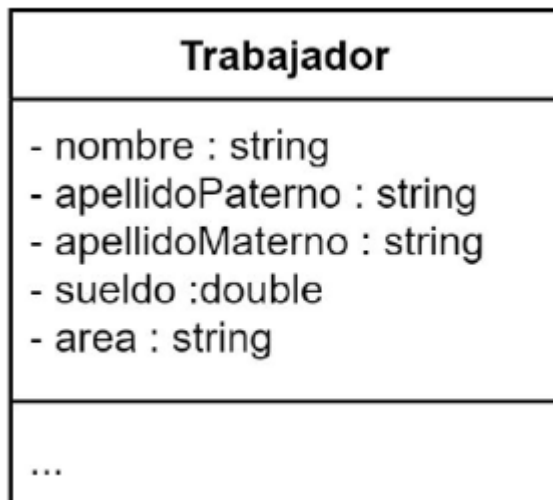
Salida:

7 4 2 2
3 0 3 0
4 2 5 3
2 8 1 1

Nueva Sección - POO

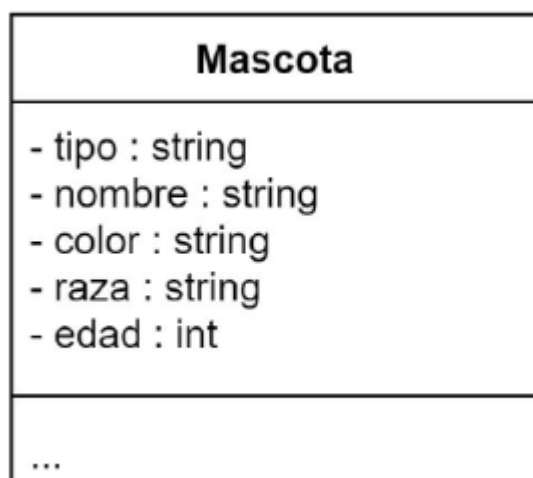
Objetos y Clases

1. Dado el siguiente diagrama de clase:



- Instanciar a 3 trabajadores.
- Mostrar a los 3 trabajadores.
- Mostrar al Trabajador que más sueldo recibe.
- Cambiar el área del trabajador 1 al área en la que se encuentra el trabajador 3.

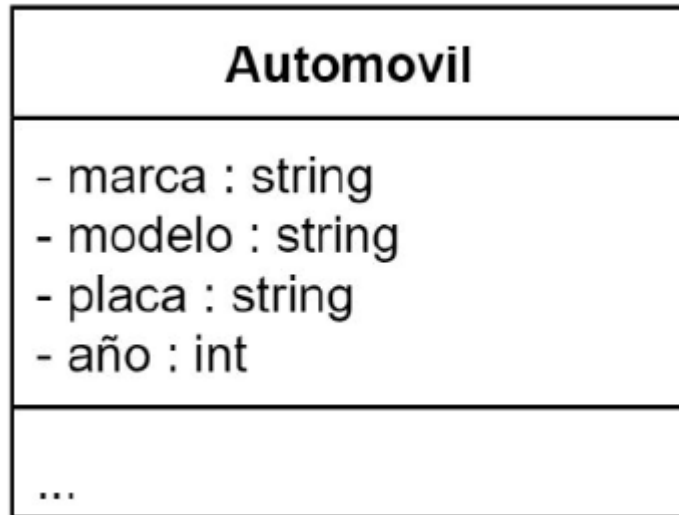
2. Dado el siguiente diagrama de clase:



- Instanciar “n” mascotas de diferente manera en un vector.
- Mostrar a todas las Mascotas de Tipo=”FELINO”.

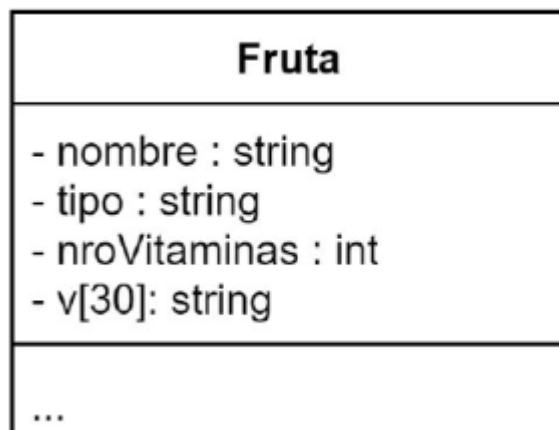
- c) Contar cuántas mascotas menores a 2 años existen.
- d) Mostrar a las Mascotas con nombre x.

3. Dado el siguiente diagrama de Clase:



- a) Instanciar 4 automóviles de diferente manera
- b) Verificar que automóvil tiene la placa x.
- c) Verificar si uno de los automóviles es de la marca x.
- d)Cuál es el automóvil más antiguo.

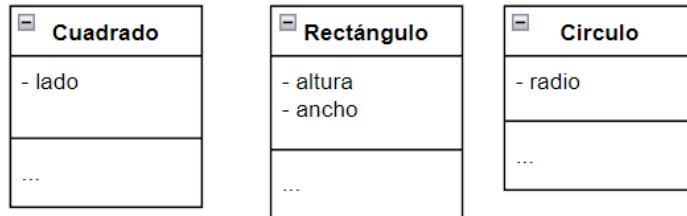
4. Dado el siguiente diagrama de clase:



- a) Instanciar de 3 maneras diferentes 3 frutas con al menos 4 vitaminas cada una.
- b) Verificar cuál es la fruta con más vitaminas.
- c) Mostrar las vitaminas que tiene la fruta x.
- d) Cuántas Frutas son cítricas.

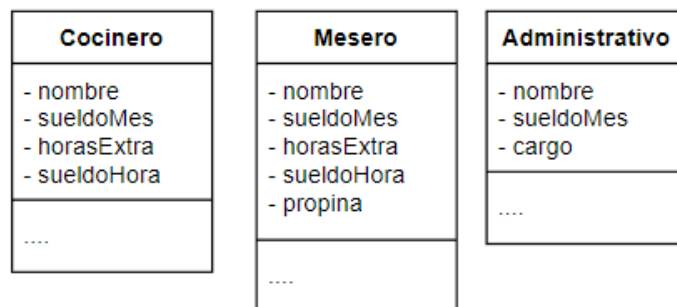
Polimorfismo

1. Sean las siguientes clases que hacen referencia a diferentes tipos de figura:



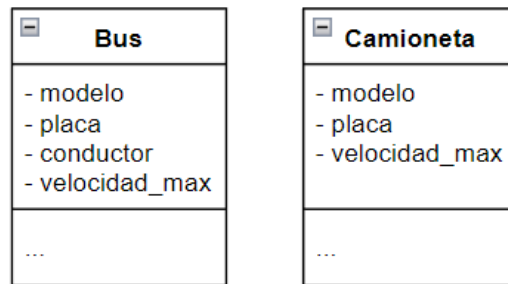
- a) Instanciar 1 Cuadrado, 1 Rectángulo y 1 Círculo
- b) Sobrecargar el método Área para mostrar el área de todas las figuras.

2. Un restaurante organiza a su personal mediante las siguientes clases:



- a) Instanciar 1 Cocinero, 2 objetos Mesero y 2 objetos Administrativo.
- b) Sobrecargar el metodo SueldoTotal para mostrar el sueldo total, sumándole las horas extra, considerando el sueldo por hora y la propina en caso de los meseros.
- c) Sobrecargar el método para mostrar a aquellos Empleados que tengan SueldoMes igual a X.

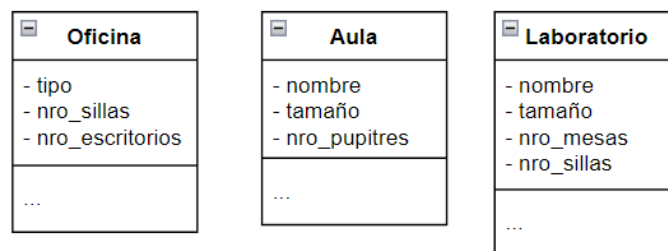
3. Sean las siguientes clases que detallan 2 tipos de automóviles:



a) Instanciar 2 objetos Bus y 1 objeto Camioneta

b) Sobrecargar el método MayorVelocidad, para mostrar la mayor velocidad de los objetos.

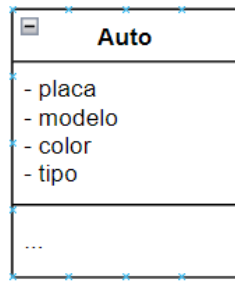
4. Se hace referencia a algunos de los diferentes ambientes de la Universidad mediante las siguientes clases:



a) Instanciar 2 objetos Oficina, 2 Aulas y 1 Laboratorio

b) Mostrar la cantidad total de muebles que tiene cada ambiente.

5. Para el comportamiento de los atributos de un Auto se tiene en cuenta la siguiente clase:



- a) Sobrecargar el constructor 2 veces
- b) Instanciar 2 objetos Auto, usando los diferentes constructores.
- c) Sobrecargar el operador ++ para tener un nuevo auto.
- d) Sobrecargar el operador -- para mostrar los atributos de los dos objetos

Observación ejercicio e: Realizar en C#

Herencia

1. **Sistema de figuras geométricas:** Crea una clase **Figura** que tenga métodos para calcular el área y el perímetro de diferentes figuras geométricas (**círculo**, **cuadrado**, **rectángulo**, **triángulo**, etc.). Luego, crea clases hijas para cada tipo de figura, que hereden los métodos de la clase padre y agreguen sus propias fórmulas para calcular el área y el perímetro de esa figura en particular.
2. **Sistema de vehículos:** Crea una clase **Vehículo** con atributos y métodos genéricos (por ejemplo, velocidad máxima, aceleración, frenado, etc.). Luego, crea clases hijas para diferentes tipos de vehículos (**coches**, **motos**, **camiones**, etc.), que hereden los métodos y atributos de la clase padre y agreguen sus propias características (por ejemplo, número de ruedas, capacidad de carga, etc.).

Herencia múltiple(en el caso de que el lenguaje no la soportara usar interfaces, y métodos abstractos)

3. **Sistema de robots:** Crea una (clase o interfaz) (Robot o IRobot) con métodos y atributos genéricos (por ejemplo, nivel de batería, capacidad de carga, movimiento, etc.). Luego, crea dos clases hijas: RobotDeCocina

y RobotDeLimpieza. Ambas clases deben heredar de la clase o interfaz (Robot o IRobot), pero también deben heredar de otra clase o interfaz llamada (AparatoElectrodomestico, IAparatoElectrodomestico), que tiene métodos y atributos relacionados con electrodomésticos en general (por ejemplo, encender, apagar, revisar si está funcionando correctamente, etc.). De esta forma, las clases hijas RobotDeCocina y RobotDeLimpieza tendrán acceso tanto a los métodos y atributos de Robot como a los de AparatoElectrodomestico y en el caso de las interfaces los métodos abstractos tendrán que ser implementados. Por ejemplo, el RobotDeCocina podría tener un método para preparar alimentos y también podría encenderse y apagarse como cualquier otro electrodoméstico, mientras que el RobotDeLimpieza podría tener un método para limpiar el suelo y también podría revisarse para asegurarse de que está funcionando correctamente.

