

Laboratorio 1: Arquitectura y Organización de Computadores

Profesor: Mauricio Solar

Ayudante de cátedra: Mauricio Cortés y Muryel Constanzo

Ayudante de Tarea: Vicente Alvear y Luciano Yevenes

23 de Agosto 2024

1 Reglas Generales

Para esta tarea se utilizará python para generar e interpretar resultados de dados comunes en los juego de mesa y convertirlos a distintas bases numéricas. La tarea va a estar compuesta de dos etapas; La primera consiste de la generación de dados aleatorios y la conversión del resultado a bases numéricas distintas (decimal \rightarrow binario/octal/hexadecimal). La segunda consiste de "desafiar" al usuario del programa haciéndole ingresar la representación correcta de un número impreso en consola (binario/octal/hexadecimal \rightarrow decimal).

2 Calabozos y Dragones Conversión Numérica

ARMando y su primo RISCardo , después de haber jugado por cientos de horas al *Baldur's Gate 3* han decidido intentar hacer su propio motor de D&D para crear sus propias aventuras. Sin embargo, el computador que tienen en su casa es muy limitado y solo puede hacer calculos en bases numericas distintas a la decimal. Para eso le han encargado a usted, con su vasto conocimiento en la asignatura **INF245 ARQUITECTURA DE COMPUTADORES**, generar un programa que pueda sentar las bases para que los primos puedan construir sus ñoñas aventuras.

2.1 Lógica de dados

La primera modalidad del programa consiste en la generación de números aleatorios para simular el lanzamiento de dados tradicional en los juegos de mesa. En este modo, el usuario va a poder ingresar lanzamientos del tipo **{cantidad de dados}d{tipo de dado}**. Por ejemplo, si quisiera lanzar 2 dados de 6 caras, el texto ingresado por consola sería **2d6**. El programa debe a generar el resultado de ese lanzamiento con la librería aleatoria que encuentre pertinente (aunque randint basta) y va a imprimirlo en consola. Existe la siguiente salvedad: **el tipo de dado define la base numerica en que se imprime el número**. Se van a permitir 3 tipos de dados y el programa debera implementar la conversión pertinente para cada uno.

1. Los dados de 6 caras se deben imprimir en base binaria.
2. Los dados de 20 caras de deben imprimir en base octal.
3. Los dados de 100 caras se deben imprimir en base hexadecimal.

Por lo tanto, si tiro dos dados de 6 caras y mi resultado es 7 (que es lo más probable) , la pantalla debería imprimir **111**. Si tiro 1d100 y mi resultado es 65, la pantalla debe imprimir **41**, etc...

```
Dados obtenidos: 2d6  
Numero obtenido: 111  
Ingrese su respuesta:
```

Figure 1: Se muestra la salida del programa que simula el lanzamiento de dados para juegos de mesa. En este caso, el usuario lanzó dos dados de 6 caras (2d6). El resultado obtenido es 7, que se imprime en consola en su representación binaria como "111".

2.2 Lógica de desafío

La segunda parte del programa interactúa con el usuario, solicitando que responda a la pregunta generada por el resultado del lanzamiento de dados. El número obtenido, mostrado en su base numérica correspondiente, debe ser ingresado por el usuario, quien debe convertirlo a decimal. Si la respuesta es correcta, el programa indica que ha superado el desafío; de lo contrario, muestra un mensaje de fracaso.

```
Dados obtenidos: 2d6  
Numero obtenido: 111  
Ingrese su respuesta: 7  
  
Desafio superado
```

Figure 2: En este caso, se lanzaron dos dados de 6 caras (2d6) y el número obtenido es "111". Se ingresa "7", que es la conversión correcta del número binario. Como resultado, el programa muestra el mensaje "Desafío superado", indicando que la respuesta del usuario fue correcta.

```
Dados obtenidos: 1d100  
Numero obtenido: 41  
Ingrese su respuesta: 72  
  
Desafio Errado
```

Figure 3: En este caso, se lanzó un dado de 100 caras (1d100) y el número obtenido es "41". Se ingresa "72", que es la conversión incorrecta del número hexadecimal. Como resultado, el programa muestra el mensaje "Desafío errado", indicando que la respuesta del usuario fue incorrecta.

3 README

Debe contener como mínimo:

- Nombre, Rol y Paralelo de los integrantes.
- Especificación de los algoritmos y desarrollo realizado.
- Supuestos utilizados

4 Consideraciones

- Se deberá trabajar de a pares.
- Se deberá entregar en Aula a mas tardar el día 7 de septiembre de 2024 hasta las 23:59 horas. Se descontarán 5 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de las tareas.
- La tarea debe realizarse usando el lenguaje python. Se asume que ha tenido vivencias con él, o que aprende con rapidez.
- Pueden crear todas las funciones auxiliares que deseen, siempre y cuando estén debidamente comentadas.
- La entrega considera un único archivo de nombre **t1.py** junto con el README. Los archivos deberán ser comprimidos y enviados en un archivo .tar.gz o en .zip (esto queda a su conveniencia) en el formato **LAB1_ROL1_ROL2**.
- Si no se entrega README, o si su programa no funciona, la nota es 0 hasta la corrección.
- Una vez entregadas las notas de la tarea existirá un plazo de 5 días para apelar. Transcurrido este plazo las notas no podrán ser modificadas.
- Usted debe implementar la conversión en ambos sentidos. **ESTA EXPLICITAMENTE PROHIBIDO CUALQUIER LIBRERÍA O FUNCIÓN QUE DERROTE EL PROPÓSITO DE LA TAREA (ES DECIR, QUE HAGAN LAS CONVERSIONES DE BASE), SI NO SE CUMPLE ESTO TENDRÁ NOTA 0**