

Guessing Game

Aquest treball se centra en els bucles `while` i els nombres aleatoris. Has de lliurar un fitxer anomenat `guessing_game.py`. El teu programa permetrà a l'usuari jugar a un joc en què el programa pensa en un enter aleatori i accepta intents de l'usuari fins que aquest encerti el número correcte. Després de cada intent incorrecte, hauràs d'indicar a l'usuari si la resposta correcta és més alta o més baixa. El teu programa ha de reproduir exactament el format i el comportament dels registres que apareixen en aquest document. El registre a continuació mostra una execució d'exemple del teu programa. La teva sortida variarà en funció dels números aleatoris triats i de les entrades de l'usuari, però l'estructura general de la sortida ha de coincidir amb la mostrada a continuació.

```
<< intro message here >>
```

```
I'm thinking of a number between 1 and 100...
Your guess? 50
It's lower.
Your guess? 25
It's higher.
Your guess? 35
It's lower.
Your guess? 30
It's higher.
Your guess? 32
It's lower.
Your guess? 31
You got it right in 6 guesses!
Do you want to play again? y

I'm thinking of a number between 1 and 100...
Your guess? 50
It's higher.
Your guess? 75
It's lower.
Your guess? 65
It's lower.
Your guess? 64
You got it right in 4 guesses!
Do you want to play again? YES

I'm thinking of a number between 1 and 100...
Your guess? 60
It's lower.
Your guess? 20
It's higher.
Your guess? 30
It's higher.
Your guess? 40
It's higher.
Your guess? 50
It's lower.
Your guess? 47
It's higher.
Your guess? 49
You got it right in 7 guesses!
Do you want to play again? no

Overall results:
Total games      = 3
Total guesses    = 17
Guesses/game     = 5.7
Best game        = 4
```

Primer, el programa imprimeix un missatge d'introducció. A continuació, es juga a una sèrie de jocs d'endevinar. En cada joc, l'ordinador tria un número aleatori entre 1 i 100, ambdós inclosos. El joc demana a l'usuari que faci intents fins que encerti el número correcte. Després de cada intent incorrecte, el programa dona una pista indicant si el número correcte és més alt o més baix que l'intent fet. Quan l'usuari escriu el número correcte, el joc acaba i el programa informa de quants intents han estat necessaris.

Després que cada joc acabi i es mostri el nombre d'intents, el programa pregunta a l'usuari si vol jugar de nou. Suposa que l'usuari escriurà una resposta d'una sola paraula a aquesta pregunta.

Un nou joc hauria de començar si la resposta de l'usuari comença amb una **Y** o **y**, ja sigui en majúscula o minúscula. Per exemple, respostes com ara "y", "Y", "yes", "YES" o "Yes" indiquen que l'usuari vol jugar de nou.

Qualsevol altra resposta significa que l'usuari no vol jugar de nou. Per exemple, respostes com "no", "No", "okay", "0", "certainly" o "hello" s'interpretaran com un **no**.

Quan l'usuari decideix no jugar més, el programa imprimeix estadístiques generals de tots els jocs jugats. Aquestes estadístiques inclouen:

1. El nombre total de jocs jugats.
2. El nombre total d'intents fets en tots els jocs.
3. El nombre mitjà d'intents per joc (com a número real arrodonit a la dècima més propera).
4. El millor joc (el nombre mínim d'intents neces-

saris per encertar en un dels jocs).

Les teves estadístiques han de ser correctes per a qualsevol nombre de jocs o intents ≥ 1 . Pots suposar que cap joc requerirà un milió o més intents.

Has de gestionar el cas especial en què l'usuari encerta el número correcte al primer intent. Imprimeix un missatge com el següent:

```
I'm thinking of a number between 1 and 100...
Your guess? 71
You got it right in 1 guess!
```

Suposa que les entrades de l'usuari sempre seran vàlides. Quan se li demani un número, l'usuari només introduirà nombres enters, i aquests estaran dins dels intervals correctes.

Implementation Guidelines:

<< your haiku intro message here >>

```
I'm thinking of a number between 1 and 5...
Your guess? 2
It's higher.
Your guess? 4
It's lower.
Your guess? 3
You got it right in 3 guesses!
Do you want to play again? yes
```

```
I'm thinking of a number between 1 and 5...
Your guess? 3
It's higher.
Your guess? 5
You got it right in 2 guesses!
Do you want to play again? Nah
```

```
Overall results:
Total games      = 2
Total guesses    = 5
Guesses/game     = 2.5
Best game        = 2
```

Defineix una constant per al número màxim utilitzat als jocs. Tot i que el registre de la pàgina anterior mostra jocs de l'1 al 100, el valor de la constant s'hauria de poder canviar per utilitzar altres intervals, com de l'1 al 50 o qualsevol màxim desitjat.

Utilitza la teva constant arreu del codi i no facis referència al número 100 directament. Prova el teu programa canviant la constant i executant-lo de nou per assegurar-te que tot fa servir el nou valor. Un joc d'endevinar números de l'1 al 5 produiria una sortida com la mostrada a l'esquerra.

Genera aleatorietat utilitzant `randint`, tal com es va discutir a classe.

Mostra nombres arrodonits utilitzant la funció `round` de Python.

Llegeix les respostes d'usuari de tipus sí/no utilitzant `input`. Per comprovar si la resposta de l'usuari representa un sí o un no, utilitza les funcions de cadena que es van discutir a la classe.

Genera repetició utilitzant bucles `while`.

```
I'm thinking of a number between 1 and 100...
*** HINT: The answer is 46
Your guess? 50
It's lower.
Your guess? 25
It's higher.
Your guess? 48
It's lower.
Your guess? 46
You got it right in 4 guesses!
```

(suggested initial simple version of program)

per provar el cas de "1 intent".

Recomano que comenceu escrivint una versió més senzilla que jugui un sol joc d'endevinar. Ignoreu altres característiques com diversos jocs i la visualització d'estadístiques generals. Mentre depureu el programa, és útil imprimir un missatge temporal de "pista" com el que es mostra a l'esquerra. D'aquesta manera sabreu la resposta correcta i podreu provar si el programa dóna pistes adequades per a cada intent. Això també és útil

Style Guidelines:

Estructura la teva solució utilitzant funcions que acceptin paràmetres i retornin valors quan sigui apropiat. Has de tenir com a mínim les següents funcions, a part de la funció **main** en el teu programa:

1. Una funció per jugar un sol joc amb l'usuari. Aquesta funció no hauria de contenir codi per preguntar a l'usuari si vol jugar de nou. Tampoc hauria de jugar diversos jocs en una sola crida.
2. Una funció per informar les estadístiques generals a l'usuari. Aquesta funció només hauria d'imprimir les estadístiques, sense fer res més com bucles `while` o jugar jocs.

Pots definir més funcions si ho desitges. Està bé que algunes instruccions `print` estiguin a la funció **main**, sempre que utilitzis una bona estructura i **main** sigui un resum concís. Per exemple, pots posar el bucle per als jocs múltiples i la pregunta per jugar de nou a **main**.

Fes servir correctament els espais en blanc i la identació. Dona noms significatius a les funcions/variables i segueix els estàndards de nomenclatura de Python. Fes servir comentaris descriptius al principi del teu programa i de cada funció. Ja que aquest programa té funcions més llargues, també posa comentaris breus dins de les funcions per a seccions de codi complexes.