



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE CIENCIAS NATURALES Y MATEMÁTICAS
DEPARTAMENTO DE MATEMÁTICAS
ICM00794 - FUNDAMENTOS DE COMPUTACION
1ra Evaluación I TÉRMINO 2015-2016 – 07/Julio/2015



MATRICULA:NOMBRE: **Luis Rodríguez Ojeda** PARALELO:

COMPROMISO DE HONOR: Al firmar este compromiso, reconozco que el presente examen está diseñado para ser resuelto de manera individual, que puedo usar un lápiz o esferográfico; que sólo puedo comunicarme con la persona responsable de la recepción del examen; y, cualquier instrumento de comunicación que hubiere traído, debo apagarlo y depositarlo en la parte anterior del aula, junto con algún otro material que se encuentre acompañándolo. Además no debo usar calculadora alguna, consultar libros, notas, ni apuntes adicionales a los que se entreguen en esta evaluación. Los temas debo desarrollarlos de manera ordenada.
Firmo el presente compromiso, como constancia de haber leído y aceptado la declaración anterior.

Firma

Tema 1. (25 puntos) Un procedimiento matemático para multiplicar dos números enteros con valores entre 1 y 1000 se describe en el siguiente ejemplo:

Escriba un algoritmo que verifique que esta regla se cumple para cada producto $a \times b$.

Las variables a y b son enteros positivos entre 1 y un valor que puede llegar hasta 1000. Muestre la respuesta mediante un mensaje.

Se desea conocer el producto de dos números:	997 y 991
Obtenga los resultados de las restas:	1000-997 = 3 1000-991 = 9
Sume los resultados de las restas:	3 + 9 = 12
Reste de 1000 el resultado de la suma anterior:	1000-12 = 988
Multiplique este resultado por 1000:	988 x 1000 = 988000
Multiplique los resultados de las restas iniciales:	3 x 9 = 27
La suma de los dos últimos resultados es el producto deseado:	988000+27 = 988027

Rúbrica: ingreso y validación (5 puntos), procedimiento matemático (5 puntos), comprobar que cumple para a (5 puntos), comprobar que cumple para b (5 puntos), Algoritmo estructurado (5 puntos)

#Multiplicación de dos números

```
while True:
```

```
    n=int(input('Valor final entre 1 y 1000: '))
```

```
    if 1<=n<=1000:
```

```
        break
```

```
prueba=True
```

```
for a in range(1,n+1):
```

```
    for b in range(1,n+1):
```

```
        ra=1000-a
```

```
        rb=1000-b
```

```
        s=ra+rb
```

```
        r=1000-s
```

```
        x=1000*r
```

```
        y=ra*rb
```

```
        p=x+y
```

```
        if p!=a*b:
```

```
            prueba=False
```

```
if prueba:
```

```
    print('Si se cumple la propiedad')
```

```
else:
```

```
    print('No se cumple la propiedad')
```

Primero se ingresa y valida n

Los dos ciclos for generan todos los pares a, b con valores entre 1 y n para verificar si cumplen la propiedad propuesta.

Si alguna pareja a, b no cumple la propiedad, la variable **prueba** toma el valor **falso**.

Finalmente se muestra el resultado mediante un mensaje

```
>>> ===== RESTART =====
```

```
>>>
```

```
Valor final: 1000
```

```
Si se cumple la propiedad
```

Tema 2. (25 puntos) El 5 de junio de 2015, la Asamblea Nacional recibió un proyecto de ley en materia económica con calidad de urgente conocido como "Ley de Herencias". El proyecto reformaría el porcentaje del impuesto sobre las herencias, el cual es progresivo basado en equivalentes al **salario básico** (SB). Al Año 2015, el salario básico (SB) se estableció en \$354,00.

Para el impuesto, considerando solo el caso por cada heredero directo, dado un valor de herencia y el número de herederos, se calcula el valor **equivalente** en salarios básicos (SB). Usando la tabla se determina el rango aplicable, para calcular el impuesto se suman: el "impuesto a fracción básica" y la "tarifa" multiplicada por el excedente; el resultado buscado es el valor convertido a dólares.

Elabore un algoritmo que dado el **valor de herencia**, el **número de herederos** y el salario básico en dólares (\$), calcule el valor propuesto del **impuesto** en dólares.

Ejemplo:

Valor de herencia (\$)	\$ 200.000,00
Número de herederos:	2
Salario básico	\$ 354,00
Equivalencia (SB):	$(200.000,00/2)/354,00 = 282,4859$
rango aplicable (SB)	200 a 400
Fracción Excedente (SB)	$282,4859 - 200 = 82,4859$
Impuesto en SB	$2,5 + 82,4859 * (7,50\%) = 8,6864$
Impuesto por heredero en (\$)	$8,6864 * \$354 = \$3.074,99$

"Artículo 4. Tarifas para liquidar este impuesto son:

a. En caso de los beneficiarios sea legitimarios del causante o donante, según corresponda, se aplicará la siguiente tabla:"

Tabla expresada en número de Salarios Básicos (SB)

Rango Aplicable		Impuesto	
Fracción básica	Exceso hasta	Fracción Básica	Tarifa Fracción Excedente
-	100	-	-
100	200	-	2,50%
200	400	2,5	7,50%
400	800	17,5	17,50%
800	1600	87,5	32,50%
1600	En adelante	347,5	47,50%

Nota: Se usa solo un escenario para la evaluación algorítmica tomando como referencia de cálculo un literal del artículo.

Referencias: Oficio No. MINFIN-DM-2015-0333, 05-jun-2015, Quito. www.eluniverso.com. Conozca el proyecto de ley para el impuesto a las herencias en

Ecuador, 2015/06/05. http://www.eluniverso.com/sites/default/files/archivos/2015/06/rd_215732correa_215732_304958.pdf

Rúbrica: ingreso (5 puntos), transformar unidades (5 puntos), determinar parámetros por rangos (10 puntos), resultado (5 puntos)

#Ley de herencias

```

vh=float(input('Valor de la herencia: '))
nh=int(input('Número de herederos: '))
sb=float(input('Ingreso el salario básico: '))
esb=vh/(2*sb)
if esb<=100:
    ifb=0
    tfe=0
    fe=esb-0
elif 100<esb<=200:
    ifb=0
    tfe=0.025
    fe=esb-100
elif 200<esb<=400:
    ifb=2.5
    tfe=0.075
    fe=esb-200
elif 400<esb<=800:
    ifb=17.5
    tfe=0.175
    fe=esb-400
elif 800<esb<=1600:
    ifb=87.5
    tfe=0.325
    fe=esb-800
else:
    ifb=347.5
    tfe=0.475
    fe=esb-1600
isb=ifb+fe*tfe
iph=isb*sb
print('Impuesto por heredero: ',iph)

```

Variables:

esb: Equivalente en salarios básicos

ifb: Impuesto a la fracción básica

tfe: Tarifa fracción excedente

fe: Fracción excedente

Los valores se asignan según la tabla del rango aplicable.

Finalmente con los valores del caso seleccionado, se calcula

isb: impuesto en salario básico

iph: impuesto por heredero

```
>>> ===== RESTART =====
```

```
>>>
```

```
Valor de la herencia: 200000
```

```
Número de herederos: 2
```

```
Ingreso el salario básico: 354
```

```
Impuesto por heredero: 3075.0
```

Tema 3. (30 puntos) Se registraron **voluntarios** para colaborar en el evento del Parque Samanes por la visita del Papa a Guayaquil.

Para **capacitar** a los voluntarios, identificados por un número, se requiere formar grupos equitativos (igual cantidad) para: orientar a las personas en el ingreso y salida, portar letreros y banderas, distribuir agua, prestar atención a discapacitados y quienes pudiesen tener algún problema de salud, entre otras.

Realice un algoritmo para distribuir a **n** voluntarios en **m** tipos de capacitación en forma **aleatoria, no repetida y equitativa**; luego muestre el listado que indica la capacitación que recibirá cada voluntario.

Nota: Un voluntario solo puede asistir a una capacitación. Suponga que **n** es múltiplo de **m**.

Ejemplo: para $n=8$, $m=4$,
 $maxtipo=8/4=2$

voluntario	capacita [voluntario]
1	3
2	1
3	4
4	3
5	1
6	2
7	2
8	4

Referencia: La república. Quito y Guayaquil ultiman detalles para visita del papa Francisco a Ecuador. 02.07.2015

<http://www.larepublica.ec/blog/sociedad/2015/06/30/quito-y-guayaquil-ultiman-detalles-para-visita-del-papa-francisco-a-ecuador/>

Rúbrica: ingreso y validación (5 puntos), sorteo capacitación única (5 puntos), cupos de capacitación (15 puntos), mostrar resultados (5 puntos).

```
#Capacitación de voluntarios
from random import*
n=int(input('Cantidad de voluntarios: '))
m=int(input('Tipos de capacitación: '))
lista=[]
for i in range(0,n,m):
    tareas=[]
    while len(tareas)<m:
        x=randint(1,m)
        if x not in tareas:
            tareas=tareas+[x]
    for j in range(m):
        lista=lista+[tareas[j]]
for i in range(n):
    print(i+1,lista[i])
```

El vector **lista** contendrá las tareas asignadas a los **n** voluntarios.

El vector **tareas** contiene **m** números aleatorios diferentes entre 1 y **m**

El vector **tareas** es trasladado en bloques sucesivos al vector **lista** con lo que la asignación es aleatoria, diferente y equitativa.

```
>>> ===== RESTART =====
```

```
>>>
```

```
Cantidad de voluntarios: 12
```

```
Tipos de capacitación: 4
```

```
1 3
2 1
3 4
4 2
5 4
6 1
7 2
8 3
9 3
10 4
11 2
12 1
```

Tema 4. (20 puntos) Para el evento del Parque Samanes por la visita del Papa en Guayaquil el área del Parque se dividió en 32 **bloques** que tendrían el apoyo de voluntarios capacitados como se describe en el tema anterior.

Realice un algoritmo que reciba el listado de **n** voluntarios y su **tipo** de capacitación, resultado del tema anterior, y los distribuya para cada bloque del parque por **orden de lista y de forma equitativa** por tipo de capacitación.

Muestre un nuevo listado indicando los voluntarios asignados a cada bloque.

Nota: Un voluntario solo puede atender un bloque. Suponga que n es múltiplo de m.

Sugerencia: inicie asignando solo para un tipo y un bloque, revise el tipo de capacitación de cada voluntario y asigne siempre que el conteo sea menor que máximo por tipo y bloque, caso contrario cambie de bloque, reinicie los contadores y continúe asignando con el nuevo bloque hasta completar todos los voluntarios. Repita el proceso cambiando el tipo.

Ejemplo:

para $n=8$, $m=4$, bloques=2

El máximo por tipo de capacitación $= 8/4=2$

El máximo por tipo capacitación por cada bloque $= (2/2)=1$

voluntario	capacita [voluntario]	Bloque [voluntario]
1	3	1
2	1	1
3	4	1
4	3	2
5	1	2
6	2	1
7	2	2
8	4	2

Rúbrica: ingreso (5 puntos), asignar ordenadamente al bloque (10 puntos), asignar equitativamente por tipo (5 puntos)

#Asignación de voluntarios a bloques

```

n=int(input('Cantidad de voluntarios: '))
m=int(input('Tipos de capacitación: '))
lista=[]
for i in range(n):
    t=int(input('Ingrese tarea para '+str(i+1)+' : '))
    lista=lista+[t]
bloques=[]
b=1
for i in range(n):
    bloques=bloques+[0]
for i in range(1,m+1):
    for j in range(n):
        if lista[j]==i:
            bloques[j]=b
            b=b+1
            if b>32:
                b=1
for i in range(n):
    print(i+1,lista[i],bloques[i])

```

Primero se ingresa la asignación a los **n** voluntarios resultante del tema anterior.

El vector **bloques** contendrá la asignación de los voluntarios a los **32** bloques del parque.

Para que la asignación sea equitativa, se distribuye en forma sucesiva en los **32** bloques los voluntarios que tienen la misma tarea, la cual va desde 1 hasta **m**