 PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA Facultad de Ingeniería y Ciencias Agrarias	EXÁMEN TEÓRICO	01 / 12 / 2020	FINAL	
		INFORMÁTICA GENERAL		
	En máquina, presencial virtual y a distancia			TEMA 1

INTRODUCCIÓN

Exam.net nos ofrece, el siguiente menú:


- 1. **Área de escritura:** En esta sección deberá contestar la opción correcta de cada uno de los ejercicios planteados en el enunciado. Alcanza con que indique claramente la letra de la opción elegida para cada ejercicio de opción múltiple, por ejemplo **Ej 1 - opción b)** , etc. ; y que copie y complete el código para los ejercicios donde debe llenar el blanco.
- 2. **EXAMEN ...:** Aquí vas a encontrar el texto con la consigna del examen.
- 3. **Chat de profesores:** Podrás realizar preguntas a tus profesores utilizando este chat.
- 4. **Enviar examen:** Es el link que te lleva al envío del examen, donde estarías realizando la entrega de tu examen. *En síntesis..terminaste el examen !!!.*

ACLARACIÓN: En esta parte del examen no se puede usar el entorno de programación y debe contestar en base a su conocimiento de la materia.

El examen teórico consta de 4 ejercicios y tiene una duración máxima de 20 minutos desde el momento que se comienza. Para aprobar se necesitan 3 ejercicios correctos.

IMPORTANTE: Sólo podemos usar los siguientes elementos del lenguaje para resolver los problemas en los **FINALES** o evaluaciones en la cátedra. Y sujeto a indicaciones del profesor.

```
- DELIMITADORES:      #   '   "   ,   :   .   \   (   )   [   ]   {   }
- OPERADORES:          +   -   *   **  /   //  <   >   <=  >=  ==  !=   =   +=   -=
- PALABRAS RESERVADAS:  and   def   elif  else  False  for   if   import  in   None
not   or   return  True   while
- FUNCIONES INTEGRADAS:  bool()  chr()  dict()  float()  id()  input()  int()  len()
list()  max()  min()  open()  ord()  print()  range()  round()  str()  tuple()
type()
- FUNCIONES LIBRERÍA math:  math.pow()  math.sqrt()  - FUNCIONES LIBRERÍA random:
random, randint()
- MÉTODOS DE STRING:      .format()  .split()
- MÉTODOS DE LIST:        .append()  .insert()  .pop()  .remove()
- MÉTODOS DE DICT:        .get()  .keys()  .pop()  .values()
- MÉTODOS DE FILE:        .close()  .readline()  .readlines()  .write()
```

 PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA Facultad de Ingeniería y Ciencias Agrarias	EXÁMEN TEÓRICO	01 / 12 / 2020	FINAL	
		INFORMÁTICA GENERAL		
	En máquina, presencial virtual y a distancia			TEMA 1

EJ.T.1

```
def fun(nombres):
    nombres.insert(1, 'Charly')
    nombres = nombres[:-1]

def main():
    names = [ 'Tom', 'Tommy', 'Tomas' ]
    fun(names)
    print(names)
main()
```

¿Qué imprime al ejecutar el código?

- a) ['Tom', 'Tommy', 'Tomas']
- b) ['Tom', 'Charly', 'Tommy']
- c) ['Tom', 'Charly', 'Tommy', 'Tomas']
- d) Da error al ejecutar el código
- e) Ninguna de las anteriores


EJ.T.2

```
def main():
    num= int('123456789')
    b=10
    e=3
    print( ((num//(b**e)) % (b**e)) % (b**e) )

main()
```

¿Qué imprime al ejecutar el código?

- a) 456
- b) 3456
- c) 0
- d) 1
- e) Da error al ejecutar el código
- f) Ninguna de las anteriores

 PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA Facultad de Ingeniería y Ciencias Agrarias	EXÁMEN TEÓRICO	01 / 12 / 2020	FINAL	
		INFORMÁTICA GENERAL		
	<i>En máquina, presencial virtual y a distancia</i>		TEMA 1	


EJ.T.3

```
def fun(lst, val):
    largo=len(lst)
    for i in range(0,largo):
        if lst[i]==val:
            lst.pop(i)
def main():
    lst = [1,2,3,4,4,3,2,1]
    fun(lst,2)
    print(lst)

main()
```

¿Qué imprime al ejecutar el código?

- a) [1,2,3,4,4,3,2,1]
- b) [1,3,4,4,3,1]
- c) [2,2]
- d) []
- e) Da error al ejecutar el código
- f) Ninguna de las anteriores

 PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA Facultad de Ingeniería y Ciencias Agrarias	EXÁMEN TEÓRICO	01 / 12 / 2020	FINAL	
		INFORMÁTICA GENERAL		
	En máquina, presencial virtual y a distancia			TEMA 1

EJ.T.4

Completar las líneas en blanco (_____), sin agregar ni cambiar renglones al código mostrado de la función cargarDi.

La función cargarDi recibe por parámetro un lista lst y debe retornar un diccionario cuyo valores del diccionario son cada uno de los valores de la lista lst, y cuya clave de cada item del diccionario deberá ser un número entero entre 1 y 100 generado en forma aleatoria.

Importante: Tener en cuenta que si al generar un número aleatorio, este ya está usado como clave en el diccionario, se deberá crear otro número aleatorio hasta encontrar uno que no esté usado como clave.

```
import random

def cargarDi (lst):
    di=_____
    for item in lst:
        clave = _____
        _____ clave in _____ :
            _____ = random.randint(___,___)
        di[_____]= item
    return _____

def main():
    lst=['Tom', 'Charly', 'Tommy', 'Tomas']
    di = cargarDi(lst)
    print(di)

main()
```

Algunas posible salida esperadas para el código completo podrían ser:

- {45: 'Tom', 30: 'Charly', 59: 'Tommy', 56: 'Tomas'}
- {83: 'Tom', 92: 'Charly', 33: 'Tommy', 59: 'Tomas'}
- {4: 'Tom', 11: 'Charly', 54: 'Tommy', 10: 'Tomas'}

FIN TEÓRICO

Puede comenzar el examen práctico