



Es tracta d'un joc de rol on el jugador crea el seu personatge i s'endinsa en un món d'aventures. Per implementar aquest joc, pots seguir els passos següents:

- a) Partim d'una **llista de personatges** que contenen la informació del personatge creat pel jugador (nom, raça, classe, nivell, estadístiques, habilitats, etc.).

```
personajes = [  
  {"nombre": "Gandalf",  
   "raza": "humano",  
   "clase": "guerrero",  
   "habilidades": {  
     "fuerza": 10,  
     "destreza": 5,  
     "inteligencia": 2,  
     "carisma": 3  
   },  
   "inventario": ["espada", "escudo", "pociones"]  
},  
  {  
    "nombre": "Aragorn",  
    "raza": "ogro",  
    "clase": "guerrero",  
    "habilidades": {  
      "fuerza": 80,  
      "destreza": 3,  
      "inteligencia": 1,  
      "carisma": 1  
    },  
    "inventario": ["garrote"]  
  },  
  {  
    "nombre": "Legolas",  
    "raza": "humano",  
    "clase": "mago",  
    "habilidades": {  
      "fuerza": 200,  
      "destreza": 4,  
      "inteligencia": 10,  
      "carisma": 5  
    },  
    "inventario": ["varita magica", "libro de hechizos"]  
  },  
  {  
    "nombre": "Frodo",  
    "raza": "ogro",  
    "clase": "mago",  
    "habilidades": {  
      "fuerza": 12,  
      "destreza": 6,  
      "inteligencia": 20,  
      "carisma": 1  
    },  
    "inventario": ["varita magica", "libro de hechizos"]  
  },  
  {  
    "nombre": "Gimli",  
    "raza": "ogro",  
    "clase": "guerrero",  
    "habilidades": {  
      "fuerza": 3,  
      "destreza": 4,  
      "inteligencia": 9,  
      "carisma": 8  
    },  
    "inventario": ["libro de hechizos"]  
  }  
]
```



- b) Partim d'un diccionari monstres que contingui informació sobre els monstres que el jugador pot trobar al joc (la clau és el nom i els valors són una llista que representa els valors en el següent ordre: nivell, vida, atac, defensa i habilitats).

```
monstres = {  
    'goblin': [1,10,5, 5,['ataque normal', 'vuelo']],  
    'rata': [2, 20,10, 10, ['ataque normal', 'invocar esqueletos']],  
    'dragon': [5,50, 20,20, ['ataque normal', 'aliento de fuego', 'vuelo']],  
    'troll': [7,77,30,10, ['aliento de fuego']]  
}
```

Per exemple, els valors: `'goblin': [1,10,5, 5,['ataque normal', 'vuelo']]`,

Representen que el monstre **goblin** té nivell= 1, vida= 10, atac= 5, defensa=5 i les habilitats= "ataque normal" i "vuelo"

Altres variables:

```
nomColumnes=("nombre", "raza", "clase", "fuerza", "destreza", "inteligencia", "carisma", "inventari")  
relacio_personajes_monstres = [  
    ("Gandalf", "dragon", 'vuelo'),  
    ("Aragorn", "rata", 'invocar esqueletos'),  
    ("Legolas", "troll", 'aliento de fuego'),  
    ("Frodo", "goblin", 'vuelo'),  
    ("Gimli", "rata", 'invocar esqueletos'),  
    ("Gandalf", "rata", 'ataque normal'),  
    ("Legolas", "rata", 'invocar esqueletos'),  
]
```

La llista de tuples **relacio_personajes_monstres**, indica a quin monstre s'ha enfrontat cada personatge i amb quin atac amb guanyat, per exemple: Gandalf s'ha enfrontat a un dragó utilitzant el "vuelo" com habilitat, Aragorn a una rata amb l'habilitat "invocar esqueletos", etc.

Funcionalitats a implementar:

- a) Crea una funció **crea_monstre** que sol·liciti a l'usuari la informació necessària per a crear el seu propi monstre i l'afegeixi al diccionari **monstres**. (0.5 punts)

Forma de la funció: **crea_monstre(monstres)** a on **monstres** és un diccionari com el que tenim a d'alt. Aquesta funció retorna el diccionari **monstres** amb el monstre afegit.

Per a fer-ho hauràs de tenir en compte que:

- Crear funció **checkTipusMonstre()**: El tipus de monstre no es pot repetir i ha de ser un valor format només per lletres, per tant, quan demanem la clau del diccionari a l'usuari hem de comprovar mitjançant una estructura try...except que aquesta clau no existeix i que està formada només per lletres. Per exemple:
 - Si li preguntem el tipus de monstre i l'usuari escriu **goblin**, el programa ens ha d'informar que ja existeix i ens demanarà un altre cop el tipus de monstre.
 - Si li preguntem el tipus de monstre i l'usuari escriu **goblin33**, el programa ens ha



- d'informar que el tipus de monstre ha d'estar format només per lletres.
- Si li preguntem el tipus de monstre i l'usuari escriu *serp*, la funció retorna aquest valor ja que és correcte: només té lletres i no existeix coma a key al diccionari monstres.

La funció retorna una string amb el valor del tipus de monstre quan el valor introduït per l'usuari és correcte

- Per a les dades numèriques **nivell**, **vida**, **atac** i **defensa** del diccionari, s'haurà de crear una funció anomenada **checkNum(valor)** que comprovi que el que l'usuari està introduint pel teclat és un número (utilitzant try...except) i que esta en el rang de valors indicat:
Nivell: De l'1 al 10
Vida: De 0 a 100
Atac: De l'1 al 10
Defensa: De 0 a 100

La funció **checkNum(valor)** retorna el valor correcte de **nivell**, **vida**, **atac** o **defensa**. D'aquesta funció no sortirem fins que data tingui un valor int dintre del rang corresponent. Exemple:

checkNum('vida') retorna un valor numèric entre 1 i 10.
checkNum('nivell') retorna un valor numèric entre 0 i 100.
checkNum('atac') retorna un valor numèric entre 15 i 150.
checkNum('defensa') retorna un valor numèric entre 20 i 250.

La funció **crea_monstre** utilitza les dos anteriors funcions per validar les dades i insereix el personatge al diccionari **monstres**.

- b) Crea la funció **mostrarDades(personajes)**: Aquesta funció rep la llista **personajes** i mostra totes les dades per pantalla en format taula. Cada columna tindrà una alineació concreta de manera que garantitzi la correcta visualització de les dades. Per a mostrar l'encapçalament de la taula farem servir la tupla: **nomColumnes**. (0.25 punts)
- c) Crea la funció **inventariOrder(personajes)**. Aquesta funció rep la llista **personajes** i mostra per pantalla les dades dels personatges ordenats de manera decreixent pel primer valor de la llista d'inventari **utilitzant el mètode de la bombolla**.
La funció mostrarà totes les dades del diccionari. (1 punt)
- d) Crea la funció **personatgesOrder(personajes)** que rep una llista de **personajes** com el de d'alt i mostra per pantalla tots els personatges que s'han enfrontat a una rata amb habilitat "*ataque normal*" en ordre decreixent segons el valor de la seva **classe**, utilitzant la funció **lambda**. S'ha de mostrar tota la informació de cada monstre. (1 punt)



e) **RECURSIVITAT:** El monstre dragó s'ha ficat en un laberint sense voler. Aquest laberint té 10 sortides identificades amb números, la funció demana a l'usuari un número del 1 al 10, si tria un número senar que no sigui el 3, després de 10 minuts torna a l'inici del laberint, si tria un número parell que no sigui el 10 després de 2 minuts torna a l'inici del laberint, en el cas d'escollir el 3 després de 7 minuts troba la sortida del laberint i si escull el número 10 automàticament surt del laberint. Crea una **funció recursiva** que ens digui quants minuts trigarà el dragó a sortir del laberint. **(0.5 punts)**

No es poden utilitzar variables globals, ni variables definides fora de la funció.

El prototipus de la funció serà: **def drago()**

f) **RECURSIVITAT: (1.75 punts)**

Escriu una funció en Python que resolgui el joc següent: se't dona una llista de nombres enters i un número objectiu. L'objectiu és obtenir el número 0 a partir de la llista donada. La funció ha de prendre dos arguments: la llista de números i el número objectiu.

A cada pas del joc, el jugador pot realitzar una de dues operacions: sumar el número a la posició actual de la llista o restar-lo. Si el resultat de la suma o la resta està fora dels límits de la llista o resulta en un nombre negatiu, aleshores l'operació no es fa.

L'objectiu de l'exercici és escriure una funció a Python que, atesa una llista de nombres enters i el número objectiu, trobeu una seqüència d'operacions (suma o resta) que porti a la llista al número objectiu (en aquest cas, zero). La funció ha de tornar una llista de tuples que representin cada operació realitzada per assolir l'objectiu.

Per exemple, si s'anomena la funció amb la llista [3, 2, 4, 1, 7] i el número objectiu 2, la funció ha de retornar la llista de tuples [(3, -1, -4, 4), (3, -1, 1, -3), (3, -1, -2, 2)]. Cada tupla representa una seqüència d'operacions per arribar a l'objectiu. En aquest cas, la primera tupla indica que per assolir l'objectiu de 2, el jugador ha de començar sumant 3, restant 1, restant 4 i després sumant 4.



NOTA:

Utilitza totes les funcions, per comprovar el seu correcte funcionament.

NOTA IMPORTANT:

1. Un apartat obtindrà la puntuació indicada si fa exactament el que marca l'enunciat, en cas contrari, la puntuació de l'apartat serà de 0.
2. NO ES PODEN CANVIAR ELS PROTOTIPUS DE LES FUNCIONS, es a dir, LES FUNCIONS HAN DE SER EXACTAMENT IGUAL A LES INDICADES EN EL ENUNCIAT. Si es canvia, sense el vist i plau del/de la professor/a, la puntuació de la funció serà un 0.
3. Si a l'alumne li sona el mòbil durant la realització de l'examen, la qualificació de l'examen serà un 0.
4. Si l'alumne es connecta a una pàgina web, servidor o similar, i/o fa ús d'internet durant l'examen, la qualificació de l'examen serà un 0.