

Turno 01:

Desarrolle un programa completo en Python que permita generar una sucesión de **17000 números enteros aleatorios, usando como semilla del generador al valor 1157 (es decir, random.seed(1157))**. Los valores de cada uno de esos 17000 números deben estar **entre 1000 y 37000 (incluidos ambos - DEBE usar random.randint(1000, 37000) para generar cada uno de estos números)**.

A modo de elemento de control, y para que cada estudiante sepa si la secuencia que está generando es la correcta, indicamos aquí cuánto debe valer la suma de todos los números generados con el mecanismo indicado:

✓ Suma de todos los números generados: 322152298

A partir de esa sucesión, el programa debe:

1. Determinar **cuántos** de esos números eran mayores o iguales a 1000 pero menores que 15000, cuál es la **suma** de los que eran mayores o iguales que 15000 y menores que 30000, y **cuántos** eran mayores o iguales que 30000.
2. Determinar el **promedio entero** de los números generados que eran **divisibles por 7 pero no por 3**. Aclaración: NO se pide el promedio redondeado, sino truncado, sin decimales.
3. Determinar el **menor** entre todos los números generados **que sean impares**.
4. Determinar el **porcentaje entero** que representa la **cantidad de números pares** generados sobre la cantidad total de números procesados. Aclaración: NO se pide el porcentaje redondeado, sino truncado, sin decimales. *Observación: en el cálculo de este porcentaje, haga primero la multiplicación que corresponda, y luego la división.*

Turno 02:

Desarrolle un programa completo en Python que permita generar una sucesión de **14000 números enteros aleatorios, usando como semilla del generador al valor 973 (es decir, random.seed(973))**. Los valores de cada uno de esos 14000 números deben estar **entre 100 y 21100 (incluidos ambos - DEBE usar random.randint(100, 21100) para generar cada uno de estos números)**.

A modo de elemento de control, y para que cada estudiante sepa si la secuencia que está generando es la correcta, indicamos aquí cuánto debe valer la suma de todos los números generados con el mecanismo indicado:

✓ Suma de todos los números generados: 149000017

A partir de esa sucesión, el programa debe:

1. Determinar **cuántos** eran menores o iguales que 11000, **cuántos** eran mayores que 11000 pero menores que 17000 y además eran divisibles por 3 y por 8 al mismo tiempo, y **cuántos** eran mayores o iguales que 17000.
2. Determinar el **promedio entero** de todos los números generados que **sean divisibles por 9 pero que sean también menores o iguales a 15000**. Aclaración: NO se pide el promedio redondeado, sino el promedio truncado, sin decimales.
3. Determinar el **mayor** entre todos los números generados **cuyo valor esté entre 1000 y 14000** (incluidos ambos).
4. Determinar el **porcentaje entero** que la cantidad de **números divisibles por 6** representa sobre la cantidad total de números. Aclaración: NO se pide el porcentaje redondeado, sino truncado, sin decimales. *Observación: en el cálculo de este porcentaje, haga primero la multiplicación que corresponda, y luego la división.*

Turno 03:

Desarrolle un programa completo en Python que permita generar una sucesión de **19000 números enteros aleatorios, usando como semilla del generador al valor 3374 (es decir, `random.seed(3374)`)**. Los valores de cada uno de esos 19000 números deben estar **entre -1000 y 15000 (incluidos ambos - DEBE usar `random.randint(-1000, 15000)` para generar cada uno de estos números)**.

A modo de elemento de control, y para que cada estudiante sepa si la secuencia que está generando es la correcta, indicamos aquí cuánto debe valer la suma de todos los números generados con el mecanismo indicado:

✓ Suma de todos los números generados: 133091344

A partir de esa sucesión, el programa debe:

1. Determinar **cuántos** eran negativos. Además, determinar **cuántos** eran mayores o iguales a 0 y menores que 12000 pero además eran divisibles por 5. Y finalmente determinar cuánto es **la suma** de los que eran mayores o iguales a 12000 pero además eran divisibles por 3.
2. Determinar el **promedio entero** de todos los números generados que estén **entre -200 y 3000 (incluidos ambos)**. Aclaración: NO se pide el promedio redondeado, sino el promedio truncado, sin decimales.
3. Determinar el **menor** entre todos los números generados **que no sean negativos y además sean divisibles por 9**.
4. Determinar el **porcentaje entero** que la cantidad de **números generados pares negativos representa sobre la cantidad total de números procesados**. Aclaración: NO se pide el porcentaje redondeado, sino truncado, sin decimales. *Observación: en el cálculo de este porcentaje, haga primero la multiplicación que corresponda, y luego la división.*

Turno 04:

Desarrolle un programa completo en Python que permita generar una sucesión de **25000 números enteros aleatorios, usando como semilla del generador al valor 7658 (es decir, `random.seed(7658)`)**. Los valores de cada uno de esos 25000 números deben estar **entre -2500 y 45000 (incluidos ambos - DEBE usar `random.randint(-2500, 45000)` para generar cada uno de estos números)**.

A modo de elemento de control, y para que cada estudiante sepa si la secuencia que está generando es la correcta, indicamos aquí cuánto debe valer la suma de todos los números generados con el mecanismo indicado:

✓ Suma de todos los números generados: 529572772

A partir de esa sucesión, el programa debe:

1. Determinar **cuántos** eran menores o iguales que -500; **cuántos** eran mayores que -500 y menores que 27000, y **cuántos** eran mayores o iguales que 27000 pero además eran divisibles por 10.
2. Determinar el **promedio entero** entre los números **mayores a 0 y divisibles por 7 o por 8**. Aclaración: NO se pide el promedio redondeado, sino truncado, sin decimales.
3. Determinar el **mayor** entre todos los números generados que sean **negativos divisibles por 4**.
4. Determinar el **porcentaje entero** que la cantidad de **números menores que 5000** representa sobre la cantidad total de números. Aclaración: NO se pide el porcentaje redondeado, sino truncado, sin decimales. *Observación: en el cálculo de este porcentaje, haga primero la multiplicación que corresponda, y luego la división.*

Turno 05:

Desarrolle un programa completo en Python que permita generar una sucesión de **30000 números enteros aleatorios**, usando como semilla del generador al **valor 2753** (es decir, **random.seed(2753)**). Los valores de cada uno de esos 30000 números deben estar **entre -15000 y 15000** (incluidos ambos - **DEBE usar random.randint(-15000, 15000) para generar cada uno de estos números**).

A modo de elemento de control, y para que cada estudiante sepa si la secuencia que está generando es la correcta, indicamos aquí cuánto debe valer la suma de todos los números generados con el mecanismo indicado:

✓ Suma de todos los números generados: 368162

A partir de esa sucesión, el programa debe:

1. Determinar **la cantidad** de números que eran negativos; también determinar **la suma** de todos los números que eran mayores o iguales a cero pero menores que 5000, y **la cantidad** de números eran mayores o iguales que 5000 pero además eran impares.
2. Determinar el **promedio entero** de todos los números generados que sean **negativos pero que además sean divisibles por 3 y por 5**. Aclaración: NO se pide el promedio redondeado, sino el promedio truncado, sin decimales.
3. Determinar el **menor** entre todos los números generados que sean **mayores a cero y divisibles por 3 pero no divisibles por 4**.
4. Determinar el **porcentaje entero** que la cantidad de **números negativos impares** representa sobre la cantidad total de números. Aclaración: NO se pide el porcentaje redondeado, sino el truncado, sin decimales. *Observación: en el cálculo de este porcentaje, haga primero la multiplicación que corresponda, y luego la división.*

Turno 06:

Desarrolle un programa completo en Python que permita generar una sucesión de **13000 números enteros aleatorios negativos**, usando como semilla del generador al **valor 1779** (es decir, **random.seed(1779)**). Los valores de cada uno de esos 13000 números deben estar **entre -25000 y -1000** (incluidos ambos - **DEBE usar random.randint(-25000, -1000) para generar cada uno de estos números**).

A modo de elemento de control, y para que cada estudiante sepa si la secuencia que está generando es la correcta, indicamos aquí cuánto debe valer la suma de todos los números generados con el mecanismo indicado:

✓ Suma de todos los números generados: -167972204

A partir de esa sucesión, el programa debe:

1. Determinar **la suma** de todos los números que eran pares; **la suma** de todos los que eran divisibles por 5, y **la cantidad** de números que eran mayores o iguales que -2000 pero además no eran divisibles por 4.
2. Determinar el **promedio entero** de todos los números generados que eran **mayores que -6000 pero menores que -2000 y que además no sean divisibles por 6**. Aclaración: NO se pide el promedio redondeado, sino el promedio truncado, sin decimales.
3. Determinar el **menor** entre todos los números generados que estén comprendidos **entre -20000 y -5000** (incluidos ambos) y que sean también divisibles por 8.



EXAMEN PARCIAL 1

Algoritmos y Estructuras de Datos

Fecha: 20/05/2023

-
4. Determinar el *porcentaje entero* que *la cantidad de números mayores que -3000 pero que sean divisibles por 3* representa sobre la cantidad total de números procesados. Aclaración: NO se pide el porcentaje redondeado, sino el truncado, sin decimales. *Observación: en el cálculo de este porcentaje, haga primero la multiplicación que corresponda, y luego la división.*