

Ejercicios de evaluación de programación sobre redes en Java

1. Introducción a Java

Test de autoevaluación

- 1) ¿Qué es el código de ensamblaje y el programa traductor ensamblador?
- 2) ¿Qué es la abstracción en informática?
- 3) Explique los diferentes niveles de abstracción en lenguajes de programación y de algunos ejemplos de lenguajes por nivel de abstracción
- 4) Defina interprete, compilador y explique la diferencia entre estos dos conceptos
- 5) ¿Qué es un paradigma de programación? Nombre algunos de ellos
- 6) Explique que es el paradigma orientado a objetos
- 7) ¿Cómo es que Java es tan popular en tecnologías webs, si cuando se creó el lenguaje no existía Internet tal y como se la conoce en la actualidad?
- 8) Liste las características más relevantes del lenguaje y desarrolle brevemente cada una de ellas
- 9) ¿Cómo es que Java es multiplataforma y al mismo tiempo sus programas se ejecutan de una manera eficiente respecto al tiempo de ejecución?
- 10) ¿Qué es y cómo funciona la máquina virtual de Java?

2. Elementos básicos del lenguaje

Test de autoevaluación

- 1) ¿Qué es un linker o cargador, y para qué sirve?
- 2) ¿Cuáles son las tres zonas de memoria utilizadas por Java?
- 3) Explique para qué sirve la zona de memoria de datos
- 4) Explique para qué sirve la zona de memoria conocida como pila o stack
- 5) Explique para qué sirve la zona de memoria conocida como montículo o heap
- 6) ¿Qué es un atributo y de qué tipos de datos puede ser definido?
- 7) ¿Cuál es la diferencia entre tipos de datos primitivos y de referencia? Explique cada una de estas categorías
- 8) El tipo de dato String, ¿Es un tipo de dato primitivo o de referencia? Explique las particularidades correspondientes al mismo
- 9) Qué es la conversión entre tipos de datos y en qué casos resultaría útil hacer este tipo de conversión. Detalle los problemas que pueden surgir al trabajar con conversiones
- 10) Liste y explique las convenciones de escritura adoptado por el común denominador de programadores

Ejercicios

- 1) Indique el valor de $x = -1 + 5 / 3 \% 2 * 5 + 7 * 2 + 1$
- 2) Indique el valor de $x = -1 + (5 / 3) \% 2 * (5 + 7) * 2 + 1$
- 3) Indique el valor de $x = 3 > 2 \ \&\& \ 6 < 10 \wedge \text{true}$
- 4) Se poseen 4 atributos:
 - a de tipo short
 - b de tipo long
 - c de tipo float
 - d de tipo String

Escriba los pasos necesarios para guardar:

- a en b
- b en c
- c en d
- d en a

- 5) Indique el valor de que tendrá x si se tiene que:

```
1 x = 5;
2 x *= x%2*3+2;
```

- 6) Indique el valor que tendrá x en cada una de las líneas

```
1 x = 1;
2 ++x;
3 x += x++;
4 --x;
```

- 7) Indique el valor resultante de cada una de las líneas

```
1 !(true ^ 10 > 5) & !(true || !(false ^ false))
2 (10 < 5) && (false) || (false == true) ^ 8.3f >= 8.3f
3 5 < 11 ^ 5 > 15 | 2 == 2 & false ^ !true ^ 10 <= 10
4 !(true && true || false ^ !(4==3) & 3 > 1 ^ !true)
```

8) Indique el valor resultante de cada una de las líneas

```

1  int i = 43;
2  int mascara = 32;
3  int bit = i & mascara;
4  bit >>= 5;
5  mascara ^= 1;
6  bit |= i & mascara;
7  bit <<= 1;
8  mascara &= (~bit) >>> 3;
9  bit += (~(~0)<<1);
10 mascara += bit + i;

```

9) Leyendo los códigos correspondientes de la tabla de código ASCII descifre el mensaje

```

1  String mensaje;
2  char caracter;
3  int encriptado = 79 >> 2;
4
5  encriptado = (~((-160) >> 3) << 2) + 3;
6  caracter = (char) encriptado;
7  mensaje = String.valueOf(caracter);
8
9  encriptado = (((encriptado >> 1) ^ 30) << 1) -7;
10 caracter = (char) encriptado;
11 mensaje += String.valueOf(caracter);

```

10) Si un color está representado por un entero en el cual, el byte más significativo corresponde al canal rojo, y los 3 bytes siguientes corresponden a los canales verde, azul y alfa respectivamente, genere un algoritmo que separe los valores rojo, verde, azul y alfa, y los guarde en cuatro atributos que correspondan a estos.

Luego averigüe estos valores para el siguiente color:



Para realizar este ejercicio debe utilizar operadores de bit.

Tabla de caracteres y símbolos ASCII

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☹	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	\$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	▣	BS	040	(072	H	104	h
009	(tab)	HT	041)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♪	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	▶	DLE	048	0	080	P	112	p
017	◀	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	!!	DC3	051	3	083	S	115	s
020	π	DC4	052	4	084	T	116	t
021	\$	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↑	ETB	055	7	087	W	119	w
024	↕	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[123	{
028	(cursor right)	FS	060	<	092	\	124	
029	(cursor left)	GS	061	=	093]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	☐

Copyright 1998, Jim Price Com Copyright 1992, Loading Edge Computer Products, Inc.

3. Estructuras, arreglos y excepciones

a. Estructuras

- No se utilizan vectores
- No se utilizan métodos
- Verificar siempre que el usuario ingresa el o los datos que se le piden de manera correcta.

I. Nivel Inicial

- 1) Crear un programa que muestre por consola la frase "Hola mundo".
- 2) Crear un programa que pida al usuario nombre y apellido y los muestre por consola.

- 3) Crear un programa que pida al usuario dos números enteros, y que imprima la suma, producto, diferencia, cociente y resto de los números.
- 4) Crear un programa que pida al usuario dos números enteros y que imprima un mensaje indicando si el primer número es mayor, menor o igual al segundo.
- 5) Crear un programa que pida al usuario tres números enteros y muestre la suma, promedio, producto, menor y mayor de esos números.
- 6) Crear un programa que pida al usuario cinco números enteros y que determine e imprima los enteros mayor y menor en el grupo.
- 7) Crear un programa que informe si un número ingresado es par o impar.
- 8) Crear un programa que pida dos números enteros al usuario, y que informe si el primero es un múltiplo del segundo o no.
- 9) Crear un programa que reciba cinco números, y que determine e imprima la cantidad de números negativos, positivos, y la cantidad de ceros recibidos.
- 10) Crear un programa que muestre los primeros 100 números pares.
- 11) Crear un programa que calcule e informe el valor de n para el cual la siguiente suma excede a un valor x : $1 + 2 + 3 + 4 + \dots + n$. Por ejemplo para un $x = 20$, el número que excede es $n = 6$, ya que $1 + 2 + 3 + 4 + 5 + 6 = 21$.
- 12) Crear un programa que imprima los primeros 100 números naturales y la suma de los mismos.
- 13) Crear un programa que determine e informe los primeros n números primos, comenzando de un número x . Se debe leer x y n .
- 14) Crear un programa que pida al usuario ingresar un carácter, y dado este, informar si es una vocal o no.
- 15) Crear un programa que dado un carácter determine e informe si este es un dígito entre 0 y 9.
- 16) Crear un programa que multiplique dos números por medio de sumas sucesivas e informe el resultado.
- 17) Crear un programa que divida dos números por medio de restas sucesivas e informe el resultado.

- 18) Crear un programa que cuente e informe la cantidad de un número entero que ingrese el usuario.
- 19) Crear un programa que calcule e informe la suma de la progresión geométrica: $1 + x + x^2 + \dots + x^n$.
- 20) Crear un programa que calcule e informe la serie: $1/2 + 2/2^2 + 3/2^3 + \dots + n/2^n$.

II. Nivel básico

- 1) Crear un programa que informe su peso efectivo en la luna, sabiendo que este será un 17% respecto a su peso en la tierra.
- 2) Crear un programa que pida al usuario un número entero de 5 dígitos, y que muestre por consola el resultado de la suma de todos sus dígitos. En el caso de que se ingrese un número de más dígitos o de menos dígitos informar el error y volver a pedir el número.
- 3) Crear un programa que muestre los cuadrados y los cubos de los números del 0 al 10. El resultado debe mostrarse como figura en la siguiente imagen.

número	cuadrado	cubo
0	0	0
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

- 4) Crear un programa que muestre por consola los caracteres ASCII que se encuentran entre los valores 30 y 126 inclusive. Los caracteres se deben mostrar en 7 filas de 15 columnas.

El resultado debe ser como se muestra en la siguiente imagen.

```

! " # $ % & ' ( ) * + , -
. / 0 1 2 3 4 5 6 7 8 9 : ; <
= > ? @ A B C D E F G H I J K
L M N O P Q R S T U V W X Y Z
[ \ ] ^ _ ` a b c d e f g h i
j k l m n o p q r s t u v w x
y z { | } ~

```

- 5) Crear un programa que escriba los primeros 20 números de la sucesión de Fibonacci sabiendo que $x_1 = 1$, $x_2 = 1$ y $x_n = x_{n-1} + x_{n-2}$ para $n > 2$.
- 6) Si la población actual fuera 7309784505 y el crecimiento poblacional por año fuera 24807909, realice un programa que estipule la población mundial luego de uno, dos, tres, cuatro y cinco años.
- 7) Crear un programa que pida al usuario ingresar un valor en pesos, e informe la cantidad mínima de billetes y monedas necesarias para la cantidad ingresada.
- 8) Crear un programa que reciba del usuario el radio de un círculo como un entero, y que imprima el diámetro, la circunferencia y el área del círculo mediante el uso del valor de punto flotante 3.14159 para π , en donde diámetro = $2r$, circunferencia = $2\pi r$ y área = πr^2 .
- 9) Crear un programa que calcule e informe el factorial de un número n . El factorial de un número n es: $n! = 1 \times 2 \times 3 \times \dots \times n$.
- 10) Crear un programa en el que se ingrese el valor de un préstamo a obtener, el porcentaje de interés anual de ese préstamo y el valor que desea abonar por mes. Mediante esos datos calcule e informe cuanto tiempo en años y en meses, se tardará en pagar el préstamo.
- 11) Crear un programa que determine e informe el monto a pagar por una llamada telefónica, teniendo en cuenta lo siguiente: Toda llamada que dure hasta 5 minutos tiene un costo de 5 pesos. Por cada minuto adicional se cobrará 50 centavos. Se debe leer el tiempo de la llamada.
- 12) Crear un programa que determine e informe si un año es bisiesto. Un año es bisiesto si es múltiplo de 4, excepto los múltiplos de 100 que no son bisiestos, salvo que a su vez también sean múltiplos de 400. (1800 no es bisiesto, 2000 si).
- 13) Crear un programa para determinar e informar si un número entero es capicúa (número que se escribe igual al derecho y al revés). No utilice cadenas.
- 14) Crear un programa que dada una cadena de fecha en el formato "dd mm aaaa" devuelva su correspondiente en texto. Ej.: Para 21 04 2015 el programa debe informar "21 de abril de 2015".
- 15) Crear un programa que lea un número entero n mayor a cero que identifica una cantidad de segundos, y calcule e imprima el número de horas, minutos y segundos contenidos en ella. Ejemplo: Para $n = 15723$ se debe imprimir 4 horas, 22 minutos y 3 segundos.

- 16) Crear un programa que dada una hora en formato hh:mm:ss determine e informe cual será el tiempo dentro de n segundos. La cantidad de segundos debe ser indicada por el usuario.
- 17) Crear un programa que resuelva cualquier ecuación de segundo grado $ax^2 - bx = c$. Por tanto, se debe pedir al usuario que introduzca los valores a, b y c, y el programa indicará las posibles raíces solución.
- 18) Crear un programa que pida al usuario un número entero y lo convierta a binario.
- 19) Crear un programa que pida al usuario una velocidad en Km/h, la convierta a m/s e informe por consola la conversión.
- 20) Crear un programa en donde se ingrese la variedad de artículos vendidos y luego de eso, se ingrese la cantidad de artículos por cada ocurrencia, y el precio del artículo. Mostrar por consola a modo de factura, Nro de artículo, cantidad, precio unitario, y al final de la misma, total a pagar.

III. Nivel medio

- Utilizar el operador if ternario siempre que sea posible

- 1) Crear un programa que pida al usuario escoger entre dos opciones:
 - a. Ingresar peso en libras y altura en pulgadas
 - b. Ingresar peso en kilogramos y altura en metros

Una vez seleccionada la opción, pedir al usuario que ingrese peso y altura, y según esos datos calcular el índice de masa corporal. Si el índice de masa corporal es menor a 18.5 se debe informar “bajo peso”, si esta entre 18.5 y 24.9 se debe informar “normal”, si esta entre 25 y 29.9 se debe informar “sobrepeso”, y si es mayor a 30 se debe informar “obeso”.

Para calcular la masa corporal se debe hacer

- $\text{masaCorporal} = (\text{pesoEnLibras} \times 703) / (\text{alturaEnPulgadas} \times \text{alturaEnPulgadas})$
ó
- $\text{masaCorporal} = \text{pesoEnKilogramos} / (\text{alturaEnMetros} \times \text{alturaEnMetros})$

- 2) Crear un programa para calcular los salarios semanales de unos empleados a los que se les paga 180 pesos la hora si éstas no superan las 35 horas. Cada hora por encima de las 35 se considerará extra y se paga a 240 pesos. La cantidad de horas no puede superar las 84.

Para calcular e informar el sueldo de un empleado se debe pedir la cantidad de horas trabajadas por el mismo. Además el programa debe preguntar si se desea calcular otro salario, si es así el programa se vuelve a repetir.

3) Crear un programa que muestre un menú con las opciones:

- Sumatoria
- Factorial
- Logaritmo natural
- Salir

Según la opción elegida, a excepción de la opción salir, se debe pedir un número y con el mismo se debe realizar la sumatoria, el factorial o el logaritmo natural del mismo. Una vez hecho el cálculo se debe volver al menú y volver a pedir que se ingrese una opción hasta que la misma sea salir.

4) Modificar el ejercicio N°20 de los ejercicios nivel básico, de modo que, para calcular el total, se pida ingresar el IVA, pudiendo ser este de un 4%, 7%, 16% o 21%, y en cualquier otro caso, se deberá rechazar el IVA y volver a pedirlo.

Además se debe aplicar un descuento, en función a la suma de los importes, siendo de un 0% si el importe final es menor a 1000, de un 5% si el importe final es mayor o igual que 1000 y menor que 10000, y de un 10% si el importe final es mayor o igual que 10000. El descuento se debe aplicar al importe final a pagar con IVA incluido.

Para finalizar se debe imprimir por pantalla el IVA a pagar en porcentaje y valor, el total a pagar sin IVA y sin descuentos, el total a pagar con IVA y sin descuentos, el descuento realizado, y el importe total a pagar con IVA y con el descuento realizado.

5) Crear un programa que adivine un número entre 1 y 1000 pensado por el usuario. El programa debe ir mostrando números y el usuario debe responder a cada número con un símbolo ">", "<" o "=", según el número pensado sea menor, mayor o igual que el mostrado por la computadora. Cuando se adivine el número se deberá mostrar un mensaje especificando cuantas preguntas ha necesitado para adivinar el número. No se debe realizar más de 10 preguntas, y se debe detectar si el usuario está haciendo trampa respondiendo con imposibles, como por ejemplo, es mayor que 4 y menor que 5.

6) El Miércoles de Ceniza es 46 días antes que el domingo de resurrección. Crear un programa que dados el día D, el mes M y el año A, calcule e informe la fecha del miércoles de Ceniza en ese año. Se debe tener en cuenta si el año es o no bisiesto.

7) Se dispone de una máquina expendedora de café que acepta y da vueltos en monedas de 2 pesos, 1 peso, 50 centavos, 25 centavos, 1 centavos y 5 centavos. Realizar un programa que dado el precio del café a consumir y la cantidad de dinero entregada por el consumidor, informe el vuelto a entregar empleando la menor cantidad de monedas posibles. Debe verificar que el precio del café pueda pagarse con las monedas expuestas y que el usuario no ingrese monedas demás.

8) Realizar un programa que muestre un menú con las siguientes opciones:

- Ingresar un número entero
- Ingresar un número binario
- Ingresar un número octal
- Ingresar un número hexadecimal
- Ingresar un número romano
- Salir

Una vez ingresada la opción se debe pedir al usuario que ingrese el número según la opción que haya ingresado, y se debe verificar que el número ingresado sea correcto dentro del sistema elegido. Luego se debe mostrar otro menú con las siguientes opciones:

- Convertir a entero
- Convertir a binario
- Convertir a octal
- Convertir a hexadecimal
- Convertir a números romanos
- Atrás

Con excepción del número desde el que se quiere hacer la conversión. Ejemplo: Si se ha elegido la opción ingresar un número octal, en el segundo menú no debe aparecer la opción convertir a octal. Una vez elegida la opción deseada se debe informar la conversión por consola y se debe volver a este menú por si se desea realizar otra conversión. Si se desea ingresar otro número se debe elegir la opción atrás para que se vuelva a mostrar el primer menú y si se desea salir del programa se debe elegir la opción salir.

9) Realizar un programa que genere un calendario. Se debe pedir al usuario que ingrese la fecha actual en formato dd mm aaaa, y mediante este se debe generar el calendario mostrando todos los días del mes actual. Así mismo debe existir un menú con las opciones ingresar fecha, mes siguiente, mes anterior y salir.

10) Crear un programa que determine e informe en cuantos números de 4 cifras se encuentran más números 1 que 2. Debe existir un 1 y un 2 para considerarlo.

11) Crear un programa que permita el ingreso de n ternas de valores enteros. Se debe calcular e informar:

- De cada terna el mayor valor y su orden en la terna.
- De cada terna cuantos valores son múltiplos de 7. Ídem de todas las ternas
- De todas las ternas el promedio del primer valor de cada una

12) Crear un programa que permita al usuario ingresar las coordenadas "x" e "y" de 10 puntos del plano, cuyos valores serán en punto flotante y deben ser distintos a 0.

Determinar e informar:

- Para cada punto ingresado, las coordenadas y el cuadrante en donde se haya ubicado.
- En cual o cuales cuadrantes no se encontró ningún punto.
- En que cuadrante se ubica el punto que tiene la máxima distancia al origen de coordenadas.
- Dibujar mediante caracteres el sistema de coordenadas y los puntos en sus correspondientes coordenadas, siempre y cuando ninguna coordenada supere los valores 50 en x tanto como en y. Dibujar en color rojo el punto que tiene la máxima distancia al origen de coordenadas.

13) Una empresa posee 8 fábricas y emite una tarjeta por cada orden de trabajo que recibe, con los siguientes datos:

- Número de fabrica
- Número de orden

Para finalizar el ingreso de orden de trabajos se debe indicar 10 como número de orden.

El programa debe determinar e informar:

- Cantidad total de trabajos realizados por cada fábrica
- Cantidad total de trabajos realizados por todas las fábricas
- Número de fábrica que realizo la mayor cantidad de trabajos

Para realizar este ejercicio debe definir un atributo entero que servirá para contabilizar la cantidad de trabajos realizados por cada fábrica, tomando 4 bits del atributo entero por cada fábrica. Esto quiere decir que una fábrica no podrá poseer más de 15 trabajos realizados.

Para determinar los puntos se deben utilizar operadores de bit.

14) Corre el año 2133. La tierra ha sido invadida por seres alienígenos, encabezados por su terrible y malvado líder “Noicamargorp”, quien lo mantiene cautivo a usted en una celda custodiada por el androide 620109. A su lado, se encuentra el cuerpo inerte de un ex-prisionero, que fue capturado mucho antes que usted. Entre sus pertenencias, usted encuentra la clave para desactivar a 620109 y así escapar y salvar a la humanidad del cruel “Noicamargorp” (de quien se dice se encarga de destruir principalmente a estudiantes de programación, con sus sádicos “Senemaxe”). En el manuscrito, se señala que para desactivar a 620109 es necesario ingresar la cantidad de pasos necesarios para reducir exactamente N números, los cuales van apareciendo en la pantalla del androide, siguiendo la siguiente regla:

- Si el número es par, dividirlo en dos.
- Si el número es impar, multiplicarlo por tres y sumarle 1.

Se debe repetir este proceso hasta que el número ingresado se transforme en 1.

Para poder solucionar el problema, usted deberá escribir un algoritmo que le permita ingresar N números enteros mayores o iguales a 1 (en caso de ingreso de un número menor que 1, se debe imprimir un mensaje de error), aplicar la regla anterior, mostrar los resultados parciales, y calcular la cantidad de pasos necesarios para convertir cada número en 1. Finalmente, deberá mostrar el número total de pasos necesarios para convertir en 1 cada uno de los N números ingresados. Por ejemplo, la salida para la primera iteración de su programa, suponiendo que el número ingresado es 20 sería:

Valor inicial: 20
 Valor siguiente: 10
 Valor siguiente: 5
 Valor siguiente: 16
 Valor siguiente: 8
 Valor siguiente: 4
 Valor siguiente: 2
 Valor siguiente: 1
 Valor final es 1, numero de pasos necesarios: 7

Esta salida se genera por cada uno de los N números ingresados, y al final, se debe mostrar la suma de todos los pasos necesarios.

15) Crear un programa en el cual se le pida al usuario datos correspondientes a la aprobación de una materia universitaria. Los datos a ingresar son:

- Cantidad de días que se cursa la materia a la semana, y respecto a eso se debe calcular cuantas clases se tendrán en un cuatrimestre.
- Por alumno, cantidad de clases presentes.
- Por alumno dos notas correspondientes a dos parciales.

Para no desaprobado una materia se debe contar con un 75% de asistencia y en el caso de que las notas sean:

- Mayores a 7, el alumno promociona la materia
- Entre 4 y 6, el alumno aprueba la materia pero no la promociona
- Menor a 4, el alumno desaprueba la materia

Informar por cada alumno ingresado si promociona, aprueba o desaprueba la materia, y la razón. Además mostrar la cantidad y el promedio de alumnos promocionados, aprobados y desaprobados. En el caso de desaprobación de la materia la cantidad y promedio de desaprobaciones por notas, y por asistencia, y cantidad y promedio que desaprobaron la primera prueba, la segunda y las dos.

Finalmente realice con caracteres gráficos de barras correspondientes a los datos que se deben informar.

16) Crear un programa que lea tres longitudes en centímetros y determine si forman o no un triángulo. Si forman un triángulo determinar si se trata de un triángulo

- Equilátero (si tiene los tres lados iguales)
- Isósceles (si tiene dos lados iguales)
- Escaleno (si tiene tres lados desiguales)

Para formar un triángulo se requiere que el mayor de los lados sea menor a la suma de los otros dos lados.

Además calcular e informar los ángulos del triángulo sabiendo que

- $\sin \alpha = \text{opuesto} / \text{hipotenusa}$
- $\cos \alpha = \text{adyacente} / \text{hipotenusa}$
- $\tan \alpha = \text{opuesto} / \text{adyacente}$

Por último informar si el triángulo es

- Acutángulo (tres ángulos agudos)
- Obtusángulo (un ángulo obtuso)
- Rectángulo (un ángulo recto)

17) Crear un programa que dado un número entre 0,0001 y 0,9999 (y de no más de 4 cifras decimales), obtenga y muestre la correspondiente fracción irreducible.

Por ejemplo, el número 0,25 se puede obtener a partir de 25/100, o de 2/8, o de 1/4, entre otros. La fracción irreducible es 1/4, que está formada por un numerador y un denominador que son primos entre sí.

18) Unos grandes almacenes encargan a un fabricante pantalones y camperas deportivas.

El fabricante dispone para la confección de 750 m de tejido de algodón y 1000 m de tejido de poliéster. Cada pantalón precisa 1 m de algodón y 2 m de poliéster. Para cada chaqueta se necesitan 1.5 m de algodón y 1 m de poliéster.

El precio del pantalón se fija en \$500 y el de la chaqueta en \$400.

Realizar un programa que calcule e informe que número de pantalones y chaquetas debe suministrar el fabricante a los almacenes para que estos consigan un beneficio máximo.

19) Escribir un programa que haga una serie de Taylor de la función seno, sabiendo que la función seno desarrollada en serie de Taylor tiene la forma

$$\sin(x) = \sum_{i=0}^n (-1)^i \frac{x^{2i+1}}{(2i+1)!}$$

- 20) Una oruga está situada en la base de una rama de un metro de largo. Su objetivo es llegar al final de la rama. Para ello, durante el día camina una distancia de 1 cm y por la noche descansa. Y es entonces, durante la noche, que la rama crece 1 metro de forma uniforme. Es decir, que en el segundo día, la oruga al despertar se encuentra a 2 cm. de la base pero la rama mide 2 metros. Y así sucesivamente.
- Crear un programa que calcule e informe exactamente el número de días que tardaría la oruga en llegar al final de la rama. Y más importante aún, que el programa sea lo más óptimo posible.

IV. Nivel avanzado

- 1) Realizar un programa para resolver el siguiente problema: Un granjero, un lobo, una gallina y un saco de maíz deben cruzar el río. Para ello se dispone de un bote con una capacidad de dos ocupantes, el hombre (el único que sabe remar) y otro. Si de un lado del río se quedan solos el lobo y la gallina, el lobo se come a la gallina. Si en una orilla se quedan solos la gallina y el maíz, la gallina se come al maíz.
- 2) Realizar un programa para resolver el siguiente problema: Clara, Luisa, María y Nélida son cuatro mujeres que aman sus trabajos. Ellas trabajan como diseñadora de moda, florista, jardinera y directora de orquesta. Cada mujer tiene un solo trabajo, y cada trabajo es ocupado por una sola mujer. Con las siguientes pistas, encontrar el trabajo realizado por cada mujer:
 - (a) Clara es violentamente alérgica a las plantas.
 - (b) Luisa y la florista comparten el departamento
 - (c) A María y Luisa les gusta solamente la música rock
 - (d) La jardinera, la diseñadora de modas y Nélida no se conocen entre sí
- 3) Realizar un programa para resolver el siguiente problema: El Sr. Ido, el químico, tiene seis frascos llenos de líquidos coloreados. Hay uno de cada color: rojo, anaranjado, amarillo, verde, azul y violeta. El señor Ido sabe que algunos de esos líquidos son tóxicos, pero no recuerda cuales... Sin embargo, sí recuerda algunos datos. En cada uno de los siguientes pares de frascos hay uno con veneno y otro no:
 - a) los frascos violeta y azul
 - b) los frascos rojo y amarillo
 - c) los frascos azul y anaranjadoEl Sr. Ido recuerda también que en estos otros pares de frascos hay uno sin veneno:
 - d) el violeta y el amarillo
 - e) el rojo y el anaranjado
 - f) el verde y el azul

¡Ah! Casi lo olvido, añade el Sr. Ido, el líquido del frasco rojo no es venenoso. ¿Qué frascos tienen veneno?

- 4) Realizar una calculadora en la cual se deberá ingresar una expresión, se analizará si la expresión ingresada es válida y se resuelva dicha expresión. Las operaciones de la calculadora son: Suma, resta, multiplicación, división, potenciación y radicación.
- 5) Se sabe que el operador << desplaza dos bits hacia la izquierda, por lo que si se realiza por ejemplo $2 \ll 2$ se obtiene como resultado 8. Al hacer $2 \ll -2$ el resultado es -2147483648, y al hacer $12 \ll -2$ el resultado es 0. ¿Por qué sucede esto?

b. Arreglos

- No se utilizan métodos
- Verificar siempre que el usuario ingresa el o los datos que se le piden de manera correcta.

I. Nivel Inicial

- 1) Crear un programa que pida 10 números enteros al usuario, los guarde en un vector, y finalmente muestre los datos del vector indicando los índices correspondientes a cada valor.
- 2) Crear un programa que llene un arreglo con los números enteros comprendidos entre 4 y 14, y luego lo muestre por consola indicando los índices correspondientes a cada valor.
- 3) Crear un programa que llene un arreglo con los números pares comprendidos entre 1 y 100 y luego los muestre por consola indicando los índices correspondientes a cada valor.
- 4) Crear un programa que llene un arreglo con los números comprendidos entre 0 y 100 divisibles por 3 y luego los muestre por consola indicando los índices correspondientes a cada valor.
- 5) Crear un programa que llene un arreglo con cinco números enteros consecutivos y haga una copia de ese arreglo en otro.
- 6) Crear un programa que llene un arreglo con los veinte primeros números pares y calcule su suma, informando el resultado.

- 7) Crear un programa que solicite cinco números, los almacene en un arreglo y luego calcule el promedio de esos números informando el resultado.
- 8) Crear un programa que tras asignar los números, -2, 5, 8, -9, 10, 15 y -4 a un arreglo calcule e informe, independientemente, la suma de los elementos positivos y la de los elementos negativos.
- 9) Crear un programa que almacene en un arreglo los números primos comprendidos entre 1 y 100, y luego los muestre por consola indicando los índices correspondientes a cada valor.
- 10) Crear un programa que informe el número de vocales de una cadena introducida por el usuario.
- 11) Crear un programa que lea una cadena de caracteres introducida desde el teclado por el usuario y la muestre al revés.
- 12) Crear un programa que determine e informe si una cadena de caracteres introducida por el usuario es o no un palíndromo, es decir, una palabra que se lee igual al derecho que al revés.
- 13) Crear un programa en donde se almacenen los números del 1 al 100 en un vector, y se informe por consola los números pares e impares.
- 14) Crear un programa en donde se pida al usuario la dimensión del vector a crear, y luego se pida al mismo que ingrese números enteros correspondientes a cada posición del vector. Se debe ordenar el mismo en forma creciente y luego mostrar el vector ordenado indicando los índices correspondientes a cada valor.
- 15) Modificar el ejercicio anterior para calcular e informar el valor máximo y mínimo del vector.
- 16) Modificar el ejercicio anterior para calcular e informar la cantidad de números negativos, positivos y ceros.
- 17) Crear un programa que mediante un menú permita consultar, asignar, modificar y eliminar datos de un vector de tipo entero de veinte posiciones. Para realizar las operaciones se debe ingresar la posición del vector a la cual se desea acceder.
- 18) Crear un programa que reciba los datos para rellenar un arreglo de 10 posiciones con números enteros, y que copie las últimas 5 posiciones en otro arreglo. Informar el arreglo resultante de la copia de valores indicando los índices correspondientes a cada valor.

- 19) Crear un programa para leer por teclado la nota de N alumnos de una clase y almacenarlas en un vector. Calcular la nota promedio de la clase y mostrar los alumnos por encima del promedio.
- 20) Crear un programa que pida ingresar los datos de N empleados. Por cada empleado se pide nombre y sueldo, que deben almacenar en dos vectores. Luego de eso calcular e informar el empleado con mayor sueldo.

II. Nivel básico

- Utilice la instrucción for mejorada siempre que sea posible

- 1) Crear un programa que pida N números enteros al usuario, los guarde en un vector y luego construir otro vector en donde se copien los valores del primer vector ignorando los números repetidos. Informar los valores del primer y segundo vector con sus índices correspondientes.
- 2) Crear un programa que pida dos enteros n y m. Crear dos vectores con dimensiones n y m, y pedir al usuario que ingrese los números correspondientes para dichos vectores. Luego de eso copiar los dos vectores a un tercer vector, y mostrar el vector resultante en orden decreciente.
- 3) A partir de dos números leídos p y q construir un vector tal que sus elementos sean sucesivamente: p, q, p+q, q+ (p+q), (p+q) + (q+(p+q)), ..., donde cada nuevo término es la suma de los anteriores, hasta que la suma de todos ellos, supere un número m. Imprimir p, q y el vector resultado.
- 4) Sean "nombre" y "sexo" dos vectores que contienen el nombre y el sexo de cada uno de los miembros de un club de fútbol. Masculino y femenino se denotan mediante "m" y "f", respectivamente. Generar dos nuevos vectores, llamados "varones" y "mujeres", de forma tal que varones contenga los nombres de todos los varones y mujeres contenga los nombres de todas las mujeres.
- 5) Se tienen dos vectores "a" y "b" de n elementos. Crear los vectores "c", "d" y "e" tales que:
 - "c" contendrá todos los elementos comunes a los vectores "a" y "b". (intersección)
 - "d" contendrá todos los elementos de "a" y "b" (unión)
 - "e" contendrá todos los elementos de "a" y "b" que no son comunes a ambos.

- 6) Crear un programa que le pida al usuario la dimensión de un vector, los valores enteros del mismo y un número entero k. Si $k = 1$ se debe ordenar el vector descendientemente, si $k > n$ se debe ordenar ascendentemente, y si $1 < k < n$ se debe ordenar en forma creciente los primeros k elementos y en forma decreciente los restantes.
- 7) Crear un programa que pida al usuario rellenar un vector entero de N posiciones, y luego crear un menú para buscar un valor dentro del vector, indicando por consola si se ha encontrado o no, y en el caso de encontrarse en qué posición del vector se encuentra.
- 8) Crear un programa que reciba dos matrices A y B de números reales de dimensiones M x N. Realice la suma de estas matrices y almacene el resultado en una matriz C. Muestre por consola las matrices de la siguiente forma:

$$\begin{bmatrix} 2 & 2 & 1 \\ 3 & 2 & 1 \\ 2 & 3 & 2 \\ 2 & 0 & 4 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 4 \\ 1 & 4 & 0 \\ 2 & 1 & 1 \\ 0 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 5 \\ 4 & 6 & 1 \\ 4 & 4 & 3 \\ 2 & 2 & 6 \end{bmatrix}$$

- 9) Crear un programa que determine si una matriz de M x N ingresada por el usuario es simétrica. Una matriz simétrica es aquella en donde la matriz A = transpuesta de A.

Ejemplo de matriz transpuesta:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & 2 & -1 \\ 3 & -2 & 0 & 4 \\ 4 & 3 & 1 & 2 \end{bmatrix} \quad A^t = \begin{bmatrix} 1 & 0 & 3 & 4 \\ 2 & -1 & -2 & 3 \\ 3 & -2 & 0 & 1 \\ 4 & 3 & 4 & 2 \end{bmatrix}$$

- 10) Crear un programa que verifique e informe si una matriz cuadrada de números enteros es mágica, en el caso de no serlo listar la suma de cada una de sus filas y columnas, así como de sus diagonales principales. Una matriz mágica es aquella donde la suma de cada una de sus filas, columnas y diagonales tienen el mismo valor.

- 11) Crear un programa que verifique e informe todos los puntos de silla de una matriz de dimensión $M \times N$ ingresada por el usuario. Un punto de silla es el máximo de su fila y el mínimo de su columna.
- 12) Crear un programa que verifique si una matriz de enteros de dimensión $M \times N$ ingresada por el usuario, es triangular superior. Una matriz triangular superior es aquella donde todos los elementos situados bajo la diagonal principal son cero.
- 13) Crear un programa que reciba una matriz de números enteros de $M \times N$ ingresada por el usuario, que verifique e informe cual es el mayor valor absoluto, si la suma de los elementos situados en la parte superior de la diagonal principal o la suma de los elementos situados en la parte inferior de la diagonal principal.
- 14) Crear un programa que reciba una matriz de números reales de $M \times N$ ingresada por el usuario, que sume los elementos de la diagonal principal y muestre el resultado por consola.
- 15) Crear un programa que lea el orden de un arreglo bidimensional y sus elementos. Luego convierta la matriz en un arreglo unidimensional. La conversión debe hacerse por columnas.
- 16) Utilizar un arreglo unidimensional para resolver el siguiente problema: una compañía paga a sus vendedores por comisión. Los vendedores reciben \$200 por semana más el 9% de sus ventas totales de esa semana. Por ejemplo, un vendedor que acumule \$5000 en ventas en una semana, recibirá \$200 más el 9% de \$5000, o un total de \$650. Escriba una aplicación (utilizando un arreglo de contadores) que determine cuántos vendedores recibieron salarios en cada uno de los siguientes rangos (suponga que el salario de cada vendedor se trunca a una cantidad entera):
 - a) \$200-299
 - b) \$300-399
 - c) \$400-499
 - d) \$500-599
 - e) \$600-699
 - f) \$700-799
 - g) \$800-899
 - h) \$900-999
 - i) \$ 1000 en adelante

Sintetice los resultados en formato tabular.

- 17) Realizar un programa que pida el ingreso de nombre y nota de 20 alumnos, almacene estos en una matriz y luego muestre un menú con las siguientes opciones:
- 1- Buscar un alumno
 - 2- Modificar su nota
 - 3- Realizar el promedio de todas las notas
 - 4- Mostrar al alumno con mejores notas
 - 5- Mostrar al alumno con peores notas
 - 6- Salir
- 18) Realizar un programa en donde se le pida al usuario ingresar las dimensiones de un arreglo tridimensional de $M \times N \times P$, luego se le pida ingresar los valores para todas las posiciones del arreglo y finalmente mostrar el valor mayor del arreglo indicando la posición del arreglo que ocupa.
- 19) Se dispone de 8 materias y una X cantidad de alumnos. Se desea realizar un programa en donde se pueda ingresar el nombre de las materias, los nombres de los alumnos y por cada alumno, si aprobó o desaprobó la materia. Para dicho programa se debe utilizar una matriz tridimensional.
Calcular e informar los alumnos que adeudan 1, 2, 3, 4, 5, 6, 7 y 8 materias.
- Los nombres de las materias se deben ingresar una sola vez, así también como los nombres de los alumnos, luego el programa deberá pedir por materia y por alumno si éste aprobó o no la materia correspondiente.
- 20) Una cadena de supermercados tiene los datos de sus empleados por número de sucursal. Los datos son DNI, apellido y cargo. Crear un programa que muestre los datos de los empleados de todas las sucursales ordenados alfabéticamente por apellido.

III. Nivel medio

- 1) Construya un algoritmo que realice las siguientes operaciones. Tiene 4 vectores A, B, C y D de 10 elementos numéricos cada uno, llene el vector A con números pares, el vector B con números impares, el vector C con la suma de $A + B$, el vector D con la resta de $A - B$. Determinar la suma de cada vector y utilizarlo para llenar el vector R que 4 elementos numéricos, es decir la suma del vector A corresponde a la posición 0 del Vector R, la suma del vector B corresponde a la posición 1 del vector R, y así respectivamente con C y D. Imprima el vector R.

2) Se dispone de la siguiente matriz:

```
String[][] figuras = { {"P", "P", "P", "P", "P", "P", "P", "P",
                        "P", "P", "P", "P", "P", "P", "P", "P"},
                      {"T", "T", "T", "T"},
                      {"C", "C", "C", "C"},
                      {"A", "A", "A", "A"},
                      {"R", "R"},
                      {"r", "r"}};
```

La misma representa a las figuras de un juego de ajedrez, siendo "P" un peón, "T" una torre, "C" un caballo, "A" un alfil, "R" un rey y "r" una reina.

Realice un programa que copie estas figuras en una matriz de 8x8 que representa al tablero de ajedrez, y dibuje con caracteres el mismo por consola.

El tablero de ajedrez tiene una disposición como la que se muestra a continuación



3) Se debe crear un programa para administrar el stock de productos de una tienda de comestibles la cual tiene la capacidad de administrar 50 variedades de los mismos. La información a ingresar por cada producto será: nombre, precio y cantidad.

Se debe mostrar un menú con las siguientes opciones:

- Consultar productos en la tienda
- Dar de alta una variedad de producto
- Dar de baja una variedad de producto
- Modificar stock de un producto
- Modificar precio de un producto
- Vender producto
- Salir

- 4) Se tiene la producción total de toneladas de cereales (arroz, avena, cebada, trigo) cosechadas durante cada mes del año anterior. Elaborar un programa que proporcione la siguiente información:
- El promedio anual de toneladas cosechadas.
 - ¿Cuántos meses tuvieron una cosecha superior al promedio anual?
 - ¿Cuántos meses tuvieron una cosecha inferior al promedio anual?
 - ¿Cuál fue el mes en el que se produjeron mayor número de toneladas?
- 5) Se tienen las temperaturas promedio diarias del año anterior de la ciudad de Buenos Aires, almacenadas en una tabla TEMP de 12 filas x 31 columnas. Elabore un programa que proporcione la siguiente información:
- La temperatura más alta registrada en el año anterior, y cuál fue el día y mes en que se registró.
 - El mes que tuvo el promedio de temperaturas más alto.
 - El promedio mensual de temperaturas
- 6) Se desea realizar una encuesta a 10 personas, en esta encuesta se debe indicar el sexo (masculino, femenino), si trabaja y su sueldo (si tiene un trabajo, sino será un cero) que estará entre 6000 y 20000 (valor entero).
Hacer un programa mediante el cual se ingresen estos datos y que calcule e informe:
- Porcentaje de hombres (tengan o no trabajo).
 - Porcentaje de mujeres (tengan o no trabajo).
 - Porcentaje de hombres que trabajan.
 - Porcentaje de mujeres que trabajan.
 - El sueldo promedio de los hombres que trabajan.
 - EL sueldo promedio de las mujeres que trabajan.
- 7) En una universidad se efectúa un examen de admisión que consta de dos pruebas: aptitud matemática y aptitud verbal. Cada pregunta tiene 5 opciones numeradas del 1 al 5. Se prepara un registro de 60 campos de una sola posición que contiene, cada uno, la respuesta correcta a la pregunta correspondiente. Las 30 primeras posiciones corresponden al examen de aptitud matemática y las restantes a las de aptitud verbal.
Se presentaron al examen N estudiantes y para cada uno de ellos se preparó un registro con los siguientes datos:
- Número de DNI
 - Respuestas al examen de aptitud matemática
 - Respuestas al examen de aptitud verbal

Se requiere saber:

- Puntaje obtenido por cada estudiante en cada examen
 - Puntaje total por estudiante
 - Puntaje promedio de cada examen
 - DNI y puntaje correspondiente a los alumnos que obtuvieron un puntaje superior o igual al promedio
 - El mayor puntaje y el número de DNI del alumno que lo obtuvo
- 8) Una fábrica produce N artículos diferentes y utiliza M sucursales para su distribución, variando el precio de venta de cada artículo según la sucursal que lo distribuye. Para esto la fábrica tiene un cuadro que muestra el precio de cada artículo según la sucursal que lo distribuye. Al final de cada periodo cada sucursal envía a la fábrica la cantidad vendida de cada artículo, formándose un nuevo cuadro. Elaborar un programa que encuentre las ventas totales de la fábrica para cada uno de los artículos y para cada una de las sucursales.
- 9) Considere un minitablero de ajedrez de 4x4 y realice un algoritmo para situar cuatro reinas, de tal modo, que no se coman entre sí. Recuerde que una reina come sobre su fila, su columna y las dos diagonales que pasan sobre su posición.
- 10) Un avión dispone de 180 plazas, de las cuales 60 son de no fumador y numeradas del 1 al 60 y 120 plazas numeradas del 61 al 180. Diseñar un algoritmo que permita reservar plazas del avión y diga cuantos pasajeros de cada clase ocupan asientos en el vuelo.
- 11) En las elecciones para Gobernador de la Ciudad de Buenos Aires, se han presentado tres candidatos (A, B, C). La ciudad está dividida en 5 zonas de votación.

El reporte de votos de la zona se escribe en orden: primero la zona 1, la 2, la 3, la 4, y por último la 5.

Elabore un algoritmo que:

- Forme una matriz de 5 filas y 3 columnas que contenga, en cada fila, los votos reportados por las zonas para cada uno de los tres candidatos.
- Encuentre el total de votos obtenidos por un candidato y el porcentaje que éste representa
- Escriba un mensaje declarando ganador a un candidato, si éste obtuvo más del 50% de la votación.

- 12) Realizar un algoritmo que realice la multiplicación de dos arreglos tridimensionales ingresados por el usuario e informe el resultado por consola.
- 13) Un arreglo tridimensional contiene la cantidad de hombres y mujeres que hay en 6 cursos de cada una de las 10 facultades de una universidad. Donde $alumnos[I,J,K]$ representa el número de estudiantes del curso I, de sexo J, de la facultad K. Encontrar:
- Cantidad de hombres en cada facultad
 - Cantidad de mujeres por facultad
 - La facultad con mayor número de mujeres
- 14) Una pequeña aerolínea acaba de comprar una computadora para su nuevo sistema de reservaciones automatizado. Se le ha pedido a usted que desarrolle el nuevo sistema. Usted escribirá una aplicación para asignar asientos en cada vuelo del único avión de la aerolínea (capacidad: 10 asientos). Su aplicación debe mostrar las siguientes alternativas: Por favor escriba 1 para Primera Clase y Por favor escriba 2 para Económico. Si el usuario escribe 1, su aplicación debe asignarle un asiento en la sección de primera clase (asientos 1 a 5). Si el usuario escribe 2, su aplicación debe asignarle un asiento en la sección económica (asientos 6 a 10). Su aplicación deberá entonces imprimir un pase de abordar, indicando el número de asiento de la persona y si se encuentra en la sección de primera clase o clase económica. Use un arreglo unidimensional del tipo primitivo boolean para representar la tabla de asientos del avión. Inicialice todos los elementos del arreglo con false para indicar que todos los asientos están vacíos. A medida que se asigne cada asiento, establezca el elemento correspondiente del arreglo en true para indicar que ese asiento ya no está disponible. Su aplicación nunca deberá asignar un asiento que ya haya sido asignado. Cuando esté llena la sección económica, su programa deberá preguntara la persona si acepta ser colocada en la sección de primera clase (y viceversa). Si la persona acepta, haga la asignación de asiento apropiada. Sino, imprima el mensaje "El próximo vuelo sale en 3 horas".
- 15) Use un arreglo bidimensional para resolver el siguiente problema: una compañía tiene cuatro vendedores (1 a 4) que venden cinco productos distintos (1 a 5). Una vez al día, cada vendedor pasa una nota por cada tipo de producto vendido. Cada nota contiene lo siguiente:
- a) El número del vendedor
 - b) El número del producto
 - c) El valor total en dólares de ese producto vendido en ese día

Así, cada vendedor pasa entre 0 y 5 notas de venta por día. Suponga que está disponible la información sobre todas las notas del mes pasado. Escriba una aplicación que lea toda esta información para las ventas del último mes y que resuma las ventas totales por vendedor, y por producto. Todos los totales deben guardarse en el arreglo bidimensional `ventas`.

Después de procesar toda la información del mes pasado, muestre los resultados en formato tabular, en donde cada columna represente a un vendedor específico y cada fila simboliza un producto. Saque el total de cada fila para obtener las ventas totales de cada producto durante el último mes. Calcule el total de cada columna para sacar las ventas totales de cada vendedor durante el último mes. Su impresión tabular debe incluir estos totales cruzados a la derecha de las filas totalizadas, y en la parte inferior de las columnas totalizadas.

- 16) El lenguaje Logo hizo famoso el concepto de los gráficos de tortuga. Imagine a una tortuga mecánica que camina por todo el cuarto, bajo el control de una aplicación en Java. La tortuga sostiene una pluma en una de dos posiciones, arriba o abajo. Mientras la pluma está abajo, el animalito va trazando figuras a medida que se va moviendo, y mientras la pluma está arriba, se mueve alrededor libremente, sin trazar nada. En este problema usted simulará la operación de la tortuga y creará un bloc de dibujo computarizado.

Utilice un arreglo de 20 por 20 llamado `piso`, que se inicialice con ceros. Lea los comandos de un arreglo que los contenga. Lleve el registro de la posición actual de la tortuga en todo momento, y si la pluma se encuentra arriba o abajo.

Suponga que la tortuga siempre empiece en la posición (0, 0) del piso, con su pluma hacia arriba. El conjunto de comandos de la tortuga que su aplicación debe procesar se muestra a continuación:

Comando	Significado
1	Pluma arriba
2	Pluma abajo
3	Voltear a la derecha
4	Voltear a la izquierda
5,10	Avanzar hacia delante 10 espacios (reemplace 10 por el nro de espacios)
6	Mostrar en pantalla el arreglo de 20 por 20
9	Fin de datos

Suponga que la tortuga se encuentra en algún lado cerca del centro del piso. El siguiente “programa” dibuja e imprime un cuadrado de 12 por 12, dejando la pluma en posición levantada:

2

5.12

3

5.12

3

5.12

3

5.12

1

6

9

A medida que la tortuga se vaya desplazando con la pluma hacia abajo, asigne 1 a los elementos apropiados del arreglo piso. Cuando se dé el comando 6 (mostrar el arreglo en pantalla), siempre que haya un 1 en el arreglo, muestre un asterisco o cualquier carácter que usted elija. Siempre que haya un 0, muestre un carácter en blanco.

Escriba una aplicación para implementar las herramientas de gráficos de tortuga aquí descritas. Escriba varios programas de gráficos de tortuga para dibujar figuras interesantes. Agregue otros comandos para incrementar el poder de su lenguaje de gráficos de tortuga.

- 17) Un número primo es cualquier entero mayor que 1, divisible sólo por sí mismo y por el número 1. La Criba de Eratóstenes es un método para encontrar números primos, el cual opera de la siguiente manera:

- a) Cree un arreglo del tipo primitivo boolean, con todos los elementos inicializados en true. Los elementos del arreglo con índices primos permanecerán como true. Cualquier otro elemento del arreglo con el tiempo cambiará a false.
- b) Empiece con el índice 2 del arreglo y determine si un elemento dado es true. De ser así, itere a través del resto del arreglo y asigne false a todo elemento cuyo índice sea múltiplo del índice del elemento que tiene el valor true. Después continúe el proceso con el siguiente elemento que tenga el valor true. Para el índice 2 del arreglo, todos los elementos más allá del elemento 2 en el arreglo que tengan índices múltiplos de 2 (los índices 4, 6, 8, 10, etcétera) se establecerán en false; para el índice 3 del arreglo, todos los elementos más allá del elemento 3 en el arreglo que tengan índices múltiplos de 3 (los índices 6, 9, 12, 15, etcétera) se establecerán en false; y así en lo sucesivo.

Cuando este proceso termine, los elementos del arreglo que aún sean true indicarán que el índice es un número primo. Estos índices pueden mostrarse. Escriba una aplicación que utilice un arreglo de 1000 elementos para determinar e imprimir los números primos entre 2 y 999. Ignore los elementos 0 y 1 del arreglo.

- 18) Crear un programa en donde se debe pedir al usuario el ingreso de datos numéricos para un arreglo de 4 dimensiones de $n \times n \times n \times n$, en donde $n = 4$; Ordenar el arreglo en forma ascendente y mostrar el resultado por consola.
- 19) Una empresa guarda en una tabla de $3 \times 12 \times 4$ las ventas realizadas por sus tres representantes a lo largo de doce meses de sus cuatro productos, `VENTAS[representante, mes, producto]`. Se desea proyectar el array tridimensional sobre uno de dos dimensiones que represente el total de ventas, `TOTAL[mes, producto]`, para lo cual se debe sumar las ventas de cada producto de cada mes de todos los representantes. Imprimir ambos arreglos.
- 20) Elaborar un algoritmo que actualice la tabla de clasificación del Campeonato Profesional de Fútbol de su país; Luego de efectuada una fecha, el algoritmo debe leer:
- Un registro por equipo que contenga los siguientes datos:
 - Código de equipo
 - Partidos jugados
 - Partidos ganados
 - Partidos empatados
 - Partidos perdidos
 - Goles a favor
 - Goles en contra
 - Puntos
 - Un registro por partido que contenga los siguientes datos:
 - Código del equipo local
 - Goles del equipo local
 - Código del equipo visitante
 - Goles del equipo visitante

El algoritmo debe mostrar la tabla leída, ordenada de acuerdo a la puntuación, luego de haber calculado los puntos para cada equipo.

IV. Nivel avanzado

- 1) Se tiene un reloj digital con LEDs de 7 segmentos. Se desea construir un programa que informe cuántos segmentos se han encendido DESPUÉS DE X segundos, desde la posición 00:00:00.
Considerar que en cada segundo, todos los led se apagan y luego se encienden los correspondientes al instante actual.
- 2) Crear un programa capaz de resolver un problema que consiste en colocar N reinas de ajedrez en un tablero de NxN de tal forma que ninguna reina esté amenazada (esto es, no puede haber dos reinas en la misma fila, ni en la misma columna, ni en la misma diagonal). El programa recibe como parámetro el número N de reinas que se desea colocar en el tablero.

La siguiente tabla muestra el número de veces que se coloca una reina en el tablero hasta que se encuentra una solución válida al problema.

Reinas	Intentos
2	Sin solución
3	Sin solución
4	9
5	6
6	32
7	10
8	114
9	42
10	103
15	1360
20	199636
25	48684

- 3) Se desea crear un programa que muestre un vector por consola de manera que si los valores de un vector son: 1,2,3 en consola muestre:

1
1,2
1,3
2
2,3
3
1,2,3

Para el caso que el vector tenga los valores 1,2,3,4

1
1,2
1,3
1,4
2
2,3
2,4
3
3,4
4
1,2,3,4

Realizar el programa de modo que la longitud del vector la ingrese el usuario, los valores que contenga el mismo serán de 1 a longitud, y finalmente lo muestre como se ha explicado.

c. Excepciones

- Agregue a los ejercicios anteriores los bloques try catch correspondientes para que, si el programa detecta un error lo informe y lo solucione sin terminar la ejecución del programa.

4. Métodos

- Para todos los ejercicios debe utilizar métodos

I. Nivel Inicial

- 1) Crear un programa que pida al usuario su nombre y mediante un método muestre un saludo por consola que realice un saludo hacia él.
- 2) Crear un programa que mediante un método incremente un número en 1 unidad.

- 3) Crear un programa que mediante un método calcule la potencia de un número. No se debe utilizar el método pow.
- 4) Crear un programa que, mediante un método, calcule el resultado de restar el doble de un número a su cuadrado.
- 5) Crear un programa que reciba un número del 1 al 12 desde el teclado y muestre el número de días correspondiente al mes que corresponda con ese día.
- 6) Crear un programa que realice el promedio de tres números mediante un método.
- 7) Crear un programa que mediante un método calcule la suma de los números del 1 al N.
- 8) Crear un programa en donde el usuario deba ingresar una serie de números y el mismo calcule e informe la cantidad de números positivos.
- 9) Crear un programa en el que se pida una nota de parcial, y mediante este informe si desaprueba (entre 1 y 3), aprueba (entre 4 y 6), o promociona (entre 7 y 10). Repetir el programa hasta que el usuario ingrese como nota el número 99.
- 10) Crear un programa que informe la última cifra correspondiente a un número entero.
- 11) Crear un programa con un método que devuelva el máximo de tres números ingresados.
- 12) Crear un programa que contenga un método que devuelva el área de un círculo dado su radio. Recordar que para calcular el área, la fórmula es $\pi * r^2$.
- 13) Crear un programa que permita convertir una cantidad de dinero dada en dólares, libras o yenes a Euros. Estas son las equivalencias aproximadas:

1 libra=1,22 euros
1 dolar=0,75 euros
1 yen=0.009 euros
- 14) Crear un programa que dibuje una pirámide invertida en pantalla como la de la figura. La altura se pasará como parámetro a un método. Si se pasa una altura = 0 o negativa, el método devolverá -1; en caso contrario devolverá 0.

Ejemplo para altura = 5

*

- 15) Escribir un programa que mediante un método que recibe por parámetros los catetos de un triángulo rectángulo, calcule la hipotenusa.
- 16) Pedir tres valores ambientales de temperatura y calcular e informar la suma y el promedio de dichos valores. Para los cálculos utilice métodos.
- 17) Pedir que se ingrese tres valores numéricos e informar si están ordenados de manera creciente.
- 18) Se pide ingresar la altura en cm de todos los alumnos de una división. Calcular e informar cual es la altura promedio y cuantos, "en porcentaje", son mayores al promedio.
- 19) Pedir el ingreso de x cantidad de números enteros y determinar mediante una función, sin utilizar arreglos, el mayor y el menor valor. Indicar también si algunos de estos valores está repetido, en cuyo caso informar la cantidad de repeticiones.
- 20) Confeccionar un programa que ingrese un valor expresado en terabyte y lo informe expresado en: Tb, Gb, Mb, Kb y Byte, con leyendas aclaratorias. (1k = 1024 bytes).

II. Nivel básico

- 1) Escribir un programa que mediante métodos obtenga la distancia entre dos puntos que se encuentran en el plano a partir de la lectura de sus coordenadas.
- 2) Una farmacia vende algunos artículos sin descuento y otros con descuento del 20%. Confeccionar un programa que recibiendo el precio original y un código que indica si es o no con descuento, informe el precio final.
- 3) La farmacia Sindical efectúa descuentos a sus afiliados según el importe de la compra con la siguiente escala:

- a) menor de \$55 el descuento es del 4,5%
- b) entre 55 y 100\$ el descuento es del 8,0%
- c) más de 100\$ el descuento es del 10,5%

Confeccionar un programa que reciba un importe e informe: el precio ingresado, el descuento y el precio neto a cobrar, con mensajes aclaratorios.

- 4) Confeccionar un programa que pueda determinar el importe a pagar por una prestación médica según su código. Los importes se cobran según la siguiente tabla:

Código	Importe \$
A	20
D	40
F	60
M	150
T	150

Por cada prestación se debe ingresar el número de la historia del paciente y el código de la prestación. Se debe emitir un ticket para cada uno, con el número de la historia, el código y el importe a pagar.

- 5) Crear un programa en el cual se deben ingresar 18 valores de temperaturas distintos de cero. Se pide determinar e informar cuantas ternas (tres valores seguidos) de valores positivos y cuantas de negativos hay.
- 6) Se ha realizado la medición de la temperatura durante todos los días de 1 año, hora por hora, desde la hora 1 hasta la 24. Suponemos todos los meses de 30 días. Cada temperatura se midió y se registró en una planilla, en secuencia, o sea ordenada por mes, día y hora.
- Confeccionar un programa para: determinar e informar:

- a) La temperatura promedio de cada mes
- b) El mes, día y hora en que se produjo la máxima temperatura
- c) Cual ha sido el mes en que se registró el menor promedio de temperaturas

- 7) Como resultado de un experimento se obtuvieron 18 valores de temperatura, distintos de cero. Se debe realizar un programa que ingrese dichos valores y determine e informe:

- a) el mayor valor ingresado y cuál fue su número de orden durante el ingreso.
- b) ídem del menor valor.
- c) el promedio de los valores negativos ingresados.

- 8) Confeccionar un programa que permita el ingreso de 6 ternas de valores enteros.
- a) de cada terna informar el mayor valor y su orden en la terna.
 - b) de cada terna informar cuantos valores son múltiplos de 7. Ídem de todas las ternas.
 - c) de las 6 ternas determinar e informar el promedio de los primeros valores de cada una.
- 9) Confeccionar un programa que pueda calcular la edad de varias personas en base a la fecha actual y a la fecha de nacimiento de las mismas. Ingresar la fecha actual (día, mes, año). Ingresar a continuación, de a uno por vez, el nombre y la fecha de nacimiento de los integrantes del grupo. Para cada uno de ellos informar el nombre y la edad con leyendas indicativas.
- 10) Confeccionar un programa que ingrese las coordenadas "x" e "y" de 10 puntos del plano, valores a punto flotante y distintos de cero. Determine e informe
- a) para cada punto ingresado, las coordenadas y el cuadrante donde se halla ubicado.
 - b) en cual o cuales de los cuadrantes no se encontró ningún punto.
 - c) en que cuadrante se ubica el punto que tiene la máxima distancia al origen de coordenadas.
- 11) Confeccionar un programa que permita el ingreso de hasta 30 ternas de valores enteros y positivos. Si se quiere finalizar el ingreso antes de los 30 se informa una terna con los 3 valores iguales. Se pide para cada terna:
- a) determinar e informar el menor valor. Confeccionar y usar la función "menor".
 - b) determinar e informar los valores pares de cada terna. Confeccionar y usar la función "par".
 - c) determinar e informar el factorial del menor valor. Confeccionar y usar la función "factorial".
 - d) determinar e informar el promedio de las ternas ingresadas de orden impar. Usar la función "promedio".
- 12) Crear un programa en el cual se deben ingresar 2 valores n y m ($n \geq m$). Calcular e imprimir el valor del número combinatorio:

$$\binom{n}{m} = \frac{n!}{m! (n-m)!}$$

- 13) Confeccionar un programa que permita ingresar varias ternas de valores a punto flotante, finalizando al ingresar una terna con los tres valores ORDENADOS en forma creciente. Para cada terna exhibir la siguiente pantalla:

<p>** OPCIONES POSIBLES **</p> <p>1 – Mayor valor</p> <p>2 – Promedio</p> <p>3 – Suma</p> <p>4 - Finalizar</p> <p>** Digite su opción :</p>

Según la opción deseada debe aparecer en pantalla el resultado, sin borrar el menú. No admitir otros valores de opción. Confeccionar y utilizar un método PARA CADA UNA de las opciones y otro para detectar la condición de final.

- 14) Hacer un programa que muestre por pantalla la representación en binario de distintos tipos de datos: short, int, long y char. El programa pedirá un valor del tipo en cuestión y mostrará la representación de dicho valor en binario. Para ello se deben usar los operadores a nivel de bits, accediendo a cada bit individualmente y mostrando su valor (0 o 1).

- 15) Escribir un programa que realice la descomposición en factores primos de un número introducido por teclado. El programa deberá ir escribiendo la tabla de los factores primos, a medida que los va calculando, tal como muestra el ejemplo siguiente:

Introduce un Nº entero -> 84

Nº | Factores primos

--- | -----

84 | 2

42 | 2

21 | 3

7 | 7

1 |

- 16) Escribir un método que tome un valor entero y devuelva el número con sus dígitos en reversa. Por ejemplo, dado el número 7631 la función deberá devolver 1367.
- 17) Crear un programa que calcule el producto escalar y vectorial de dos vectores de 3 elementos cuyos valores se introducen por consola.
- 18) Julio Cesar enviaba mensajes a sus legiones encriptando los mensajes mediante el siguiente algoritmo:

Se escogía un número n como clave y se sumaba a cada letra en el alfabeto n posiciones. Así, la clave escogida fuese 5, la 'a' pasaría a ser la 'f', mientras que la 'f' pasaría a ser la 'k'. Con las últimas letras del abecedario se seguiría desde el principio. Así, con la clave 5 la 'y' pasaría a ser la 'd'.

Se pide crear un programa que encripte mensajes mediante este algoritmo.

19) Escribir una función en el que dados los numeradores y los denominadores de dos fracciones devuelva la suma de estos.

20) Crear un programa que muestre el siguiente menú:

CALCULO DE AREAS

=====

- 1 - Calcular el área de un triángulo
- 2 - Calcular el área de un trapecio
- 3 - Calcular el área de un rectángulo

Y lea la opción introducida por el usuario. En función de la opción el programa deberá pedir los datos necesarios para calcular el área, realizar el cálculo y mostrar el resultado por pantalla.

Utilice un método para mostrar el menú, y otros tres para calcular el área de los distintos polígonos (triángulo, trapecio rectángulo). El valor de retorno de cada uno de estos tres últimos métodos debe ser el área calculada.

III. Nivel medio

1) Leer de una primer planilla los valores X_C e Y_C que corresponden a las Coordenadas del centro de una circunferencia y R_1 , R_2 , R_3 que corresponden a los radios de 3 circunferencias concéntricas con centro en X_C , Y_C . Confeccionar y utilizar un método, llamado lectura, que ingrese los datos y controle que $R_1 < R_2 < R_3$. No ingresar datos erróneos.

Ingresa a continuación los valores de las coordenadas de diversos puntos X_P e Y_P , que se informan a un par por planilla, en la última hay dos valores nulos.

Se pide determinar e informar:

- a) cantidad de puntos que están dentro de la circunferencia R_1 .
- b) cantidad de puntos dentro de la corona $R_1 - R_2$ (inclusive). Hacer y usar un método.
- c) cantidad de puntos dentro de la corona $R_2 - R_3$ (inclusive).
- d) cantidad de puntos fuera la circunferencia R_3 . Hacer y usar un método.

$$d = \sqrt{(x_c - x_p)^2 + (y_c - y_p)^2}$$

- 2) Se dice que un número entero es un número perfecto si sus factores, incluyendo 1 (pero no el número entero), al sumarse dan como resultado el número entero. Por ejemplo, 6 es un número perfecto ya que $6 = 1 + 2 + 3$. Escriba un método llamado `esPerfecto` que determine si el parámetro `numero` es un número perfecto. Use este método en una aplicación que muestre todos los números perfectos entre 1 y 1,000. Imprima los factores de cada número perfecto para confirmar que el número sea realmente perfecto. Ponga a prueba el poder de su computadora, evalúe números más grandes que 1,000. Muestre los resultados.
- 3) El máximo común divisor (MCD) de dos enteros es el entero más grande que puede dividir de manera uniforme a cada uno de los dos números. Escriba un método llamado `mcd` que devuelva el máximo común divisor de dos enteros. [Sugerencia: tal vez sea conveniente que utilice el algoritmo de Euclides. Puede encontrar información acerca de este algoritmo en http://es.wikipedia.org/wiki/Algoritmo_de_Euclides.] Incorpore el método en una aplicación que reciba como entrada dos valores del usuario y muestre el resultado.
- 4) Escriba una aplicación que juegue a “adivinar el número” de la siguiente manera: su programa pide al usuario el número a adivinar, el mismo debe ser un entero en el rango de 1 a 1,000. La aplicación muestra el indicador Adivine un número entre 1 y 1,000. El jugador escribe su primer intento. Si la respuesta del jugador es incorrecta, su programa debe mostrar el mensaje Demasiado alto. Intente de nuevo, o Demasiado bajo. Intente de nuevo., para ayudar a que el jugador “se acerque” a la respuesta correcta. El programa debe pedir al usuario que escriba su siguiente intento. Cuando el usuario escriba la respuesta correcta, muestre el mensaje Felicidades. Adivino el número! y permita que el usuario elija si desea jugar otra vez.
- 5) Las resistencias electrónicas suelen ir identificadas por un código de colores que permite marcar cada resistencia con su valor (en Ohmios, Ω) y su Tolerancia (%). Este código de colores viene representado en la siguiente tabla:

Dígito	Color	Multiplicador	Tolerancia
	Ninguno		20%
	Plata	0.01	10%
	Oro	0.1	5%
0	Negro	1	
1	Marrón	10	
2	Rojo	10^2	2%
3	Naranja	10^3	
4	Amarillo	10^4	
5	Verde	10^5	
6	Azul	10^6	
7	Violeta	10^7	
8	Gris		
9	Blanco		

El código que suele emplearse en las resistencias es un código de 4 colores, es decir, cada resistencia está marcada con 4 bandas y cada una de ellas puede ser de diferente color. Cada banda tiene un significado, que depende de cada color:

- Las primeras 2 bandas indican un número de 2 dígitos: Esos dos dígitos vienen dados por el color de esas bandas, según la columna "Dígito" de la tabla.
- La tercera banda es un valor por el que se multiplicará el número obtenido por las bandas anteriores. Una vez multiplicados ambos valores, obtenemos el valor de la resistencia en Ohmios (Ω).
- La cuarta banda indica la tolerancia de la resistencia y, como puede verse en la tabla, no puede ser de cualquier color.

Ejemplo: Unas resistencias con los siguientes colores, tienen los siguientes valores de resistencia y tolerancia:

Verde-Azul-Amarillo-Oro	560k Ω , 5%
Rojo-Negro-Rojo-Rojo	2k Ω , 2%
Rojo-Rojo-Marrón-Plata	220 Ω , 10%

Según todo lo anterior, implemente un método que permita calcular la resistencia y la tolerancia de una resistencia, sabiendo los códigos de colores. El método tendrá, como mínimo, 4 argumentos, que serán números naturales, y que indicarán el color de las bandas según la columna "Dígito". Los colores Oro, Plata y Ninguno tomarán los valores 10, 11 y 12 respectivamente.

Implementar otro método que muestre por pantalla el dígito que le corresponde a cada color (incluyendo los dígitos 10, 11 y 12).

Implementar también un método que pida los colores de las 4 bandas y muestre los valores devueltos por el método anterior. El programa mostrará el dígito que le corresponde a cada color usando el procedimiento ya creado y leerá de teclado 4 números que corresponderán a los colores de las 4 bandas.

Tras esta lectura mostrará los datos de la resistencia con esos colores en las bandas. El programa se repetirá indefinidamente hasta que lea un valor negativo como color de una banda.

- 6) Crear un programa que realice operaciones sobre una matriz de números reales de tamaño DIMxDIM, donde DIM es una constante, declarada como tal, de valor 10. Escribir los siguientes métodos:

- a) Método `traspuestaM`: Acepta una matriz como único argumento y obtiene su matriz traspuesta.
- b) Método `simetricaM`: Acepta una matriz como único argumento y devuelve 1 si dicha matriz es simétrica y 0 si no lo es. Para averiguar si la matriz es o no simétrica se debe usar la función `traspuestaM` del apartado anterior.
- c) Método `sumaM`: Acepta tres matrices en donde la tercera de ellas la suma de las dos primeras.
- d) Método `restaM`: Acepta tres matrices en donde la tercera de ellas la resta de las dos primeras.
- e) Método `multiplicaM`: Acepta tres matrices en donde la tercera de ellas la multiplicación de las dos primeras.
- 7) Quizás el más famoso de todos los sistemas de codificación es el código Morse, desarrollado por Samuel Morse en 1832, para uso en el sistema telegráfico. El código Morse asigna una serie de puntos y rayas a cada letra del alfabeto, a cada dígito y a unos cuantos caracteres especiales. La separación entre palabras se indica por un espacio o por la ausencia de un punto o una raya. La versión internacional del código Morse aparece en la tabla siguiente:

Carácter	Código	Carácter	Código
A	.-	T	-
B	-...	U	..-
C	-.-.	V	...-
D	-..	W	.-..
E	.	X	-..-
F	..-.	Y	-.--
G	--.	Z	--..
H		
I	..	Números	
J	.-..	1	.----
K	-.-	2	..---
L	.-..	3	...--
M	--	4-
N	-.	5
O	---	6	-....
P	.-..	7	--...
Q	--.-	8	---..
R	.-.	9	----.
S	...	0	-----

Escriba un programa que lea una frase escrita en español y cifre dicha frase en código Morse y que también lea una frase en código Morse y la convierta en el equivalente en español. Utilice un espacio en blanco entre cada letra codificada Morse y tres espacios en blanco entre cada palabra codificada en Morse.

El programa deberá incorporar un método Menu() que muestre las siguientes opciones:

- 1) Pasar una frase a código Morse, lo cual se implementará en un método que se llame frase2Morse()
 - 2) Pasar código Morse a una frase, implementando un método que se llame morse2Frase()
 - 3) Salir.
-
- 8) Análisis de texto. La disponibilidad de computadoras con capacidades de manipulación de cadenas nos proporciona interesantes métodos para analizar lo escrito por grandes autores. Se ha puesto, por ejemplo, gran atención al hecho de saber si William Shakespeare alguna vez existió. Algunos estudiosos creen que existen evidencias indicando que Christopher Marlowe fue el que escribió las obras maestras atribuidas a Shakespeare. Los investigadores han utilizado ordenadores para localizar similitudes en los textos de estos dos autores. Realice un programa que lea varias líneas de texto y analice las siguientes características del texto:
- a) Imprimir una tabla indicando el número de veces que aparece cada letra del alfabeto en dicho texto.
 - b) Imprimir una tabla que indique el número de palabras de una letra, de dos letras, de tres letras... etc. que aparecen en el texto.
 - c) Imprimir una tabla indicando el número de ocurrencias de cada palabra distinta en el texto. Para ello supondremos que el texto tiene como máximo 100 palabras distintas, con lo que deberá almacenarlas en un array de estructuras de tamaño 100. Cada estructura deberá contener una cadena de caracteres con la palabra (máximo 20 caracteres) y otro campo con el número de veces que aparece esa palabra en el texto. Considere opcionalmente la posibilidad de que las palabras aparezcan ordenadas alfabéticamente. Para ordenarlas puede utilizar cualquier algoritmo de ordenación teniendo en cuenta que hay que intercambiar estructuras completas (la cadena y el número de ocurrencias).

El programa deberá mostrar un menú con las siguientes opciones:

- 1) Introducir texto
- 2) Número de instancias de cada letra
- 3) Número de palabras de cada longitud
- 4) Número de ocurrencias de cada palabra distinta
- 5) Salir

- 9) Escribir un programa que permita guardar las cuentas de un banco con sus respectivos saldos. Para ello se guardará la información en una matriz, que corresponderá a números de cuenta y saldos de cuenta.
El programa deberá mantener las cuentas ordenadas, de menor a mayor, por número de cuenta para facilitar la búsqueda en una cuenta.
El programa mostrará un menú con las siguientes opciones:
- 1- Dar de alta una nueva cuenta (comprobando que no esté lleno y colocando la cuenta en la posición correspondiente)
 - 2- Eliminar una cuenta (comprobando que no este vacío y reposicionando las cuentas)
 - 3- Mostrar una cuenta (mostrará el número de cuenta y el saldo correspondiente)
 - 4- Mostrar información (Número de cuentas dadas de alta y dinero total de todas ellas)
 - 5- Calcular el saldo medio, máximo y mínimo de las cuentas.
 - 6- Mostrar todas las cuentas (1 línea por cuenta con su número y su saldo)
 - 7- Salir
- 10) Crear un programa que permita reservar asientos de una sala de cine (8 filas x 20 columnas). La posición de cada asiento se definirá con una letra (A-H) para la fila y un número (1-20) para la columna. El programa deberá visualizar que sitios están disponibles para que el cliente pueda decidir donde sentarse. Antes de hacer la reserva, el programa deberá comprobar que el asiento está libre, en caso contrario devolverá un mensaje de error.

Ejemplo de visualización:

```

1...5...0 1...5...0
A  *      *      A
B *** **      ** B
C ** *  ****   C
D *  **  ****  D
E *****   E
F *****   F
G  ** ** *  *  G
H  ** ** *  *  H
1...5...0 1...5...0

```

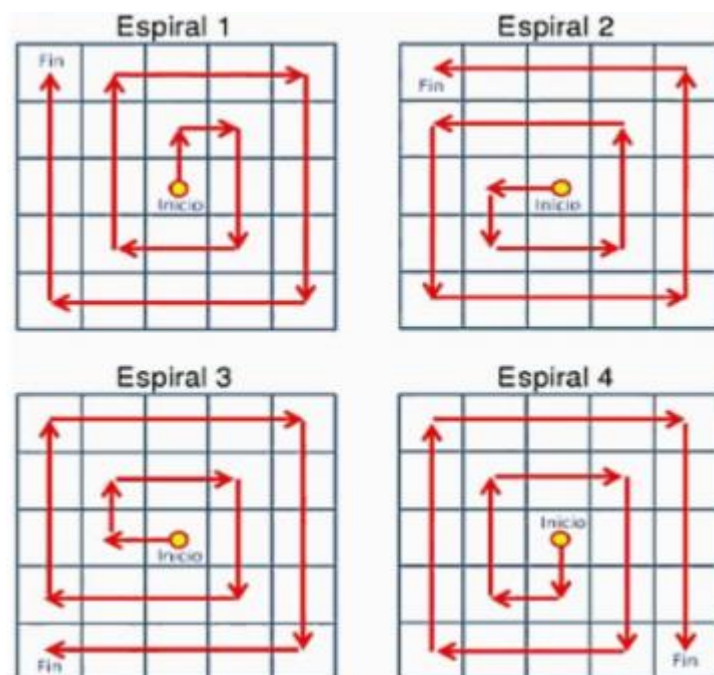
- 11) Implementar un programa que determine si un vector ingresado por teclado es simétrico o no, un vector es simétrico si el elemento de la i-ésima posición es igual al elemento de la posición n – i-ésima; por ejemplo: X = [1,2,3,4,3,2,1].
- 12) La fábrica de muebles “El hermano gorgojo” tiene en el proceso de construcción de sillas de oficina en 5 fases (A,B,C,D,E) que se pueden realizar en 6 máquinas diferentes (1..6), se necesita calcular el costo máximo y mínimo tanto para fases como para máquinas; en la tabla siguiente se muestra la relación costo-proceso-máquina que estos eventos conllevan:

	A	B	C	D	E
1	6	5	9	19	11
2	45	23	21	17	30
3	15	7	23	25	2
4	75	26	38	49	19
5	2	15	76	1	45
6	92	82	4	20	13

- 13) Con el uso de matrices cuadradas de orden “n” (n controla el tamaño de los caracteres) con ceros y unos, elabore el abecedario, los números del 0 al 9 y los caracteres especiales: signos de admiración, interrogación, puntuación, etc. Con el uso de estas estructuras implemente un programa que simule un tablero electrónico de publicidad, el cual lea una cadena de caracteres, el tamaño de los caracteres (n) y despliegue la información animada de derecha a izquierda de la pantalla. Por ejemplo, si se ingresa la cadena “HOLA” y como tamaño 5, en la pantalla se visualizará de forma animada una gráfica de forma similar a lo siguiente:

1				1	1	1	1	1	1	1					1	1	1	1	1
1				1	1				1	1					1				1
1	1	1	1	1	1				1	1					1	1	1	1	1
1				1	1				1	1					1				1
1				1	1	1	1	1	1	1	1	1	1	1	1				1

- 14) Construya una solución que genere en su respectivo recorrido en espiral las matrices cuadradas de orden “n” (n impar ≥ 5) mostradas en el siguiente figura:



- 15) Escribir un programa que lea dos números complejos y permita realizar con ellos las siguientes operaciones aritméticas: suma, resta, multiplicación y división.

La suma y resta de números complejos se obtiene sumando o restando las partes reales y las partes complejas.

El producto de dos números complejos se obtiene multiplicando sus módulos y sumando sus argumentos.

El cociente de dos números complejos se obtiene dividiendo sus módulos y restando sus argumentos.

- 16) Escribir un algoritmo que, dado el infinitivo de un verbo regular de la primera conjugación, obtenga la conjugación en singular y plural de presente de indicativo. Por ejemplo, para el verbo cantar el resultado es yo canto, tu cantas, el canta, nosotros cantamos, vosotros cantáis, ellos cantan.

- 17) El programa solicita el tipo de cálculo a realizar, combinaciones, permutaciones o variaciones, tanto con repetición como sin repetición. También solicita los parámetros necesarios para el cálculo. El programa aplica la fórmula pertinente y presenta el resultado.

Permutaciones de m elementos:	$m!$	Combinaciones de m elementos tomados de n en n :	$\frac{m!}{n!(m-n)!}$
Variaciones con repetición de m elementos tomados de n en n :	m^n		
Variaciones sin repetición de m elementos tomados de n en n :	$\frac{m!}{n!}$	Combinaciones con repetición de m elementos tomados de n en n :	$\frac{(m+n-1)!}{n!(m-1)!}$

Es interesante observar los límites prácticos para los cálculos producidos por el desbordamiento. Estúdiense distintas posibilidades para evitarlo

- 18) Realice un programa que pida al usuario una fórmula química en una cadena de caracteres. Esta fórmula podrá tener los siguientes caracteres:

- 'C', 'H', 'O', 'N' como identificadores de los elementos básicos de la química del carbono
- '2', '3', '4', '5' y '6' como índices posibles de estos elementos

Se pide analizar esta fórmula para:

- Decidir si contiene algún carácter no válido, en cuyo caso se deberá volver a pedir una nueva fórmula al usuario

- En el caso de que todos los caracteres sean válidos mostrar por pantalla el número de átomos de cada tipo que contiene la fórmula
- Por último, mostrar por pantalla el peso molecular

Se recomienda dividir el problema en subproblemas más sencillos para que sea más fácil la resolución del problema final. Notar que el número de átomos de un elemento es función del carácter situado a continuación de él en la fórmula

NOTA: Para que el programa considere que una fórmula es correcta es suficiente con comprobar que sólo contiene caracteres válidos. Es decir, no es necesario que el programa decida si la fórmula es químicamente correcta o no. Son ejemplos de fórmulas consideradas válidas: CH₃CH₃, H₃CCH₃, C₂H₆.

- 19) Crear un programa para jugar al ta te ti en el que participen dos jugadores.
- 20) El doctor Maligno se encuentra a salvo en su super laboratorio secreto a 500 metros bajo tierra. Para atraparlo, sólo tienes que acertar el código de seguridad de la puerta acorazada de la entrada. Este código cambia cada día, así que deberás darte prisa en acertarlo.

El código de seguridad consiste en una secuencia de cuatro colores. Los colores posibles con su letra asociada son:

rojo	R
azul	A
amarillo	M
verde	V
naranja	N
violeta	I

Al introducir la secuencia de cuatro colores, el sistema responderá indicando el número de colores correctos pero fuera de su orden correcto (una X por cada acierto) y el número de colores correctos y en su posición correcta (un * por cada acierto). En la salida, primero se listan todas las 'X' y luego todos los '*', sin importar el orden en el que se hayan detectado.

En el código de seguridad se pueden repetir colores.

Algunos ejemplos de entrada y salida:

código de seguridad	entrada	salida	Descripción
RANI	YNYI	X*	'X' indica el acierto en posición de 'I' y '*' indica el acierto del color 'N'
RANI	RMVI	XX	'XX' indica el acierto en posición de 'R' y 'I'
NRRI	RRVN	X**	'X' por acierto en posición de la primera 'R', '*' por acierto del color de la segunda 'R' y '*' por la 'N'

El juego te irá pidiendo un nuevo código hasta que aciertes el código de seguridad o hasta que el malvado científico te descubra y... se ocupe de ti.

¿Quieres complicar el juego? Para ello puedes añadirle más posibles colores a la combinación o un límite máximo de intentos.

Resumen de las normas

- Al comenzar el juego, se genera una clave de colores, que puede ser pedida a otro usuario diferente al que jugará.
- En cada intento hay que introducir 4 letras correspondientes a los colores posibles.
- Después de cada intento, el juego informará de los aciertos encontrados (según las reglas explicadas).
- El juego termina cuando el código de seguridad ha sido acertado.
- El límite de intentos lo definirá el programador

IV. Nivel avanzado

- 1) Crear un programa capaz de calcular la suma de los números que se indicarán en la entrada estándar, separados por espacios, y mostrar los resultados en consola. Los números pueden ser negativos, grandes y las líneas pueden contener espacios adicionales, por lo que el programa debe ser robusto.

Ejemplo de entrada

```
123456789012 1
2 3
4 5 -1
```

Ejemplo de salida

```
123456789013
5
8
```

- 2) Dada una serie de palabras separadas por espacios, escribir la frase formada por las mismas palabras en orden inverso. Cada palabra estará formada

exclusivamente por letras, y existirá exactamente un espacio entre cada pareja de palabras. La salida debe ser "Caso #" seguido del número de caso, de un símbolo de "dos puntos", de un espacio en blanco y de la frase invertida.

El primer dato de entrada será la cantidad de valores que se van a analizar.

Ejemplo de entrada

```
3
esto es una prueba
programación
si llegue hasta acá es porque voy a aprobar
```

Salida correspondiente

```
Caso #1: prueba una es esto
Caso #2: programación
Caso #3: aprobar a voy porque es acá hasta llegue si
```

- 3) Eres un biólogo que examina secuencias de ADN de formas de vida diferentes. Se te darán dos secuencias de ADN, y el objetivo es encontrar el conjunto ordenado de bases adyacentes de mayor tamaño que es común en ambos ADNs.

Las secuencias de ADN se darán como conjuntos ordenados de bases de nucleótidos: adenina (abreviado A), citosina (C), guanina (G) y timina (T):

```
ATGTCTTCCTCGA TGCTTCCTATGAC
```

Para el ejemplo anterior, el resultado es CTCCT porque que es el conjunto ordenado de bases adyacentes de mayor tamaño que se encuentra en ambas formas de vida.

Ejemplo de entrada

```
ctgactga actgagc
cgtaattgcat cgtacagtagc
ctgggccttgaggaaaactg gtaccagtactgatagt
```

Salida de la muestra

```
actga
cgta
actg
```

- 4) Se está estudiando un enjambre de N luciérnagas. Cada luciérnaga se mueve en línea recta a velocidad constante. Usted se encuentra en el centro del universo, en la posición $(0, 0, 0)$. Todas las luciérnagas tienen la misma masa, y se desea saber a qué distancia mínima desde su ubicación (el origen de coordenadas) se llega a encontrar el centro del enjambre.

Se conoce la posición y la velocidad de cada luciérnaga en $t = 0$, y pero sólo se está interesados en instantes $t \geq 0$. Las luciérnagas tienen velocidad constante, y pueden moverse libremente a través de todo el espacio. Siendo $M(t)$ la ubicación del centro de masa de las luciérnagas N en el momento t , $d(t)$ la distancia entre su posición y $M(t)$ en el instante t . Debe encontrar el valor mínimo de $d(t)$, D_{\min}

Entrada

La primera línea de entrada contiene un único entero T , el número de casos de prueba. Cada caso de prueba se inicia con una línea que contiene un entero N , la cantidad de luciérnagas, seguido por N líneas de la forma

$x \ y \ z \ v_x \ v_y \ v_z$

Cada una de estas líneas se describe una luciérnaga: (x, y, z) es su posición inicial en el instante $t = 0$, y (v_x, v_y, v_z) es su velocidad.

Salida

Para cada caso de prueba, la salida será

Caso #X: $d_{\min} \ t_{\min}$

donde X es el número de caso de prueba, a partir de 1. Cualquier respuesta con un error absoluto o relativo de un máximo de 10^{-5} , serán aceptable.

Límites

Todos los números en la entrada serán enteros.

$1 \leq t \leq 100$

Los valores de x, y, z, v_x, v_y y v_z será entre -5000 y 5000 , ambos inclusive.

Conjunto de datos pequeño

$3 \leq n \leq 10$

Conjunto de datos grande

$3 \leq n \leq 500$

Entrada de ejemplo

3 -4 0 0 0 3

-3 -2 -1 3 0 0

-3 -1 2 0 3 0

-5 0 0 1 0 0

-7 0 0 1 0 0

-6 3 0 1 0 0

1 2 3 1 2 3

3 2 1 3 2 1

1 0 0 0 0 -1

0 10 0 0 -1 -10

Salida correspondiente

Caso #1: 0.00000000 1.00000000

Caso #2: 1.00000000 6.00000000

Caso #3: 3.36340601 1.00000000

Notas

Teniendo en cuenta los puntos $N(x_i, y_i, z_i)$, su centro de masa es el punto (x_c, y_c, z_c) , que cumple:

$$x_c = (x_1 + x_2 + \dots + x_n) / N$$

$$y_c = (y_1 + y_2 + \dots + y_n) / N$$

$$z_c = (z_1 + z_2 + \dots + z_n) / N$$

- 5) El alfabeto inglés contiene 26 caracteres y los teléfonos móviles sólo tienen diez dígitos en el teclado. Varias letras se mapean en cada tecla numérica, como se muestra en la imagen inferior, de forma que se pueda escribir un mensaje con pulsaciones repetitivas. Por ejemplo, para insertar el carácter B, se deberá presionar 22. Para introducir una secuencia de dos caracteres a partir de la misma tecla, el usuario debe hacer una pausa antes de pulsar el botón una segunda vez. Por ejemplo, 2 2 indica AA mientras que 22 indica B (se muestra un "carácter de espacio" para indicar una pausa).

Debe crear un programa que calcule la secuencia de teclas que hay que pulsar para escribir un cierto mensaje.



Entrada

La primera línea de entrada contiene un único entero N , el número de casos de prueba. Cada caso es una línea de texto con formato

mensaje_deseado

Cada mensaje consta sólo de minúsculas de la "a" la "z" y el carácter " " (espacio, que se emite presionando cero).

Salida

Para cada caso de prueba, la salida de una línea que contiene "Caso # x:", seguido por el mensaje traducido en la secuencia de pulsaciones de teclas.

Entrada de ejemplo

```
4
hi
yes
foo bar
hello world
```

Salida

```
Caso #1: 44 444
Caso #2: 999337777
Caso #3: 333666 6660 022 2777
Caso #4: 4433555 555666096667775553
```

- 6) Yuki es una chica con suerte. Su abuela Chika es una encantadora anciana, propietaria de una granja de ovejas, experta repostera y ninja de la costura, y tiene un avellanero detrás de casa. Yuki pide avellanas continuamente, pero su sabia abuela no permite que Yuki coma demasiadas para que no caiga enferma.

Chika es aficionada a los números binarios, así que propuso este juego a Yuki: ella diría un número N y Yuki escogería dos números, x e y , tales que $x + y = N$. Entonces, Chika daría a Yuki tantas avellanas como la cantidad de unos que tengan x e y en base 2. Por ejemplo, si Chika propone 7 y Yuki escoge 3 y 4, recibiría 2 (porque 3 en base 2 es 11) y 1 (porque 4 en base 2 es 100): 3 avellanas.

Yuki quiere saber la máxima cantidad de avellanas que podría obtener de Chika.

Entrada: La primera línea es el número de casos, y a continuación, cada caso en una nueva línea (un número entero, entre 0 y 10^{19} , que puede no caber en un entero de 32 bits).

Salida: Para cada caso, una línea conteniendo "Caso #x: M", donde x es el número de caso (desde 1) y M es el número máximo de avellanas.

Ejemplo de entrada:

```
1
6
2135
```

Salida correspondiente:

```
Caso #1: 1
Caso #2: 4
Caso #3: 14
```

- 7) A partir de una cadena de texto recibida, se debe informar cuántas veces aparece en ella la palabra "hola" como subsecuencia formada a partir de sus letras en orden.

Por ejemplo, si se recibe la palabra "Hhoola" la respuesta sería 4, porque se podría tomar la primera H con la primera O (y con la L y con la A), la primera H con la segunda O, la segunda H con la primera O, o bien la segunda H con la segunda O. Si se recibe "hobla", la respuesta sería 1. Si se recibe "ohla", la respuesta sería 0, porque tras la H no hay ninguna O que permita completar la secuencia en orden.

Como el resultado puede ser un número muy grande, se deberá mostrar sólo las 4 últimas cifras. Si el resultado tiene menos de 4 cifras, se deberá completar con ceros por la izquierda hasta que sean exactamente 4 cifras.

Entrada

La primera línea de entrada contiene un único entero N, que es el número de casos de prueba. Las siguientes N líneas contienen los casos de prueba, uno por línea.

Salida

Para cada caso de prueba, la salida será

Caso #X: nnnn

donde X es el número de caso de prueba, a partir de 1, y nnnn son las cuatro últimas cifras de resultado (rellenas con ceros por la izquierda si es necesario).

Límites

$1 \leq N \leq 100$

Conjunto de datos pequeño

longitud de cada línea ≤ 30

Conjunto de datos grande

longitud de cada línea ≤ 500

Entrada de ejemplo

```
h olaa
hoho la
Hola, hola, hola, Que tal estas? Hola. Bienvenido a este lugar.
hhhhhoooooIIIIaaaa es casi lo mismo que hola pero con las letras repetidas
hhhhhhhhooooooooooIIIIIIaaaaaaa
```

Salida

Caso #1: 0002

Caso #2: 0003

Caso #3: 0133

Caso #4: 1184

Caso #5: 3136

- 8) En Google hemos diseñado el mejor lenguaje, que hemos llamado Googlerense. Para traducir textos a Googlerense, basta tomar cualquier mensaje en inglés y reemplazar cada letra con otra letra del alfabeto inglés. Esta asignación es de uno a uno y sólo uno, lo que significa que una misma letra de entrada siempre será reemplazada con una misma de salida, y que diferentes letras de entrada siempre se reemplazan con letras de salida diferentes. Una letra puede ser sustituido por ella misma. Los espacios se dejan tal cual.

Por ejemplo (y ésta es una pista), nuestro impresionante algoritmo de traducción incluye las tres siguientes asignaciones: 'a' -> 'y', 'o' -> 'e' y 'z' -> 'q'. Esto significa que "a zoo" se convertirá en "y qee".

El Googlerense se basa en las mejores asignaciones de reemplazo posibles, por lo que estas asignaciones nunca cambiarán. Siempre serán las mismas, en todos los casos de prueba. No diremos el resto de nuestro mapeo, porque eso haría que el fuera problema demasiado fácil, pero los ejemplos que verás más abajo te pueden ayudar.

Dado un texto en Googlerense, ¿puedes traducirlo a volver al texto normal?

Como datos de prueba, este problema sólo tiene una "entrada pequeña". Una vez que haya resuelto con esos datos, habrás terminado de solucionar este problema.

Entrada

La primera línea de la entrada da el número de casos de prueba, T. Luego siguen T casos de prueba, uno por línea.

Cada línea se compone de una cadena G en Googlerense, compuesto por una o más palabras que contengan las letras de la 'a' a la 'z'. Habrá exactamente un espacio entre palabras consecutivas y no habrá espacios al comienzo o al final de una línea.

Salida

Para cada caso de prueba, la salida debe ser una línea que contenga "Case # X: S", donde X es el número de caso y S es la cadena inglesa en que se convierte la original Googlerense G.

Límites

$1 \leq T \leq 30$

G contiene un máximo de 100 caracteres.

No se garantiza que el texto sea inglés correcto.

Entrada de ejemplo

```
ejp mysljylc kd kxveddknmc re jsicpdrysi
rbcpc ypc rtsra dkh wyfrepkym veddknkmkrkcd
de kr kd eoya kw aej tysr re ujdr lkgc jv
```

Salida

Case #1: our language is impossible to understand

Caso #2: there are twenty six factorial possibilities

Caso #3: so it is okay if you want to just give up

- 9) *Historial ayer borrado,
anteayer hubo pecado.*

El texto anterior es un pareado: una estrofa con dos versos que riman entre sí con rima consonante. ¿Sabrías hacer un programa que identifique distintos tipos de estrofa?

En concreto, nos bastará con identificar las rimas (no tendremos en cuenta el número de sílabas de cada verso), existiendo dos rimas distintas:

Rima consonante: se dice que entre dos versos hay rima consonante cuando todos los sonidos, tanto vocales como consonantes, riman. Para las comparaciones se tienen en cuenta todos los sonidos a partir de la última vocal acentuada.

Rima asonante: similar a la anterior pero únicamente riman las vocales. Por ejemplo, el siguiente cuarteto de Diego de Silva y Mendoza:

Una, dos, tres estrellas, veinte, ciento, (A)
mil, un millón, millares de millares, (B)
¡válgame Dios, que tienen mis pesares (B)
su retrato en el alto firmamento! (A)

tiene esquema ABBA consonante, pues coinciden las vocales y consonantes del primer y último verso, así como las del segundo y tercero.

Nos piden ser capaces de identificar los siguientes tipos de estrofa:

De dos versos:

Pareado: rima consonante AA.

De tres versos:

Terceto: rima consonante en el primer y último verso (A-A). Ten en cuenta que AAA no se considerará terceto.

De cuatro versos:

Cuarteto: rima consonante ABBA.

Cuarteta: rima consonante ABAB.

Seguidilla: rima asonante en los pares (-a-a). Ten en cuenta que otras combinaciones con más rimas o con rima consonante en lugar de asonante (por ejemplo -aaa o -A-A) no se consideran seguidillas.

Cuaderna via: rima consonante igual en todos los versos (AAAA).

Entrada

La entrada estará formada por un número indeterminado de casos de prueba. Cada caso de prueba comienza con una línea que contiene un único entero con el número de versos del siguiente poema. A continuación aparecen tantas líneas como versos contiene la estrofa a analizar. Podemos asumir que la última palabra de cada verso es llana (la vocal acentuada está en la penúltima sílaba), y que ninguno tendrá más de 70 letras. La entrada no contendrá tildes para facilitar la programación, aunque esto signifique cometer errores ortográficos. Tampoco tendremos en cuenta que distintos elementos gráficos pueden tener el mismo sonido. Es decir, un verso terminado en -aba, no rimará de forma consonante con un verso terminado en -ava.

La entrada termina cuando el siguiente caso de prueba contiene 0 versos. Para ese caso de prueba no se generará ninguna salida.

Salida

Para cada caso de prueba el programa indicará el nombre de la estrofa, utilizando mayúsculas (PAREADO, TERCETO, CUARTETO, CUARTETA, SEGUIDILLA, CUADERNA VIA) o la palabra DESCONOCIDO si no conoce la estrofa dada.

Notas

El enunciado ha hecho simplificaciones en las definiciones de las estrofas encaminadas a hacer el ejercicio más sencillo; ejemplos de esto son no considerar el número de sílabas, manejar sólo palabras llanas, tener faltas de ortografía, etc. El resultado ha sido unas definiciones que poco tienen que ver con las aceptadas en la literatura. Por favor, no utilices el programa final delante de un experto en poesía.

Entrada de ejemplo

Historial ayer borrado
anteayer hubo pecado

Esto no pega
ni con cola.

Era un simple clérigo, pobre de clerecía,
dicie cutiano missa de la sancta Maria;
non sabie decir otra, diciela cada día,
mas la sabie por uso qe por sabiduria.

Un manotazo duro, un golpe helado,
un hachazo invisible y homicida,
un empujon brutal te ha derribado.

Salida de ejemplo

PAREADO
DESCONOCIDO
CUADERNA VIA
TERCETO