

[T1] TAREA 1

INSTRUCCIONES

Antes de comenzar, lea atentamente las instrucciones, los requerimientos de entrega y las preguntas de la evaluación.

- La evaluación es de carácter individual y está sujeta a las normas del código de honor de la asignatura. Cualquier indicio de intervención de otra persona o de cualquier otro acto sancionable como falta al código de honor significará la calificación mínima al promedio de Teoría.
- **La solución debe desarrollarse directa y completamente en Python y debe entregarse por UVirtual.**
- Estudiantes que entreguen fuera de plazo serán calificados con nota mínima en esta evaluación.
- Entregas que se realicen por vías distintas a Campus Virtual serán calificadas con nota mínima. En caso de que la plataforma presente un problema, puede enviar su archivo como respaldo al correo de contacto del profesor de Teoría, y posteriormente subir el archivo a Campus Virtual.
- Estudiantes que entreguen sin seguir los criterios de identificación estipulados en el apartado de “entrega” serán calificados con nota mínima en esta evaluación.
- Considere que cualquier supuesto que el estudiante haga debe ser explicitado en los comentarios de código.
- La subida del archivo es responsabilidad de su autor, por lo que archivos que no estén en el formato estipulado, que vengan corruptos o con problemas para ser leídos no serán revisados.
- **La evaluación (T1) se evaluará considerando 100% puntaje de casos de prueba.**

ENTREGA

Para su archivo de desarrollo cree un archivo .py con su RUN. Agregue al encabezado de su programa los siguientes datos de identificación. Considere que, de no agregarlos, el puntaje de su solución no contará.

FUNDAMENTOS DE PROGRAMACIÓN PARA INGENIERÍA
FUNDAMENTOS DE COMPUTACIÓN Y PROGRAMACIÓN
SECCIÓN DEL CURSO:
PROFESOR DE TEORÍA:
PROFESOR DE LABORATORIO:

AUTOR
NOMBRE:
RUN:
CARRERA:

Suba su solución al apartado especificado en UVirtual en la pestaña “EVALUACIONES”, apartado “[T1] ENTREGA”.

DE PARTITURAS Y TRANSCRIPCIONES

Cuando aprendemos de música, se nos enseña normalmente que existen siete notas musicales que representan sonidos que un instrumento (o un cantante) puede ejecutar. Estos, según la notación, pueden ser: Do, Re, Mi, Fa, Sol, La, Si, o en la notación anglosajona: C, D, E, F, G, A, B. Cada una de estas notas representa una frecuencia en específico con la que el instrumento o la cuerda vocal del cantante vibra. Por ejemplo, en una guitarra, la segunda cuerda más gruesa, al estar afinada en “afinación estándar”, debería producir un La (o A), lo que significa que la cuerda está oscilando con una frecuencia de 110 veces por segundo o a 110 Hz. Mientras más agudas las notas, más alta es la frecuencia.

Una de las características interesantes de la notación de siete notas es que, una vez que se alcanza la última nota, se vuelve a la primera, es decir, luego de un Si, viene un Do y la escala vuelve a comenzar, esta vez con una frecuencia más alta. Al ciclo completo entre una nota y la siguiente, se le conoce como una *octava* pues hay ocho notas en dicho intervalo. También, la frecuencia de la primera nota en el intervalo es siempre exactamente la mitad de la frecuencia de la última nota en el intervalo. Así, por ejemplo, en una escala que va de La de 110 Hz, al siguiente La, encontrará este en la frecuencia de 220 Hz.

Además de estas siete notas, es común encontrar entre ellas sonidos intermedios, Para denotarlos usamos sostenidos (#) y bemoles (b):

- Un sostenido (#) indica que una nota debe tocarse más alto, en particular un semitono más arriba. Por ejemplo, un La#, es una nota levemente más aguda que un La.
- Un bemol (b) indica que una nota debe tocar levemente más bajo, en particular un semitono más abajo. Por ejemplo, una Lab, es una nota levemente más grave que un La.

En el teclado esto puede verse de forma más clara (Figura 1), pues entre las teclas blancas, que representan las notas principales, hay teclas negras, estas corresponden a los sostenidos y bemoles.

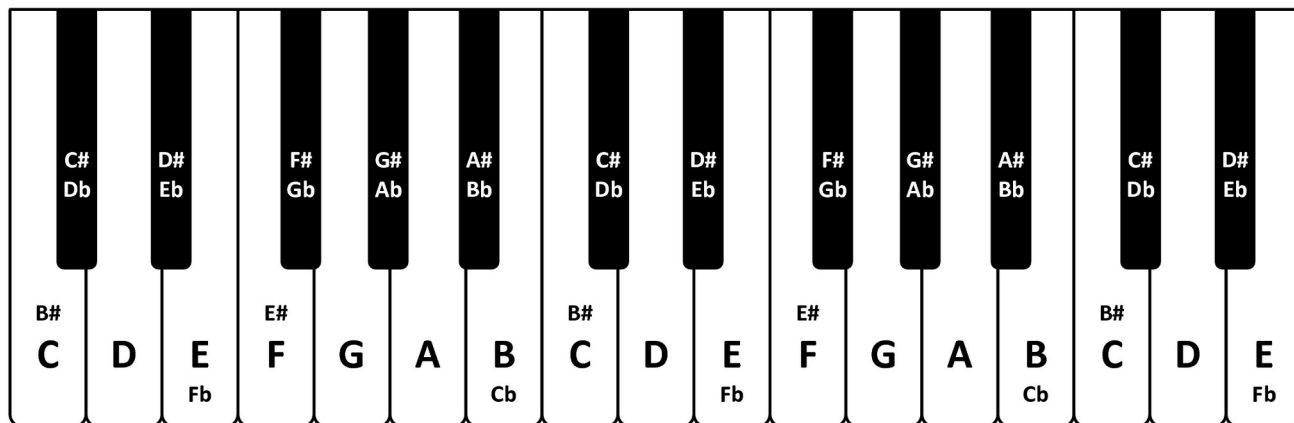


Figura 1 - Organización de teclas en el piano (Fuente: Yeshua Music School)

Como se puede ver, para efectos de la música basada en piano (que utiliza lo que se conoce como *temperamento igual*), algunas notas pueden tener distintos nombres (**enarmónicos**), dependiendo si miramos sostenidos o bemoles. Por ejemplo, un Mi (E) puede también entenderse como un Fa bemol (Fa \flat o F \flat) y un Do, puede entenderse como un Si sostenido (Si \sharp o B \sharp).

Con esto, a las 12 notas que conforman las teclas de una octava de piano se le denomina la **escala cromática**, que representa los 12 sonidos que se usan comúnmente para escribir música en la música occidental:

- Do o C/Si \sharp o B \sharp
- Do \sharp o C \sharp /Re \flat o D \flat
- Re o D
- Re \sharp o D \sharp /Mi \flat o E \flat
- Mi o E/Fa \flat o F \flat
- Fa o F/Mi \sharp o E \sharp
- Fa \sharp o F \sharp /Sol \flat o G \flat
- Sol o G
- Sol \sharp o G \sharp /La \flat o A \flat
- La o A
- La \sharp o A \sharp /Si \flat o B \flat
- Si o B/Do \flat o C \flat

Para poder lidiar con toda esta complejidad, los músicos han desarrollado un sistema ingenioso para escribir y leer música denominado pentagrama. En él, se dibujan cinco

líneas horizontales y en ellas se van representando las notas escribiéndolas ya sea sobre los espacios o directamente sobre las líneas (Figura 2).



Figura 2 - Notas en un pentagrama en clave de sol

Vale la pena destacar que, si una melodía escapa de una escala, se necesitarán líneas adicionales. Por ejemplo en el caso de la Figura 2, el primer Do está representado una línea más abajo del final de las 5 líneas del pentagrama, debido a que, al ir bajo el Re, que va justo debajo de la última línea, requeriría una línea adicional. Estas líneas solo se escriben en la nota que lo requiere y no en todo el pentagrama y pueden ir sobre o bajo las 5 líneas principales. La Figura 3, muestra cómo se anotarían distintas notas, con notación anglosajona, en el pentagrama.

Por otro lado, cuando en el pentagrama deben aparecer sostenidos o bemoles, estos se anotan antes de la nota para que el músico ejecutante sepa que se está usando dicha alteración (Figura 4). Como se puede ver, una nota en sostenido o en bemol ocupará la misma posición en el pentagrama que la nota original, sin embargo, tendrá anotado si corresponde a una alteración de bemol o sostenido antes de la nota.

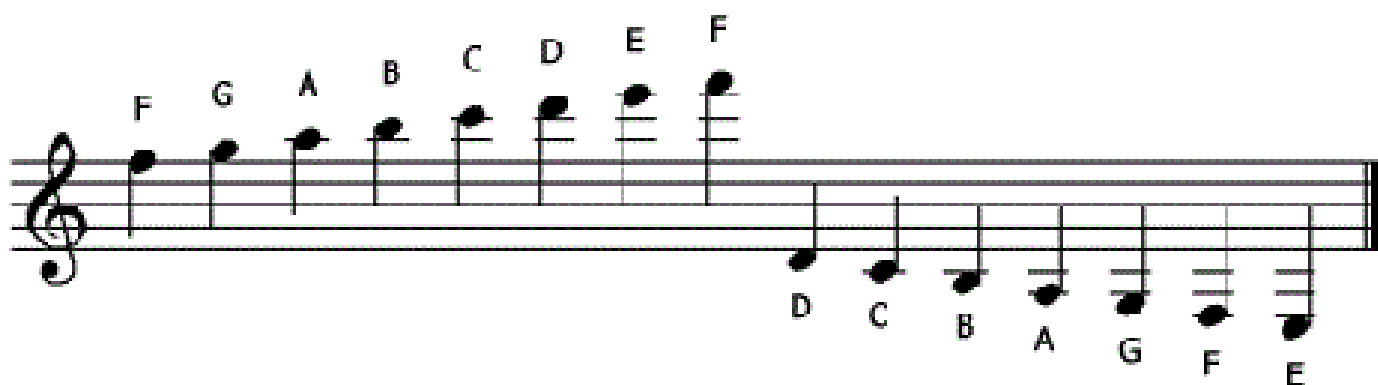


Figura 3 - Pentagrama en escala inglesa con líneas adicionales

Con esto en consideración, se nos pide construir un programa en Python que pueda generar pentagramas, tanto en notación latina (Do, Re, Mi, Fa, Sol, La, Si), como en notación anglosajona (C, D, E, F, G, A) según desee el usuario a partir de una melodía que será ingresada por teclado.

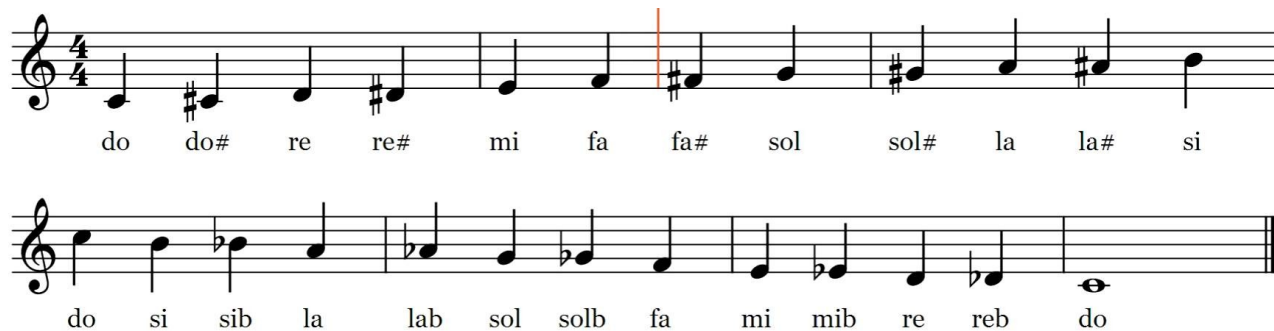


Figura 4 - Pentagrama con notas sostenidas y bemoles

El programa deberá validar que la melodía sea correcta, es decir:

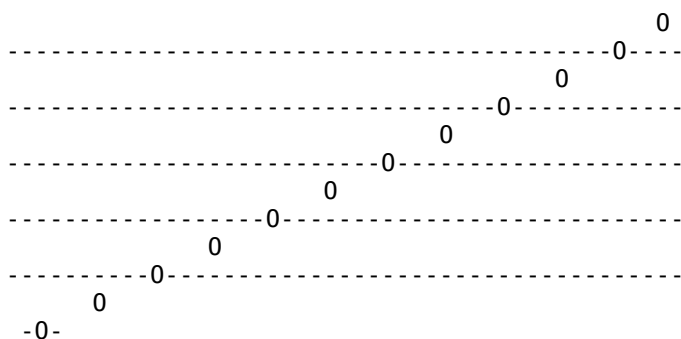
- Que use solo notas de la escala inglesa o de la escala española, sin mezclar.
- Que solo use bemoles o sostenidos.
- Que el total de notas siempre sea un múltiplo de 4. A fin de terminar en el tiempo correcto.

Las melodías serán ingresadas como un único string donde las notas irán separadas por espacio. Cada nota tendrá una nota y un número, en donde el primer elemento denotará la nota a ejecutar (pudiendo ser en notación anglosajona, latina, con o sin bemoles, y con o sin sostenidos), mientras que el número denotará en que punto de la escala se ubica la nota. Por ejemplo, para las notas de la Figura 2, la entrada sería:

Do4 Re4 Mi4 Fa4 Sol4 La4 Si4 Do5 Re5 Mi5 Fa5 Sol5

Considere como referencia que el Do que va en la primera línea bajo el pentagrama en clave de sol (la que usan los ejemplos) se denomina Do central (Do4 o C4) y que la numeración se desprende desde ahí.

A partir de esto, el programa deberá imprimir un string representando el pentagrama con las notas que componen la melodía. Para ello, las líneas del pentagrama se representarán con guiones medios (-) y las notas se representarán con caracteres O mayúscula (O), mientras que los sostenidos con el carácter numeral o gato (#) y los bemoles con un carácter b minúscula (b). Considere que cada nota dura lo mismo, es decir, no se considerará la representación de los tiempos de estas. Así el pentagrama de la Figura 2, debería verse como:



Nótese que entre cada nota existen tres caracteres. Por otro lado, si la entrada fuese con sostenidos, como:

Do#4 Re#4 Fa4 Fa#4 Sol#4 La#4 Do5 Do#5 Re#5 Fa5 Fa#5 Sol#5

La transcripción se vería así:

```

                                     #0
-----0--#0-----
                                     #0
-----0  #0-----
                                     #0
-----#0-----
0  #0
-----
#0
-#0-
```

En donde se han agregado los caracteres sostenidos donde correspondan. Finalmente, si la entrada fuese con bemoles, como:

Reb4 Mib4 Fa4 Solb4 Lab4 Sib4 Do5 Reb5 Mib5 Fa5 Solb5 Lab5

La transcripción se vería como:

```

                                     -b0-
                                     b0
-----0-----
                                     b0
-----b0-----
0
-----b0-----
b0
-----b0-----
0
-----b0-----
b0
```

Nótese que, en el pentagrama, las alteraciones (sostenidos y bemoles) van **antes** de la nota, no después, como cuando escribimos el nombre de las notas.

ENTRADA

La entrada solicitada corresponde al string con las notas que conforman la melodía, el que debe solicitarse con el mensaje:

Ingrese la melodía: <string>

Considere que si se produce un error en alguna de las entradas, el programa deberá informar al usuario con el mensaje '**Error en la entrada**' y deberá solicitar nuevamente la entrada incorrecta. Tenga en cuenta que un usuario puede cometer más de una vez un error en una entrada y el programa no debería avanzar hasta recibir una entrada correcta.

Considere que, para simplificar, las notas en la entrada siempre que deben ser consideradas válidas son desde el do central (Do4) hasta el la sobre el pentagrama (La5), y todas las notas pueden ser alteradas con bemoles o sostenidos.

SALIDA

La salida consiste en imprimir el pentagrama que represente la melodía con las opciones del usuario. Tenga en cuenta que el largo de las líneas y la cantidad de líneas adicionales varía según la cantidad de notas y el rango entre el cuál se mueven las notas, pero el pentagrama siempre tiene cinco líneas (de ahí que sea *pentagrama*) y cuatro espacios, donde la línea más grave corresponde, en la clave de sol utilizada en este problema, a Mi4 y la línea más aguda, a Fa5. En otras palabras, si no hay notas bajo o sobre el pentagrama, no se deben mostrar líneas adicionales, pero este **siempre** incluye las cinco líneas mencionadas.

CONSIDERACIONES

- Es común encontrar melodías que tienen sostenidos y bemoles combinados (ver, por ejemplo, melodías construidas sobre el arpeggio de [Fa# con séptima disminuida](#), que contiene las notas Fa#, La, Do y Mi♭, existen razones en la teoría musical para utilizar esa notación y no los enarmónicos), sin embargo, las entradas para este problema contarán todas con solo sostenidos o solo bemoles.
- Considere que cada nota en la partitura tiene un ancho de cuatro caracteres, donde:
 - El primer carácter es un espacio o un guión, según si la nota está en una línea del pentagrama o en un espacio. Si la nota está bajo o encima del pentagrama y tiene una alteración (como Do#4, en la Figura 5), su primer carácter será un guión:

-#0-

- (Nótese que este ejemplo es para demostrar la nota, no es una entrada válida, al no tener una cantidad de notas que sea un múltiplo de cuatro).
- El segundo carácter corresponde a la alteración (si la tuviera) o lo que le corresponda acompañar a la nota entre línea y espacio.
 - El tercer carácter corresponde a la nota (0).
 - El cuarto carácter corresponde a lo que le corresponda acompañar a la nota entre línea y espacio.
 - Para su entrega, considere que no pueden importarse módulos y que, como todas las demás evaluaciones, está sometida al Código de Honor del curso.



Figura 5: Do#4,
primera línea
bajo el
pentagrama

Ejemplos

Para el ejemplo 1, su partitura se puede ver en la Figura 6

Entrada 1:

Ingresa la melodía: B4 E5 C#5 B4 B4 A4 F#4 G#4 A4 C#5 B4 E5 C#5 B4 B4 B4

Salida 1:

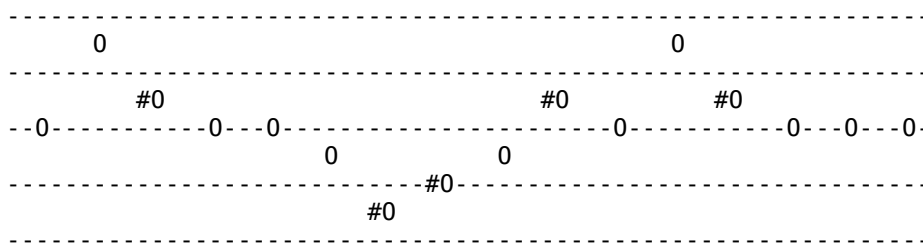


Figura 6: Melodía en el Ejemplo 1.

Entrada 2:

Salida 2:

- 0 -



Figura 7: Melodía del Ejemplo 2