UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Intro. a Ciencia de la Computación

Ejercicios
Pregrado
2019-I
Ciencia de la Computación
Laboratorio

1. Un *palíndromo* es una palabra que se lee igual hacia delante y hacia atrás.

   Escribe un programa que verifique si el texto ingresado por el usuario es palíndromo. Algunos ejemplos de diálogo de este programa serían:

```
Ingrese: reconocer
True
```

```
Ingrese: hola
False
```

2. Un *palíndromo* es una palabra que se lee igual hacia delante y hacia atrás y un *capicúa* es un número que se lee igual hacia delante y hacia atrás.

   Escribe un programa que verifique si el número o texto ingresado por el usuario es palíndromo o capicúa.

   Considerar los siguientes puntos:

   - El programa lee un número o un texto del usuario.
   - Sí el número es capicúa imprimir:
     - *"numero"* es un numero capicúa
   - Sí el número no es capicúa imprimir:
     - *"numero"* no es un numero capicúa
   - Sí el texto es palindromo, imprimir:
     - *"texto"* es un texto palindromo
   - Sí el texto no es palindromo, imprimir:
     - *"texto"* no es un texto palindromo

   Algunos ejemplos de diálogo de este programa serían:

```
Ingrese un numero o texto: 12345
12345 no es capicua.
```

```
Ingrese un numero o texto: 123321
123321 es un numero capicua.
```

```
Ingrese un numero o texto: hola
hola no es un texto palindromo.
```

```
Ingrese un numero o texto: reconocer
reconocer es un texto palindromo.
```

3. Dada una *cadena* de ADN generar su *cadena* complementaria. Respetando la correspondencia $(A-T; G-C)$ Imprima la cadena complementaria.

   Veamos algunos ejemplos:

```
input:
cccaactgaa

output:
gggttgactt
```

```
input:
gtactcgggg
output:
catgagcccc
```

4. Dada *dos cadenas* generar una tercera *cadena* que sea el resultado de no considerar en la segunda cadena los caracteres de la primera. Imprimir el resultado.

   Veamos algunos ejemplos:

```
input:
abcde
las casas verdes vuelan
output:
ls ss vrs vuln
```

```
input:
gato de schrodinger
la fascinante historia de la mecanica cuantica.
output:
lflmu.
```

5. Generar un traductor de palabras al latinBot(un lenguaje artificial). Se recibe una **palabra** en español y se traduce de la siguiente forma: Se toma la **primera letra** y se pone al final, ademas se le agrega la palabra **bot**. Tener en cuenta que la primera palabra en el lenguaje latinBot siempre está en mayuscula.

   Veamos algunos ejemplos:

```
input:Hola
output:Olahbot
```

```
input:mundo
output:Undombot
```

6. El usuario ingresa 3 strings.: *string1*, *string2*, *string3*.

   El usuario ingresa 6 números enteros: *n1*, *n2*, *n3*, *n4*, *n5*, *n6*.

   Los números n1 y n2 representan los índices de inicio y fin de los caracteres de una subcadena del string1. Los números n3 y n4 representan los índices de inicio y fin de los caracteres de una subcadena del string2. Los números n5 y n6 representan los índices de inicio y fin de los caracteres de una subcadena del string3.

   Las 3 subcadenas se concatenan y forman una nueva cadena llamada *sumaCadenas*. Imprima la cadena *sumaCadenas*.

   Algunos ejemplos de diálogo de este programa serían:

```
Ingrese cadena 1: Los amigos
Ingrese cadena 2: son
Ingrese cadena 3: la mejor compania

Ingrese n1: 1
Ingrese n2: 3
Ingrese n3: 1
Ingrese n4: 3
Ingrese n5: 1
Ingrese n6: 4
osona m
```

```
Ingrese cadena 1: Bienvenidos
Ingrese cadena 2: sean
Ingrese cadena 3: todos

Ingrese n1: 1
Ingrese n2: 4
Ingrese n3: 2
Ingrese n4: 4
Ingrese n5: 1
Ingrese n6: 5
ienanodos
```

7. Realizar un programa en el cual se ingresen **N** palabras y se analicen todas las palabras indicando cuantas palabras cuentan con más de 3 vocales. Algunos ejemplos de diálogo de este programa serían:

```
Ingresan: 6 javier elevador alberti comienza uber monitor
Resultado: 4
```

```
Ingresan: 3 andres celular ana
Resultado : 1
```

8. Escribir un programa en Python, que permita validar la clave de un usuario, considerando lo siguiente:

   - Debe tener mínimo 8 caracteres.
   - Debe contener al menos un digito.
   - Debe contener al menos una mayúscula.
   - Debe contener al menos una minúscula.
   - Debe contener al menos uno de los siguientes caracteres $,&

   Si cumple las condiciones debe mostrar el mensaje "Correcto", en caso contrario mostrar el mensaje "Incorrecto" Ejemplo:

```
Ingrese Clave: Uteclandia1$
Correcto
```

```
Ingrese Clave: Utec
Incorrecto
```

9. Diseñe e implemente una función *puntaje* que reciba tres parámetros: una lista que contiene unicamente tres valores: G (ganado), E (empatado) y P (perdido). Los dos parámetros restantes serán dos enteros que representen el puntaje a asignar en caso de partido ganado y empatado (el partido perdido siempre valdrá cero). En caso estos dos parametros no se envíen al momento de invocar a la función se deberá considerar tres puntos por partido ganado y un punto por partido empatado. Puede asumir que los únicos valores que vendrán en la lista son "G", "E" o "P".

```
lista = ["G", "G", "P", "P", "E", "G"]
print(puntaje(lista, 5, 2))
print(puntaje(lista))



17
10
```

10. Diseñe e implemente un algoritmo que calcule la suma de cubos de los dígitos de un número divisible por 3. Para ello, ingrese un número entero y determine si es divisible por 3. En caso no lo sea, pida otro numero que sea divisible por 3. Luego calcule la suma

de cubos de los dígitos del número ingresado. A continuación, calcule la suma de cubos de los dígitos del resultado anterior. Continúe el procedimiento hasta que el cálculo sea igual a 153.
¿Qué observa en el resultado?
E.g., para el número 333:

```
Ingrese un numero: 333
la suma de cubos de 333 es: 81
la suma de cubos de 81 es: 513
la suma de cubos de 513 es: 153
la suma de cubos de 153 es: 153
```

11. Erika Fernandez Bodas - Eventos, tiene más de 5 años en el rubro de eventos. Ofrece servicios de calidad, deliciosos catering y una decoración personalizada. Realizan todo tipo de eventos como coffee breaks, aniversarios, fiestas de fin de año, bodas, celebraciones de cumpleaños, despedidas de solteras, comidas de empresa, cenas temáticas y etc.

    Todos los eventos de Erika están calendarizados, pero a ella le interesaría tener la cantidad de días que falta para cada evento.

    Procedimiento:
    Considerando que hoy es 17/01/2019 y la fecha del evento es: 16/05/2019, calculamos los días por mes:

    - Enero: 14 días
    - Febrero: 28 día
    - Marzo: 31 días
    - Abril: 30 días
    - Mayo: 16 días

    Total días que falta: 14 + 28 + 31 + 30 + 16 = **119 días**.

    Ayudemos a Erika a obtener la cantidad de días que falta para un evento. Escribe un programa que permita al usuario ingresar el día, mes y año de una fecha, y el programa debe obtener la cantidad de días, y al final debe imprimir la cantidad de días.

    Para esto tu programa debe implementar y usar la siguiente función:

    - `obtenerDias` recibe como parámetro `día, mes y año` y retorna `cantidad de días`, considere que hoy es 17 de enero del 2019.

    Algunos ejemplos de diálogo de este programa serían:

```
Ingrese d a: 16
Ingrese mes: 5
Ingrese a o: 2019
Dias faltantes:  119
```

```
Ingrese d a: 20
Ingrese mes: 9
Ingrese a o: 2019
Dias faltantes:  246
```

12. El sistema hexadecimal es el sistema de numeración posicional que tiene como base el 16. Su uso actual está muy vinculado a la informática y ciencias de la computación donde las operaciones de la CPU suelen usar el byte u octeto como unidad básica de memoria.

    Dado que el sistema usual de numeración es de base decimal y, por ello, sólo se dispone de diez dígitos, se adoptó la convención de usar las seis primeras letras del alfabeto latino para suplir los dígitos que hacen falta. A continuación se muestra la tabla de equivalencia.

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

    El método para convertir un número decimal a hexadecimal es:

    - Dividimos el número entre 16.
    - Si el cociente es mayor o igual que 16, lo dividimos entre 16.
    - Continuamos así hasta obtener un cociente menor que 16.

    O puedes crear o aplicar otro método o algoritmo.

    Escribe un programa que permita al usuario ingresar un número decimal, y el programa debe convertir el número a Hexadecimal, y al final debe imprimir el nuevo número.

    Para esto tu programa debe implementar y usar la siguiente función:

    - `getHexadecimal` recibe como parámetro `número en base decimal` y retorna `el número equivalente en base hexadecimal`.

    Algunos ejemplos de diálogo de este programa serían:

```
Ingrese un n mero decimal: 15963
El equivalente en Hexadecimal es: 3E5B
```

```
Ingrese un n mero decimal: 987654321
El equivalente en Hexadecimal es: 3ADE68B1
```

13. Los números perfectos son aquellos que son iguales a la suma de todos sus divisores exceptuando el mismo número, por ejemplo el número 6 es un número perfecto porque $6 = 1 + 2 + 3$. Por otro lado, un número primo es aquel que sólo es divisible por el 1 y por el mismo número, por ejemplo el número 7.

Escribe un programa en python que reciba 2 números *min* y *max*, a partir de estos números tu programa deberá mostrar todos los números entre *min* y *max* que son perfectos o primos. Para esto tu programa debe implementar y usar las siguientes funciones:

- `es_perfecto` recibe como parámetro un número entero positivo y retorna `True` si el número es perfecto o `False` en caso contrario.
- `es_primo` recibe como parámetro un número positivo y retorna `True` si el número es primo o `False` en caso contrario.

Adicionalmente, si no hay ningún número perfecto o primo en el rango de *min* y *max*, deberá mostrar un mensaje apropiado. Algunos ejemplos de este programa serían:

```
Ingrese min: 1
Ingrese max: 7
2 3 5 6
```

```
Ingrese min: 8
Ingrese max: 10
No hay numeros perfectos ni primos
```

14. Para dar solución a este problema se requiere lo siguiente:

    - Crea una función que reciba una cadena de caracteres y te devuelva el caracter que más se repite.
    - Crea un programa que pida al usuario 3 frases.
    - El programa mostrará en pantalla una cadena formada por lo caracteres más repetidos de cada una de las tres frases

    Algunos ejemplos de diálogo de este programa serían:

```
Ingrese la frase 1: mi mama me mima
Ingrese la frase 2: ojos que no ven, ciego es
Ingrese la frase 3: alabama
moa
```

15. Diseñe e implemente una función *extrae_intervalo* que reciba tres parámetros: una lista y dos enteros opcionales, el primero de ellos representa el límite inferior y el segundo el superior. La función debe retornar una lista que contenga aquellos elementos que se encuentran dentro del intervalo formado por el límite inferior y superior (incluyendo los límites). En caso de que no se envien los límites al momento de invocar la función deberá considerarse 0 como límite inferior y 100 como límite superior.

```
lista = [1, 80, 15, 30, 25, 90, 110, 105, 180, 200, 10]
print(extrae_intervalo(lista, 10, 70)
print(extrae_intervalo(lista)
```

```
15 30 25 10
1 80 15 30 25 10
```

16. Usted se encuentra trabajando en una empresa de educación y ha sido encargado con la misión de calcular la rentabilidad que un taller que están ofreciendo, esto le permitirá tomar decisiones sobre el negocio. Para ello, debe entender bien la dinámica del negocio y cuando lo haya entendido debe automatizar el proceso de cálculo en un programa en python.

    Diseñe e implemente un programa que obedezca la siguiente lógica:

    - El programa debe recibir la cantidad de estudiantes matriculados en el taller.
    - El programa debe llamar una función llamada costos fijos que devuelta el costo fijo total sumando todos los costos fijos:
    - Alquiler: $ 160
    - Publicidad: $ 50
    - El programa debe llamar una función llamada costos variables que devuelva el costo variable total:
    - Hasta los 10 alumnos el profesor cobra: $ 10.00 por hora
    - Desde los 11 hasta los 15 alumnos el profesor cobra: $ 13.00 por hora
    - Desde los 16 a 20 el profesor cobra: $. 17.00
    - A partir de 21 alumnos cobra: $ 20.00
    - Todos los talleres duran 16 horas.
    - El programa debe llamar una función llamada 'evaluorenta' que:
    - Debe calcular los ingresos en función al precio de venta ($ 80.00 por estudiante).
    - Restarle los costos totales (suma de los costos fijos y los costos variables)
    - Devolver 'es rentable' si el resultado es mayor a 0 y 'no es rentable' si el resultado es negativo.
    - El programa debe usar un solo módulo donde estén creadas las funciones solicitadas.
    - El resultado de la función 'evaluorenta' debe ser impreso al final del programa.

    Algunos casos de prueba:

```
Ingrese cantidad de estudiantes: 20
Resultado:  es rentable
```

```
Ingrese cantidad de estudiantes: 3
Resultado: no es rentable
```

17. Escribe un programa en Python que permita al usuario ingresar números enteros positivos hasta que ingrese el número 0. Luego de esto el programa deberá mostrar al usuario los números pares ingresados como se muestra en el siguiente ejemplo:

```
6
4
3
1
5
8
0
[6, 4, 8]
```

18. Estamos en la final del torneo interescolar de programación más grande del país y ya tenemos los resultados de las evaluaciones individuales del comité técnico para cada equipo, el problema es que los puntajes totales todavía no han sido consolidados y por ello se ha solicitado su colaboración.

    Se cuenta con la siguiente información:

```
Equipos = ['Hands','Pae','Choice','Huatcher']
Resultados = [[3,1,3,2],[3,2,3,2],[3,3,1,2],[3,3,3,3]]
```

    Cada elemento de la lista `Resultados` es una sublista que obedece a la siguiente lógica:

    - Primer elemento corresponde a impacto.
    - Segundo elemento a diseño.
    - Tercero a potencial.
    - El último a innovación.
    - La calificación del equipo corresponde a la sumatoria de las cuatro calificaciones.

    Para esta lista, se solicita que el programa:

    - Genere e imprima la calificación de todos los equipos.
    - Detecte el puntaje más alto en diseño e imprima el nombre del equipo. Si hay empate, debe imprimir el nombre del que tiene puntaje total más alto.
    - Genere e imprima el puntaje promedio correspondiente a innovación.

    Caso de prueba:

```
Resultadostotales = [9,10,9,12]
Puntajedisenoalto = Huatcher
Promedioinnovacion = 2.5
```

19. Diseñar y escribir un programa que solicite un valor n y que genere una lista de tamaño n cuyos valores sean números enteros aleatorios entre 1 y $\frac{n}{3}$ y que genere una nueva lista con todos aquellos números que se repitan 3 o más veces.

```
Ingrese la cantidad: 10
Lista original:  [3, 3, 2, 1, 2, 3, 3, 3, 2, 1]
Lista solicitada:  [2, 3]
```

```
Ingrese la cantidad: 15
Lista original:  [4, 5, 5, 5, 3, 2, 4, 3, 4, 3, 3, 1, 5, 5,
   4]
Lista solicitada:  [3, 4, 5]
```

```
Ingrese la cantidad: 6
Lista original:  [1, 1, 1, 2, 2, 1]
Lista solicitada:  [1]
```

20. Diseñar y escribir un programa que solicite un valor n y que genere una lista de tamaño n cuyos valores sean números aleatorio entre 1 y $\frac{n}{3}$ y que genere una nueva lista con todos aquellos números donde el siguiente numero sea igual al numero actual + 1.

```
Ingrese la cantidad: 10
Lista original: [3, 2, 2, 3, 2, 3, 3, 1, 3, 3]
Lista solicitada: [2, 2]
```

```
Ingrese la cantidad: 15
Lista original: [2, 2, 2, 1, 1, 1, 4, 4, 4, 1, 2, 1, 4, 1, 1]
Lista solicitada: [1]
```

```
Ingrese la cantidad: 4
Lista original: [1, 1, 1, 1]
Lista solicitada: []
```

21. El profesor esta llevando a todos sus alumnos a comer a cierto restarurante, el cual todavia no ha sido definido, por lo que permite que el primero de la lista sea quien eliga el restaurante para ir a comer. Dado que es una decisión importante, el primero de la lista será el alumno que tenga la más alta nota del curso.

    Se le pide desarrollar un algoritmo que recibe como dato de entrada una lista de notas y debe generar otra lista en donde la nota mas alta aparezca al inicio de la lista y las demás notas mantegan su orden original.

    [frame=single] Ingresar lista: 15,11,12,13,18,08,14 Nueva lista: [18, 15, 11, 12, 13, 8, 14]

22. Escribir un programa que pida del usuario el número de filas **M**, y el número de columnas **N** y que por medio de una función llamada *"generar_matriz"* se imprima la matriz donde todos los elementos de la matriz tengan el valor por default establecido en la función.

    Considerar los siguientes puntos:

- La función *"generar_matriz"* recibe tres parámetros:
  - *filas*, como Número de Filas (Parámetro Obligatorio)
  - *columnas*, como Número de Columnas (Parámetro Obligatorio)
  - *elemento*, con el valor default igual al string *"a"* (Parámetro NO Obligatorio)
- La función *"generar_matriz"* tiene que ser llamada con los parámetros obligatorios.

```
python ejercicio1.py
Ingrese el numero de Filas: 5
Ingrese el numero de Columnas: 6
La matriz con 5 filas y 6 columnas es la siguiente:
a a a a a a
a a a a a a
a a a a a a
a a a a a a
a a a a a a
```

23. La Liga 1 2019 (por razones de patrocinio Liga 1 Movistar) es la edición número 103 de la Primera División del Perú y la primera bajo la denominación de Liga 1. La Federación Peruana de Fútbol (FPF) organiza y controla el desarrollo del torneo a través de la Comisión Organizadora de Competiciones. El campeonato está constituido de dos torneos cortos: Torneo Apertura y Torneo Clausura; semifinales que serán jugadas entre los ganadores de los dos torneos cortos y los dos equipos que ocupen el primer y segundo lugar del acumulado; y una final entre los ganadores de estas llaves.

    Y la tabla de posiciones hasta la fecha 14 es la siguiente:

Equipo	**Jugados**	**Ganados**	**Empatados**	**Perdidos**
Deportivo Municipal	13	5	5	3
Alianza Lima	13	5	4	4
Universidad César Vallejo	13	6	2	5
Universidad Técnica de Cajamarca	13	4	7	2
Deportivo Binacional	14	11	0	3
Sporting Cristal	13	8	4	1

    El Federación Peruana de Fútbol le pide ayuda para generar las siguientes datos y estadísticas.

    - El puntaje de cada equipo, que se obtiene con el siguiente calculo: por un partido ganado son tres puntos, por un partido empatado es un punto y por un partido perdido cero puntos.
    - El equipo con el puntaje más alto.
    - El equipo con el puntaje más bajo

Convierta los datos proporcionados (Equipos y los puntajes) en una lista de listas, y escriba un programa que genere e imprima los datos y las estadísticas solicitadas.

Las estadísticas serían:

```
Deportivo Municipal: 20
Alianza Lima: 19
Universidad C sar Vallejo: 20
Universidad T cnica de Cajamarca: 19
Deportivo Binacional: 33
Sporting Cristal: 28
El equipo con el mayor puntaje: Deportivo Binacional
El equipo con el menor puntaje: Alianza Lima
```

24. Escribe un programa en Python que permita al usuario ingresar dos frases, y luego ponga las palabras que empiezan con vocal de la primera frase en una lista, y las palabras que empiecen con consonante de la segunda frase en otra lista, y luego imprima ambas listas, como en este ejemplo:

```
frase 1: El burrito parrandero
frase 2: Me gusta la bilirrubina
["El"]
["Me","gusta","la","bilirrubina"]
```