

# Nagy Beadandó

A feladatokban kért algebrai adatszerkezetekhez hozzá kell rendelni az `Eq` és `Show` típusosztályokat, amennyiben a leírás nem kér mást. A megfelelő tesztelés érdekében kérlek, az alábbi függvényeket másold be a kódodba:

```
showState a = show a
showMage a = show a
eqMage a b = a == b
showUnit a = show a
showOneVOne a = show a
```

## A végső megmértetés

## Típus magyarázó

A teljes olvasási élmény érdekében pár alap típust átneveztünk. Kérlek, ezeket vegyétek figyelembe! (Másoljátok ezeket a típuszinonímákat a forrásfájlba.)

```
type Name = String
type Health = Integer
type Spell = (Integer -> Integer)
type Army = [Unit]
type EnemyArmy = Army
type Amount = Integer
```

A hónapokon át tartó felkészülésed ehhez a ponthoz vezetett. Minden, amit eddig tanultál, most hasznosítanod kell. A Haskell mágus csapatai már felsorakoztak, de neked is készülnöd kell. Ez már nem gyakorlás. A téren izzik már a levegő és láthatóan feszült a hangulat. Itt most minden el fog dőlni pár napon belül.

## Első feladat (Felkészülés)

Első körben az egységeid állapotát számon kell tartanod. Alkosd meg a `State` állapotjelző adatszerkezetet. A `State`-nek legyen egy típusparamétere, hiszen több különböző harcegységeink kezelésére fel kell készülni. Az `Alive` konstruktorral jelöljük meg, amennyiben az egység még életben van, ekkor tudni szeretnénk, hogy milyen egység van életben. Illetve a `Dead` konstruktor az egység sajnálatos elesését jelöli, ilyenkor nem érdekes, hogy ki esett el.

```
data State = ...
```

Jelenítsük meg az információkat (tehát készítsünk saját `Show` példányt a `State`-re) úgy, hogy amennyiben az egység még él, az `Alive` jelölés ne jelenjen meg, minden más hozzátartozó információ igen. Ellenkező esetben, ha halott, akkor a `"Dead"` felirat jelenjen meg.

```
showState (Alive 5) == "5"
showState (Alive True) == "True"
showState Dead == "Dead"
showState (Alive 'a') == "'a'"
```

**Remek!** Ez segíteni fog figyelemmel követni a csapataid állapotát.

---

Most, hogy ez megvan, meg kell idézned csapataidat, akik az oldaladon harcolnak majd. Ezek lesznek a **Golem**-ek (konstruktor). Sietned kell, mert a Haskell mágus is megidézte már saját vérszomjas **HaskellElemental**-jait (konstruktor). Ezek a lények mind tartalmazzák az életerejüket egy **Health** paraméterben.

```
data Entity = ...
```

---

- Ugye még nem maradtunk le a mókázásról?

Hallod mestered hangját a háttérből. Egy tucat mágussal maga mögött érkezett. Minden erősítésre szükség lesz.

A **Mage**-ek mind a képző **Master**-ei (konstruktor). Egy párat felismersz, amíg tartott a képzésed. Mindegyik rendelkezik egy **Name**-mel, **Health**-tel és egy **Spell**-lel, amit a csatában használni tud majd.

```
data Mage = ...
```

A mágusok, akik eljöttek segíteni a harcban nem mások, mint:

```
papi = let
  tunderpor enemyHP
    | enemyHP < 8 = 0
    | even enemyHP = div (enemyHP * 3) 4
    | otherwise = enemyHP - 3
  in Master "Papi" 126 tunderpor
java = Master "Java" 100 (\x -> x - (mod x 9))
traktor = Master "Traktor" 20 (\x -> div (x + 10) ((mod x 4) + 1))
jani = Master "Jani" 100 (\x -> x - div x 4)
skver = Master "Skver" 100 (\x -> div (x+4) 2)
```

Az ő információikat is szükséges megjelentetni. A mestereknek elég csak a nevüket kiírni, viszont amint az életerejük 5 alá csökken, egy **Wounded** jelzést kapnak a nevük elé.

Továbbá őket egyszerűen meg lehet különböztetni (**Eq** példány) a nevük és életerejük alapján.

```
showMage papi == "Papi"
showMage java == "Java"
showMage (Master "Skver" 4 id) == "Wounded Skver"
showMage (Master "Papi" 4 id) == "Wounded Papi"

eqMage papi papi
not $ eqMage papi traktor
eqMage jani jani
not $ eqMage java jani
```

---

Most, hogy a Mestereket is sikerült megismerned, ideje összeállítanod a csapatokat. Először egységesíteni kell az embereidet. Innentől mindenkit **Unit**-ként kell kezelj, akik ha Mesterek, **M** konstruktorral kell jelölnöd, ha pedig **Entity**, akkor **E** konstruktorral.

```
data Unit = ...
```

Ezeket a konstruktorokat ne tüntesd fel az információjuk kiírása során!

```
showUnit (M (Alive papi)) == showMage papi
showUnit (E (Alive (Golem 5))) == show (Golem 5)
```

```
showUnit (E (Alive (HaskellElemental 5))) == show (HaskellElemental 5)
showUnit (E Dead) == "Dead"
showUnit (M Dead) == "Dead"
```

A harcosaid felsorakoztak. Az ellenség valamire készül ... De mire?

Ebben a pillanatban feltűnik valami az égen. Vakító lila fény árad szét a környéken és hatalmas robajlással becsapódik valami. A föld megremeg ettől és mire visszanyered látásod, a környék lángokban áll.

## Második feladat (Elesettek)

**Gyorsan!** Rendezned kell soraidat. Az elesetteket hátra kell vinni. Erre írd meg a `formationFix` függvényt, amely az elesett harcosokat egyesével sorrendben hátra viszi a seregben.

```
formationFix :: Army -> Army
```

Meg kell tisztítani a környéket!

```
formationFix [E Dead, E Dead, E (Alive (Golem 4)), M (Alive (Master "Papi" 126 (\x -> x - 1)))] == [E (Alive (Golem 4)), M (Alive (Master "Papi" 126 (\x -> x - 1))), E Dead, E Dead]
formationFix [E (Alive (Golem 4)), E Dead, M (Alive (Master "Papi" 126 (\x -> x - 1))), M Dead, E (Alive (Golem 10)), E (Alive (Golem 10)), M Dead] == [E (Alive (Golem 4)), M (Alive (Master "Papi" 126 (\x -> x - 1))), E (Alive (Golem 10)), E (Alive (Golem 10)), E Dead, M Dead, M Dead]
formationFix [M (Alive (Master "Java" 100 (\x -> x - (mod x 9)))), M Dead, E Dead, M (Alive (Master "Papi" 126 (\x -> x - 1))), M Dead, E (Alive (Golem 10)), E Dead, E Dead] == [M (Alive (Master "Java" 100 (\x -> x - (mod x 9)))), M (Alive (Master "Papi" 126 (\x -> x - 1))), E (Alive (Golem 10)), M Dead, E Dead, M Dead, E Dead, E Dead]
```

## Harmadik feladat (Vége?)

Taktikázni kell! Figyelembe kell vened, hogy meddig tudsz küzdeni. A csatának akkor van vége, ha a sereg minden tagja elesett. Alkosd meg az `over` függvényt, amivel nyomon tudod követni, hogy a sereg minden tagja elesett-e.

```
over :: Army -> Bool
```

Lássuk őket!

```
over []
over [E Dead]
not $ over [E (Alive (Golem 10)), E Dead]
not $ over [E (Alive (Golem 10)), M Dead, E (Alive (Golem 1))]
not $ over [E Dead, M Dead, E (Alive (Golem 1))]
over [E Dead, E Dead, E Dead, E Dead, E Dead, E Dead, E Dead, E Dead, E Dead, E Dead]
```

A csata még folytatódik. A győzelem még messze van.

A távoli erdőben mozgásra leszel figyelmes. Jönnek! A fák közül lilán füstölő kőszörnyetegek rontanak elő. Érzed a vérszomjukat. Ezek nem kímélnek semmit és senkit. A szörnyek oldalán meglátod a régi bájjital mestered is.

```
-- Ezt is szintén másold a kódba!
potionMaster =
  let plx x
    | x > 85  = x - plx (div x 2)
    | x == 60 = 31
    | x >= 51 = 1 + mod x 30
    | otherwise = x - 7
  in Master "PotionMaster" 170 plx
```

- Áruló! Na gyere, mókázzunk!

Hallod mestered szavait. Ez nem lesz egyszerű!

## Negyedik feladat (Ütközet)

Amikor két sereg összeütközik, az egységek csak a velük szemben lévőket tudják támadni, a **Golem**-ek egyet, míg a **HaskellElementalok** hármat sebeznek ellenfelükbe, kivéve a mágusok. Ők a képességüket minden olyan ellenséges egységen használják, akik velük szembe vagy hátrébb helyezkednek el. Alkosd meg a **fight** függvényt, amivel megtudod az ellenséges csapat (1. paraméter) támadása után a saját csapatunk (2. paraméter) állapotát.

```
fight :: EnemyArmy -> Army -> Army
```

TÁMADÁÁÁS!

```
fight [] [E (Alive (Golem n)) | n <- [70,70]] == [E (Alive (Golem 70)),E (Alive (Golem 70))]
fight [E (Alive (HaskellElemental n)) | n <- [50,50]] [] == []
fight [E (Alive (HaskellElemental n)) | n <- [50,50,50,50]] [E (Alive (Golem n)) | n <- [70,70,70,70,70]] == [E (Alive (Golem 67)),E (Alive (Golem 67)),E (Alive (Golem 67)),E (Alive (Golem 67)),E (Alive (Golem 70))]
fight [M (Alive potionMaster)] [E (Alive (Golem n)) | n <- [70,70,70,70,70]] == [E (Alive (Golem 11)),E (Alive (Golem 11)),E (Alive (Golem 11)),E (Alive (Golem 11)),E (Alive (Golem 11))]
fight [E (Alive (HaskellElemental 50)), M (Alive potionMaster)] [E (Alive (Golem n)) | n <- [70,70,70,70,70]] == [E (Alive (Golem 67)),E (Alive (Golem 11)),E (Alive (Golem 11)),E (Alive (Golem 11)),E (Alive (Golem 11))]
fight [E (Alive (HaskellElemental 50)), M (Alive potionMaster), E (Alive (HaskellElemental 50))] [E (Alive (Golem n)) | n <- [70,70,70,70,70]] == [E (Alive (Golem 67)),E (Alive (Golem 11)),E (Alive (Golem 8)),E (Alive (Golem 11)),E (Alive (Golem 11))]
```

Ez így nem néz ki jól. Vissza kell vonulni!

Nem áll jól a csata. Az ellenfél túl erősnek bizonyul. Valami új taktika kell ide. Ahogy a csapataid hátrálnak az ellenség csapatai előtt, az égen újabb fények jelennek meg és egyenesen felétek tartanak.

## Ötödik feladat (Robbanások)

A gömbök precízen mindig oda érkeznek, ahol legkorábban a legnagyobb sebzést tudják okozni. Ezek egy 5 nagyságú területen sebeznek (előző feladattól kiderül, hogy mit jelent a "terület"), minden egységbe 5-öt. Írd meg a **haskellBlast** függvényt, ami megadja egy ilyen becsapódás után a sereg állapotát.

```
haskellBlast :: Army -> Army
```

## Vigyázz!

```
haskellBlast [] == []
haskellBlast [E (Alive (Golem 10))] == [E (Alive (Golem 5))]
haskellBlast [M (Alive traktor)] == [M (Alive (Master "Traktor" 15 (\x -> div (x + 10)
((mod x 4) + 1))))]
haskellBlast ([M (Alive traktor)] ++ [E (Alive (Golem n)) | n <- [5,5,5,5]]) == [M (Alive
(Master "Traktor" 15 (\x -> div (x + 10) ((mod x 4) + 1))))],E Dead,E Dead,E Dead,E Dead]
haskellBlast ([E (Alive (Golem n)) | n <- [5,5,5,5,5]]) == [E Dead,E Dead,E Dead,E Dead,E
Dead]
haskellBlast ([E (Alive (Golem n)) | n <- [5,5,5,5,5]] ++ [M (Alive papi)]) == [E Dead,E
Dead,E Dead,E Dead,E Dead,M (Alive papi)]
haskellBlast ([E (Alive (Golem n)) | n <- [10,10,10,10,10,10]]) == [E (Alive (Golem 5)),E
(Alive (Golem 5)),E (Alive (Golem 5)),E (Alive (Golem 5)),E (Alive
(Golem 10))]
haskellBlast ([E (Alive (Golem n)) | n <- [4,10,10,10,10,10]]) == [E (Alive (Golem 4)),E
(Alive (Golem 5)),E (Alive (Golem 5)),E (Alive (Golem 5)),E (Alive (Golem 5)),E (Alive
(Golem 5))]
```

A csapataid nagy sérüléseket szenvedtek. Valamit nagyon gyorsan ki kell találnod. De mit?

- Fogd ezt!

Mestered egy zöld kavicsot dob neked. Ez meg mi a fene lehet?

- Használd!

De mire? Ahogy ezen agyalsz, a kő egyszerűen csak elkezd világítani. Zöld fénye beragyogja egységeid.

## Hatodik feladat (Heal)

Használd a követ a **multiHeal** függvény segítségével, ami egy megadott mennyiséget gyógyít a csapatod tagjain egyenletesen szétszórva a legközelebbitől, a sereg elejétől a legtávolabbiig.

```
multiHeal :: Health -> Army -> Army
```

A csapataidnak szüksége van rád.

```
multiHeal 5 [] == []
multiHeal 5 [E Dead] == [E Dead]
multiHeal 5 [E (Alive (Golem 5))] == [E (Alive (Golem 10))]
multiHeal 5 [E (Alive (Golem 1)), M Dead, E (Alive (Golem 1))] == [E (Alive (Golem 4)),M
Dead,E (Alive (Golem 3))]
multiHeal 5 [E (Alive (Golem n)) | n <- [1,1,1,1]] == [E (Alive (Golem 3)),E (Alive (Golem
2)),E (Alive (Golem 2)),E (Alive (Golem 2))]
multiHeal 5 [E (Alive (Golem n)) | n <- [1,1,1,1,1]] == [E (Alive (Golem 2)),E (Alive
(Golem 2)),E (Alive (Golem 2)),E (Alive (Golem 2)),E (Alive (Golem 2))]
multiHeal 5 [E (Alive (Golem n)) | n <- [1,1,1,1,1,1]] == [E (Alive (Golem 2)),E (Alive
(Golem 2)),E (Alive (Golem 2)),E (Alive (Golem 2)),E (Alive (Golem 2)),E (Alive (Golem 1))]
```

Az embereid már jobban vannak. Újult erővel tudják folytatni a harcot.

Eddig voltak kötelezőek a feladatok, ezen a ponton túl lehet pontokat szerezni, **LEGFELJEBB 5 pont** szerezhető (pontokból több van, így lehet válogatni).

## Hetedik feladat (Harc) (2 pont)

Alkosd meg a `battle` függvényt, amivel láthatod a harcok kimenetelét. A harc során a két csapat összeütközik, majd a Haskell mágus egy lövedéke becsapódik a csapatod közé. Amint ez megtörténik, a kristály segítségével 20 pontot tudsz gyógyítani az embereidbe. Végül mindkét csapat kivonja az elesett egységeit a csata közepéről. Ez addig ismétlődik, amíg valamelyik csapat meg nem nyeri a párharcot. Ennek a csapatnak a végső állapotát, vagy amennyiben nincs győztes, `Nothing`-ot várhatsz vissza.

```
battle :: Army -> EnemyArmy -> Maybe Army {- vagy EnemyArmy lesz az eredmény. -}
```

### Harcra fel!

```
battle [] [] == Nothing
battle [] [E (Alive (HaskellElemental 5))] == Just [E (Alive (HaskellElemental 5))]
battle [E (Alive (Golem 5))] [] == Just [E (Alive (Golem 5))]
battle [E (Alive (Golem n)) | n <- [5,5,5,5]] [E (Alive (HaskellElemental 5))] == Just [E (Alive (HaskellElemental 4))]
battle [E (Alive (Golem n)) | n <- [30,30]] [E (Alive (HaskellElemental 5))] == Just [E (Alive (Golem 40)), E (Alive (Golem 55))]
battle [E (Alive (Golem n)) | n <- [30,30]] [M (Alive potionMaster), E (Alive (HaskellElemental 5))] == Just [E (Alive (Golem 52)), E Dead]
battle [E (Alive (Golem n)) | n <- [30,30]] ([M (Alive potionMaster)] ++ [E (Alive (HaskellElemental 10)) | n <- [50,50,50]]) == Just [E (Alive (Golem 352)), E Dead]
battle [M (Alive papi)] [E (Alive (HaskellElemental 10)) | n <- [100..200]] == Just [M (Alive (Master "Papi" 150 id))]
battle [E (Alive (Golem n)) | n <- [50,50,50,50]] [E (Alive (HaskellElemental 10)) | n <- [50,50,50,50,50,50,50,50]] == Just [E (Alive (HaskellElemental 5)), E (Alive (HaskellElemental 5)), E (Alive (HaskellElemental 5)), E (Alive (HaskellElemental 5)), E Dead, E Dead, E Dead, E Dead]
battle [E (Alive (Golem 20)), M (Alive traktor)] [E (Alive (HaskellElemental 10)) | n <- [100..200]] == Just [E (Alive (Golem 1840)), M (Alive (Master "Traktor" 1867 id))]
```

A csata kiegyenlített, de ez nem elég a győzelemhez. Valamit még ki kell találni. Kell ide egy nagy dobás. Aztán eszedbe jut. Cselekedned kell!

## Nyolcadik feladat (Lánc támadás) (2 pont)

Alkosd meg egy új függvényt `chain` néven. Ezt a függvényt ugráztatod sorban a két sereg között. A saját egységeidet gyógyítod, míg az ellenfelet sebzéd a megadott erőddel, de ennek az értéke minden ugrással csökken 1-gyel, amíg el nem fogy vagy nem tud továbbugrani. De vigyázz, a halottakat csak átugorja.

```
chain :: Amount -> (Army, EnemyArmy) -> (Army, EnemyArmy)
```

Gyerünk, ennek működnie kell!

```
chain 5 ([], []) == ([], [])
chain 5 ([E (Alive (Golem 5))], [E (Alive (HaskellElemental 5))]) == ([E (Alive (Golem 10))], [E (Alive (HaskellElemental 1))])
chain 5 ([E (Alive (Golem n)) | n <- [5,5,5,5]], [E (Alive (HaskellElemental n)) | n <- [10,10,10,10]]) == ([E (Alive (Golem 10)), E (Alive (Golem 8)), E (Alive (Golem 6)), E (Alive (Golem 5))], [E (Alive (HaskellElemental 6)), E (Alive (HaskellElemental 8)), E (Alive (HaskellElemental 10)), E (Alive (HaskellElemental 10))])
```

```
chain 5 ([E (Alive (Golem n)) | n <- [5,5]], [E (Alive (HaskellElemental n)) | n <- [10,10,10,10]]) == ([E (Alive (Golem 10)),E (Alive (Golem 8)),E (Alive (HaskellElemental 6)),E (Alive (HaskellElemental 8)),E (Alive (HaskellElemental 10)),E (Alive (HaskellElemental 10))])
chain 5 ([E (Alive (Golem n)) | n <- [5,5,5,5]], [E (Alive (HaskellElemental n)) | n <- [10]]) == ([E (Alive (Golem 10)),E (Alive (Golem 8)),E (Alive (Golem 5)),E (Alive (Golem 5)),E (Alive (HaskellElemental 6))])
chain 5 ([E Dead, E (Alive (Golem 5)), E Dead], [E (Alive (HaskellElemental n)) | n <- [10,10,10]]) == ([E Dead,E (Alive (Golem 9)),E Dead],[E (Alive (HaskellElemental 5)),E (Alive (HaskellElemental 7)),E (Alive (HaskellElemental 8))])
chain 5 ([E Dead | _ <- [1..5]], [E (Alive (HaskellElemental n)) | n <- [10,10,10,10,10]]) == ([E Dead,E Dead,E Dead,E Dead,E Dead],[E (Alive (HaskellElemental 5)),E (Alive (HaskellElemental 6)),E (Alive (HaskellElemental 7)),E (Alive (HaskellElemental 8)),E (Alive (HaskellElemental 9))])
```

Ezaz, ezzel a csatát a te oldaladra tudtad fordítani.

Ezzel az új fegyverrel nagy előnyre tettél szert. Ezzel meg is nyerhetitek a csatát. Még egy utolsó hajrá kell és le tudjátok győzni a Haskell mágus csapatait. Már nem kell sok.

## Kilencedik feladat (Győzelem) (1 pont)

Az új fegyvereddel új taktikát használhatsz a csatában. Minden körben, mielőtt még rendeznétek az elesetteket, használd a korábban elkészített **chain** függvényt és vidd vele győzelemre a seregged. A varázslatot 5-ös erősséggel tudod alkalmazni.

```
battleWithChain :: Army -> EnemyArmy -> Maybe Army {- vagy Maybe EnemyArmy -}
```

Ez a győzelem már a zsebedben van.

```
battleWithChain [] [] == Nothing
battleWithChain [] [E (Alive (HaskellElemental 5))] == Just [E (Alive (HaskellElemental 5))]
battleWithChain [E (Alive (Golem 5))] [] == Just [E (Alive (Golem 5))]
battleWithChain [E (Alive (Golem n)) | n <- [5,5,5,5]] [E (Alive (HaskellElemental 5))] == Nothing
battleWithChain [E (Alive (Golem n)) | n <- [30,30]] [E (Alive (HaskellElemental 5))] == Just [E (Alive (Golem 37)),E (Alive (Golem 38))]
battleWithChain [E (Alive (Golem n)) | n <- [30,30]] [M (Alive potionMaster), E (Alive (HaskellElemental 5))] == Just [E (Alive (Golem 44)),E (Alive (Golem 36))]
battleWithChain [M (Alive papi)] [E (Alive (HaskellElemental 10)) | n <- [100..200]] == Just [M (Alive (Master "Papi" 160 id))]
battleWithChain [E (Alive (Golem n)) | n <- [50,50,50,50]] [E (Alive (HaskellElemental 10)) | n <- [50,50,50,50,50,50,50,50]] == Just [E (Alive (Golem 68)),E (Alive (Golem 53)),E (Alive (Golem 39)),E (Alive (Golem 33))]
battleWithChain [E (Alive (Golem 20)), M (Alive java)] [E (Alive (HaskellElemental 10)) | n <- [100..200]] == Just [E (Alive (Golem 734)),M (Alive (Master "Java" 616 id))]
```

## Győzelem!

A csapataid győzelmet arattak a Haskell mágus csatlósai felett, de ő még nem fedte fel magát.

Egyszer csak egy hang hív téged az erdőből. Egy rekedt nevető hang. A mágus az. Követed a hangot, de egyedül. Ez a ti kettőtök leszámolása lesz.

Amint belépsz az erdőbe, a levegő megfagy körülötted. A távolban a kék köpenyes alak néz szembe veled. Csapzott haja lobog a szélben. Az őrület tüze fénylik szemében. Itt az idő.

- Most eldől minden. Innen te nem távozol.

## Tizedik feladat (A párbaj előkészülete) (1 pont)

A párbaj szerkezete rekurzívan ismétlődik, amíg valaki meg nem nyeri azt. Egy elem tartalmazza, hogy az kinek a köre volt és mennyi az életereje.

```
data OneVOne = Winner String | You Health OneVOne | HaskellMage Health OneVOne deriving Eq
```

Ennek a párbaj megjelenítése (Show példány) < > jelek között történik. A lépések ; -vel vannak elválasztva és a nyertes állapot || -jelek közé kell essen.

```
showOneVOne (Winner "") == "<|| Winner ||>"
showOneVOne (You 5 (Winner "")) == "<You 5; || Winner ||>"
showOneVOne (HaskellMage 10 (You 5 (Winner "a"))) == "<HaskellMage 10; You 5; || Winner a ||>"
showOneVOne (You 10 (You 5 (Winner "a"))) == "<You 10; You 5; || Winner a ||>"
```

A Haskell mágus neked támad. Mágiája felettébb erős, de te még állod a sarat. Folyamatosan bombáz és bombáz mágiájával. Egy visszatámadásod egyenesen telibe találja őt. Erre hátra hőköl és gyógyítani kezdi magát, de közben érzed, hogy a te életedből meríti az erőt. Kihasználod a lehetőséget és betalálsz megint egy képességeddel.

## Tizenegyedik feladat (A végső leszámolás) (3 pont)

Ahogy a küzdelem halad, kiismered a Haskell mágus taktikáját. Fontos neki, hogy életben maradjon. Ha az életereje négy alá csökken, akkor a te életerődből merít energiát, ezzel feleazve azt és a sajátját négyszeresére növeli. Ha sok életereje van, és amíg a te saját életerőd húsz felett van, akkor az életed háromszorosát a negyedére csökkenti. Minden más esetben az eddigi információid alapján tizenegyet sebez beléd. Ezekből egyszerre csak egyet tud használni, így két képessége között van idő a sajátjaiddal visszatámadni.

Ezért érdemes a következő stratégiát követned. Élned fontos, ellenkező esetben a Haskell mágus nyer, akkor az csak további káoszt fog okozni. Ha az életerőd 4 alá csökken, akkor gyógyítsd fel magadat a négyszeresére. Ha sok életed van, akkor ameddig a Haskell mágus élete tizenöttnél nagyobb, be tudsz támadni egy erősebb mágiával, amivel az életének háromszorosát az ötödére csökkented, különben kilencet sebzél rá.

Használd a `finalBattle` függvényt, amivel nyomon tudod követni az eseményeket. A függvény számon fogja tartani, hogy éppen ki használt képességet és akkor mennyi volt annak életereje, végül a győztest is.

```
finalBattle :: Health -> Health -> OneVOne
show (finalBattle 500 500) == "<HaskellMage 500; You 375; HaskellMage 300; You 281; HaskellMage 180; You 210; HaskellMage 108; You 157; HaskellMage 64; You 117; HaskellMage 38; You 87; HaskellMage 22; You 65; HaskellMage 13; You 48; HaskellMage 4; You 36; HaskellMage 0; || Winner You ||>"
```

A csatának vége. A Haskell mágus halott. Még utolsó leheletével egy varázslaot küld rád, ami telibe talál. A sebzés nagyon súlyos. A még megmaradt csepp erőddel próbálsz magad gyógyítani, amikor is a mágus testéből fény kezd áradni. A fény feléd irányul; újult erőre kapsz



tőle. Hirtelen begyógyulnak sebeid és hatalmas erő birtokába kerülsz. A párbajt megnyerted. A Haskell mágus legyőzve, de az univerzum stabilitásának érdekében mindig kell legyen egy Haskell mágus. A sorsod megpecsételődött. Rád esett a választás és ezzel te lettél az új Haskell mágus.

Az utad itt még nem ért véget. Lesznek, akik megkérdőjelezik hatalmad és tudásod. Még sok kihívás vár rád, de azokat meg kell ugranod.

Mesteredhez visszatérve beszámolsz a párbajról, a nagy csatáról részletesen. A mestered arcáról vegyes érzelmeket olvasol le. Látszódik rajta, hogy büszke rád; de sokkal jobban aggaszt, hogy *már megint* ki fog oktatni, hogy mit tehettél volna jobban. Szerencsédre ezúttal a hosszú monológjától megkímél, mégis csak nagy eredményt értél el, a mestered nem gondolta volna, hogy ennyire könnyűszerrel ilyen rövid idő alatt megtanult varázslatokkal le tudod győzni a Haskell mágust.

Elindultok visszafelé az Elit Tekercsmágiai Integrációs Képző területére, épületébe. Út közben a mestered jó szokásához híven mégsem kímélt meg a mondanivalójától:

- Amikor már érzed, hogy gyengülsz, már mozdulni nem bírsz, de a másikon hasonló jeleket látsz, akkor kell összeszorítani a fogad, szád, lábad, két kezed, és akkor kell a legutolsó minden erőt beleadni, egy erős támadást indítani, azzal befejezni a csatát. Ettől lehet fejlődni, hogy a saját határainkat megkérdőjelezzük, azokat átlépjük, nem adjuk fel soha, ekkor lehet sokkal magasabbra jutni.

Ahogy visszaértek a képző területére, a mestered a beszédét még nem hagyta abba (pedig lassan megtehetné, valószínűleg ennyi beszéd után már kiszáradt az ajka, szája):

- A világ furcsa hely, de van egy gondolatmenete, ami alapján működik, ha azt megismered, akkor már senki nem tud megállítani, hogy azt tehess, amit szeretnél. Hogy ezt a tudást jóra vagy rosszra használod fel, ez már a te döntésed.

Az épületben a 2.620-as szobába visszaérve mestered már előkészített egy oklevelet, amely szerint a mágusképzőt sikeresen elvégezted. Mestered búcsúzóul az alábbiakat mondja:

- Én már több mókás dolgot nem tudok mutatni neked. Egy kis tapasztalat a nagy világban is vár kint rád, megismerkedhetsz új emberekkel, új varázslatokkal, mókás helyzetekkel, amikből és akiktől még többet tanulhatsz. Sok szerencsét! Szükséged lesz rá! De ne feledd, ha bármi kérdésed van, az ajtó nyitva áll!

A mestered korábbi beszédén merengsz, vajon mit érthetett az alatt, hogy "senki nem tud megállítani, hogy azt tehess, amit szeretnél"? Mind eközben visszagondolsz a kezdetekre, az első órákra, az akkori sok verejtékre és könnycsepre, mestered feladataira. Idővel megszépültek ezek az emlékek, visszagondolva azokat játszó könnyedséggel tudnád most már megcsinálni.

Az életed egy nagy szakasza zárult most le, ezzel együtt egy új kezdődik. Vajon milyen izgalmas dolgokat rejteget? Nagy izgatottsággal teszed meg az első lépést az új világ felé, és emlékszel mestered szavaira: "Sok szerencsét! Szükséged lesz rá!"

***Created by: Tóth Máté, Bense Viktor***