1. feladat (alap): Szám Kitalálós Játék

Készíts egy számkitalálós játékot, amely:

A) Játék mechanika:

- A számítógép „gondol” egy véletlen számra 1 és 100 között
- A felhasználó tippeli a számot
- A program jelzi, hogy a tipp nagyobb, kisebb vagy helyes
- Számolja a próbálkozások számát

B) While ciklus használata:

- A játék addig folytatódjon, amíg el nem találják a számot
- Minden hibás tipp után újra bekéri a következő tippet
- Helyes tipp esetén gratuláljon és írja ki a próbálkozások számát

C) Extra funkciók:

- Ha 10-nél több próbálkozás van, segítő üzenet: „Gondolj logikusan!”
- Érvénytelen bemenet (nem szám, vagy tartományon kívül) esetén ne számítson bele a próbálkozásokba

Tipp: Használd a `random.randint(1, 100)` függvényt!

2. feladat (matematika): Prímszám Vizsgáló és Generátor

Írj programot, amely egy bekért pozitív egész számról eldönti, hogy prím-e:

A) Prímvizsgálat:

- Bekér egy pozitív egész számot
- While ciklussal vizsgálja, hogy van-e osztója 2-től $\sqrt{n}$-ig
- Ha nincs osztója: „X prímszám”
- Ha van osztója: „X összetett szám, osztható Y-nal”

B) Prím generálás:

- Kérje be, hogy hány prímszámot szeretne a felhasználó
- While ciklussal keresse meg az első N prímszámot
- Minden prímszámot írjon ki, amikor megtalálja

C) Statisztika:

- Számolja, hogy hány számot vizsgált meg összesen
- Írja ki a hatékonyságot: „N prím megtalálásához M számot vizsgáltam”

Segítség: A `math.sqrt()` vagy `x**0.5` használható gyökvonásra.

3. feladat (szimuláció): Pénzfeldobás Szimulátor

Szimuláljunk pénzfeldobás sorozatokat és elemezzük az eredményeket:

A) Egyszerű szimuláció:

- Kérje be, hány feldobást szeretne szimulálni (pl. 1000)
- While ciklussal dobjon N-szer (0=fej, 1=írás vagy „F”/„I”)
- Számolja a fejek és írások számát
- Számítsa ki a fej valószínűségét: $P = \frac{\text{fejek száma}}{\text{összes dobás}}$

B) Sorozat elemzés:

- Keresse meg a leghosszabb fej sorozatot
- Keresse meg a leghosszabb írás sorozatot
- Számolja, hányszor váltott a sorozat (FI vagy IF)

C) Elméleti összehasonlítás:

- Hasonlítsa össze a szimulált valószínűséget az elméleti 0.5-tel
- Írja ki a különbséget százalékban
- Ha 1000+ dobás és az eltérés $< 5\%$: „Jó szimuláció”

Használj `random.randint(0,1)` vagy `random.choice([,,F'',,,I''])` függvényt!

4. feladat (fizika): Szabadesés Szimuláció

Szimulálj egy szabadesés folyamatát lépésenként:

A) Fizikai paraméterek:

- Kezdő magasság: bekért érték (pl. 100 m)
- Kezdősebesség: 0 m/s (ejtés)
- Gravitációs gyorsulás: $g = 9.81$ m/s^2
- Időlépés: $\Delta t = 0.1$ s

B) Szimuláció while ciklussal:

- Amíg a magasság > 0, számítsa ki minden lépésben:
- Új sebesség: $v_j = v_{rgi} + g \cdot \Delta t$
- Új pozíció: $h_j = h_{rgi} - v_{rgi} \cdot \Delta t$
- Eltelt idő növelése: $t = t + \Delta t$

C) Eredmények és validáció:

- Írja ki a végső sebességet és esési időt
- Hasonlítsa össze az analitikus megoldással:
- $t_{elmleti} = \sqrt{\frac{2h}{g}}$, $v_{elmleti} = g \cdot t_{elmleti}$
- Minden 1 másodpercben írja ki: „t=X.Xs, h=Y.Ym, v=Z.Zm/s”

Extra: Légellenállással is próbáld ($F_{lgellenlls} = k \cdot v^2$, ahol $k = 0.1$)!

5. feladat (komplex): Jelszó Generátor és Erősség Ellenőrző

Készíts egy összetett jelszó generátor és ellenőrző programot:

A) Jelszó generálás:

- Kérje be a kívánt jelszó hosszt (8-50 karakter)
- Válas'tható kategóriák: kisbetű, nagybetű, számok, különleges karakterek
- While ciklussal generáljon véletlenszerűen karaktereket
- Biztosítsa, hogy minden választott kategóriából legyen legalább 1 karakter

B) Jelszó erősség ellenőrzés:

- Pontrendszer: kisbetű(+1), nagybetű(+2), szám(+2), különleges(+3)
- Hosszúság bónusz: 8+ kar.(+5), 12+ kar.(+10), 16+ kar.(+15)
- Ismétlés büntetés: ugyanaz a kar. 2x (-2), 3x+ (-5)
- Gyakori minták büntetése: „123", „abc", „qwe" (-10)

C) Értékelés és újragenerálás:

- 0-20 pont: „Gyenge", 21-40: „Közepes", 41-60: „Erős", 60+: „Kiváló"
- Ha a jelszó gyenge, kérdezze meg, hogy generáljon-e újat
- While ciklussal addig generáljon, amíg legalább „Közepes" nem lesz

Karakterkészletek:

```
kisbetű = ,,abcdefghijklmnopqrstuvwxyz''
nagybetű = ,,ABCDEFGHIJKLMNOPQRSTUVWXYZ''
számok = ,,0123456789''
különleges = ,,!@#$%^&*()-_=+[]|;:,.<>?''
```

Programozási tippek:

1. Karakterkészletek kezelése:

- Használj string változókat a karakterkészletekhez
- `osszes_karakter = kisbetu + nagybetu + szamok + kulonleges`
- `random.choice(string)` - véletlen karakter választás stringből
- `karakter in string` - ellenőrzi, hogy a karakter benne van-e

2. Jelszó generálás logika:

- Először generálj egy teljesen véletlen jelszót
- Ezután ellenőrizd, hogy minden kategóriából van-e karakter
- Ha hiányzik valamelyik: `continue` és újra generálás
- Vagy: előre beleszel 1-1 karaktert minden kategóriából, aztán keverd össze

3. Erősség ellenőrzés implementálása:

- `pontszam = 0` - kezdő érték
- For ciklus helyett while: `i = 0`, `while i < len(jelszo):`
- Karakterek számlálása: `kisbetu_db = 0`, majd minden kisbetűnél `kisbetu_db += 1`
- Ismétlések: `jelszo.count(karakter)` - hányszor fordul elő