

1.Házi Feladat

A kezdetek

Egy zord és viharos éjszakán hőseink a fővárosi fogadók mélyén mesélnek kalandozásaikról. A fogadók telis tele vannak élettel, ahogy a világ minden sarkáról érkeznek erre kalandorok, hogy szerencsét próbáljanak. Ahogy zajlik az élet, a kinti viharból egy sebzett vándor esik be ezzel megzavarva az örömteli mulató hangulatot. Amint beesik az ajtón észereveszed, hogy a testét súlyos sebek borítják. GYORSAN! CSELEKEDNED KELL.

Első feladat (gyógyítás)

Mágus létedre jártas vagy pár gyógyító mágiában, ami segíthet ebben a szituációban. Alkoss egy függvényt `lesser_heal` néven, ami megnöveli a megadott életerőpontot 3-mal. Mivel még nem vagy jártas annyira ezekben a körökben, így ehhez csak a tréningeden tanult `inc` függvényt használhatod.

```
lesser_heal :: Int -> Int
```

Most, hogy megalkottad, próbáld ki a vándor életének megnövelését ezzel.

```
lesser_heal (-1) == 2
lesser_heal 2 == 5
lesser_heal 5 == 8
```

A vándor már jobban érzi magát, de még nem lankadhat a figyelmed. Ha egy halandó túl sok mágikus energiának lesz kitéve, akkor a teste nem tudja mindet befogadni és annak súlyos következménye lehet.

Második feladat (megfigyelés)

Még a régmúltbeli tanulmányaid során emlékszel, hogy egy mestered a halandók anatómiájáról beszélt nektek. Megmutatta, hogyan tudod eldönteni egy emberi testről, hogy alkalmas-e még mágikus energia befogadására. Készíts egy függvényt `lookout` néven. Ez a függvény megnézi, hogy a halandó életeréje (első paraméter) nagyobb-e, mint a testébe bejutott mágikus energia (második paraméter) 10-zel való egészrésztes hányadosa. Ha ez igaz, akkor a teste be bír még fogadni mágikus energiát.

```
lookout :: Int -> Int -> Bool
```

Most, hogy kész a függvény, vizsgáld meg vele a testet.

```
lookout 5 10
lookout 8 40
lookout 11 90
not (lookout 14 150)
not (lookout 14 180)
```

Harmadik feladat (fáradtság)

Fuh, ez közel volt. A vándor már jobban van, viszont ez téged eléggé kifárasztott. Észreveszed, hogy ez a művelet a te testedből több mágikus energiát vont el, mint amennyit az idegen teste befogadott. Alkosd meg a `volume` függvényt, ami megadja, hogy mennyi mágikus energiára van

szükséged, hogy egy megadott mennyiségű energiát (első paraméter) bele tudj áramoltatni egy másik testbe. A tőled elvont mágikus energia megegyezik a bejuttatott mágikus energia (első paraméter) és az elveszett energia összegével. Az elveszett energiát megkapod, ha a fáradtsági szintedet (második paraméter) megszorozod a bejuttatott energia (első paraméter) 12-vel való osztási maradékával.

```
volume :: Int -> Int -> Int
```

Nézd meg, hogy mennyi energiára volt szükséged a művelet során.

```
volume 10 1 == 20
```

```
volume 20 2 == 36
```

```
volume 30 2 == 42
```

```
volume 30 3 == 48
```

Szerencsére mindenki megmenekült, de ez közel volt. Vajon mi okozta azokat a sebeket?

2. Házi Feladat

Típus magyarázó

A teljes olvasási élmény érdekében pár alap típust átneveztünk. Kérlek ezeket vegyétek figyelembe.

```
type CurrentDistrict = Int
type NextDistrict = Int
type HealthDamage = Int
type ArmorDamage = Int
type Health = Int
type Armor = Int
type Enhance = Int
```

Látogatás

Az előző este nem hagyott nyugodni. Azok a sebek ... mi okozhatta? Nem fegyver által ejtett sebek voltak, inkább állati marcangolásra hasonlítottak. A főváros közepén hol szerezhetett ilyen sebeket ez a szerencsétlen? Elhatározod, hogy utánajársz ezeknek a furcsa körülményeknek, de előbb még elfogyasztod a reggeli Jeve italodat. Ahogy a szádhoz emeled megcsap a frissen gőzölgő aromatikusan illata. A reggeli elfogyasztása után érzed, hogy feltöltődött az energiád és újjult erővel tudsz nekivágni a napnak. A napod első úticélja a Sürgős Orvoslási Tudományos Ellátóba azzal a reménnyel, hogy többet fogsz megtudni azokról a különös sebekről, amiket tegnap láttál.

Első feladat (Tájékozódás)

A fővárost ominózus nagysága miatt 23 kerületre osztották, hogy megkönnyítsék a tájékozódást. A helyiektől megtudod, hogy az ellátó a 8-as kerületben van. Ahhoz, hogy eljuss oda alkoss egy függvényt `move` névvel, ami segít neked a tájékozódásban és mindig továbbléptet a következő kerületbe.

```
move :: (CurrentDistrict, NextDistrict) -> NextDistrict
```

Most, hogy tudsz már tájékozódni a városban, juss el az ellátóba.

```
move (11,9) == 9
move (9,5) == 5
move (5,6) == 6
move (6,7) == 7
move (7,8) == 8
```

Megérkeztél az ellátóba. Az út kicsit hosszabb volt, mint vártad. Lehet, kicsit eltévedtél út közben, de semmi baj, idővel bele fogsz jönni.

Az ellátóban összefutsz egy régi barátoddal, akivel még együtt tanultatok előző mestereitektől. Már egy ideje nem találkoztatok. Neki nagyobb affinitása volt a támogatói mágikák felé, így ő arra folytatta tanulmányait. Elkezdtek beszélgetni, a régi szép időkről merengeni. Ennek folyamán tudomást szerzel egy különös betegről, akit az este hoztak be. Az illetőnek fura sebek borították a testét, amik valami miatt az este folyamán elkezdtek elfertőződni. A barátod volt épp, aki felvette a beteget és megkezdte az ellátását. Megemlítette, hogy a sebeket valaki már elkezdte kezelni, de nagyon kontár munkának tartotta. Állítása szerint az illető olyan primitív mágiaival látta el a

sebeket, hogy csoda, hogy nem okozott nagyobb kárt a testben. Ez a beteg felkelti az érdeklődésed, mivel a leírás alapján nagyon hasonló esetnek tűnik, mint amivel tegnap találkozta. Többszöri győzködés után, barátod megengedi, hogy te is szemügyre vedd a beteget, hátha előrébb jutsz nyomozásodba.

Amint beléptek a kórterembe, egy horrorisztikus látvány fogad benneteket. A beteg sebei elfertőződtek és valamiféle halványlilas elváltozás mutatkozik a bőrén a sebek körül. Barátod gyorsan cselekedni kezd. Odarohan a vészjelző gombhoz és riadót fúj az épületben. A beteg teste egyszer csak elkezd rángatózni, majd felpattan az asztalról. Szemében csak a sötétséget látod. Nekiront barátodnak, aki éppen sikerrel el tud ugrani előle, de megsérül közben. Rajtad áll, hogy le tudod-e győzni ezt a szörnyeteget.

Második feladat (Támadás)

Gyorsan! Cselekedned kell. Visszaemlékszel arra, amit a képzésen tanultál.

Egy `arcane_missiles` nevű függvénnyel támadd meg a bestiát. (Az `arcane_missiles` bekéri a támadás erejét és a célpont életerejét, majd visszaadja a célpont életerejét a támadás következtében)

```
arcane_missiles :: (HealthDamage , ArmorDamage) -> (Health , Armor) -> (Health , Armor)
```

Ezaz! TÁMADÁÁÁÁÁÁÁÁÁÁÁÁ!!

```
arcane_missiles (5, 1) (100, 200) == (95, 199)
```

```
arcane_missiles (4, 2) (95, 199) == (91, 197)
```

```
arcane_missiles (0, 0) (91, 197) == (91, 197)
```

```
arcane_missiles (1, 1) (91, 197) == (90, 196)
```

A szörnyetegnek ez meg se kottyant, viszont most felmérgetted. VIGYÁZZ! A szörny feléd veti magát, de még épp kitérsz az útjából. Ez így nem fog működni. Az a lila dolog olyan, mintha valamilyen páncélként védené. Valami erősebbre lesz szükség. Eszedbe jut egy másik függvény, `arcane_blast`. Ezt még egy régi mesterreddel tanultad, de sosem sikerült tökéletesen elsajátítanod. Bajban vagy. A barátod harcképtelen, túl nagy a sérülése. Gondolkoznod kell! A barátod nem tud harcolni, de a te mágiádat még tudja támogatni. EZAZ, ez lesz a megoldás!

Harmadik feladat (Erősítés)

A barátod besegít az `arcane_missiles` támadásod felerősítésével. A felerősítés egy szorzót fog adni a páncél és az életerő sebzésben.

```
arcane_missiles_mark_1 :: Enhance -> (HealthDamage , ArmorDamage) -> (Health , Armor) -> (Health , Armor)
```

Érzed, hogy erősebb vagy, talán így sikerülni fog.

```
arcane_missiles_mark_1 1 (1, 1) (90, 196) == (89, 195)
```

```
arcane_missiles_mark_1 3 (2, 6) (89, 195) == (83, 177)
```

```
arcane_missiles_mark_1 2 (5, 10) (83, 177) == (73, 157)
```

```
arcane_missiles_mark_1 0 (10, 15) (73, 157) == (73, 157)
```

Áh! Pedig már majdnem jó volt, de ez még mindig kevés. Meg kell próbálnotok az `arcane_blast`-ot közösen.

Negyedik feladat (Arcane blast)

Az `arcane_blast` egy páros támadást igénylő függvény, ami a két támadás skaláris szorzatának megegyező értékét sebez az ellenfele páncéljába és életerejébe. A használata nagyon veszélyes, ugyanis ha az egyik támadónak is lankad a figyelme, azzal csökkenhet a támadás hatása.

```
arcane_blast :: (HealthDamage , ArmorDamage) -> (HealthDamage , ArmorDamage) -> (Health , Armor) -> (Health , Armor)
```

Most, hogy felkészültetek, itt az idő! **MOST VAGY SOHA!**

```
arcane_blast (3, 0) (8, 15) (73, 157) == (49, 133)
arcane_blast (5, 1) (1, 5) (49, 133) == (39, 123)
arcane_blast (10, 10) (0, 0) (39, 123) == (39, 123)
```

GYERÜNK!! ÖSSZHANGBAN!!

```
arcane_blast (15, 10) (10, 11) (39, 123) == (-221, -137)
```

EZAZ! MEGCSINÁLTÁTOTK! Bár az utolsó támadás után nem sok minden maradt a testből ... és igazából a hátsó falból se ..., szerencsére megmenekültetek.

A nap hátralévő részét azzal töltöd, hogy a sebeidet ellásd, amíg nem kapsz értesítést, hogy a barátod látni akar téged. A délutáni találkozó alkalmával pár érdekes dologra hívja fel a figyelmed, amit az ellátóban történt csata után találtak. A lila elváltozás, ami azon a szegény emberen volt nagyon erős mágia okozta fertőzésből jöttek. Ilyen erőset még sose láttak, de még ilyen színűt sem. Lila. Ez sajnos csak egy dolgot jelenthet. A régi mestered egy fontos feladattal látott el téged, ami miatt a fővárosba látogattál. Megbízott téged azzal, hogy fejleszd tovább képességeidet és vizsgálj ki egy gyanúját a helyi mágikus energiák körül. Azt gyanítja, hogy valami szörnyen nagy és veszélyes mágus korruptálódott az általa birtokolt mágia miatt. Sajnos ezek a jelek is ezt igazolják. A Haskell mágus elszabadult.

3. Házi Feladat

Az új mester

Típus magyarázó

A teljes olvasási élmény érdekében pár alap típust átneveztünk. Kérlek ezeket vegyétek figyelembe.

```
type Room = String
type Air = Int
type Fire = Integer
type MagicLevel = Int
```

Az újonnan szerzett felfedezésed elég váratlanul ért. Mindig azt gondoltad, hogy az öregnek nincs ki mind a négy kereke. A másik dolog, ami igencsak aggaszt, hogy még nem vagy elég erős. A tegnapi harc nagyon kimerített téged, és túl közel volt. Fejlődnöd kell, de hogyan? Találnod kell egy új mestert! A közelben található az **ELit TE**kercsmágiai Integrációs **Képző**. Ott talán akad valaki, aki felfogadna téged tanoncának.

Első feladat (A mester)

A Képzőben el irányítanak egy szobába, a 2.620-as szobába. Azt mondják, ott van egy mester, aki hajlandó új tanoncokat oktatni. Keresd meg a 2.620-as szobát.

```
find :: [Room] -> [Room]
```

Lássuk, hol van az a szoba.

```
find ["00.412", "00.503", "00.112", "00.114"] == []
find ["0.814", "0.826", "0.409", "0.411"] == []
find ["2.513", "2.709", "2.710", "2.107"] == []
find ["2.620"] == ["2.620"]
```

Áh, meg is van.

Bekopogsz a szobába, majd belépsz. Bent egy magas, hosszú hajú férfi látványa fogad téged. Szemüvegén keresztül vizsgál végig, amikor egy alacsony köpcösebb figura jelenik meg mögötted. A második figura tudni akarja, mit is keresel te itt!

Erre te magad se tudod a választ. Te csupán egy mestert keresel magadnak.

Ennek hallatán a köpcös figura elmosolyodik, kitörli a reggeli fáradság maradványát is a szemeiből (pedig már délután 2, ha nem 3 fele járhat) és arra int, hogy kövessed őt. Látszik, hogy még életében nem láthatta a napfelkeltét, maximum, ha megvárta ébren. Te követed és reméled a legjobbakat. Elvezet téged egy hátsóbb terembe.

Megtudod tőle, ha te itt akarsz tanulni, akkor ki kell állj pár próbatételt. Még gyengének érzed magad az előző nap eseményeitől, de reméled, hogy gyorsan túl leszte rajta.

Második feladat (Első próba)

Az első próba arra irányul, hogy tudod-e kontrollálni és változtatni a mágiád jellegét. Írj egy `add` függvényt, ami kombinálja `Fire` és `Air` típusú mágiákat és ez egy erősebb mágiát eredményez.

```
add :: (Integral magic1, Integral magic2, Num magic3) => magic1 -> magic2 -> magic3
```

És az első próba eredménye:

```
add (2 :: Air) (3 :: Fire) == (5 :: Fire)
add (2 :: Air) (3 :: Air) == (5 :: Air)
add (2 :: Fire) (3 :: Air) == (5 :: Fire)
add (2 :: Fire) (3 :: Fire) == (5 :: Fire)
```

Remek, megbizonyosodtak arról, hogy tudod kontrollálni a mágiád típusát. Az első próbát sikeresen kiálltad, de több van még hátra.

Harmadik feladat (Második próba)

Most arról kell számot adnod, hogy a mágiád mértéke felett mekkora irányításod van. Nem lehet úgy csatába menni, hogy már az első mágia kimerít téged. Minden mágia irányíthatóság szempontjából könnyebb, ha a mennyisége megegyezik egy prím számmal. Ezeket hívják prímamágiának. Ezen a próbán adott két mágiaszint. A feladatod az, hogy a két mágiaszint között üsd meg az összes prímamágia szintet.

```
prime_magic :: MagicLevel -> MagicLevel -> [MagicLevel]
```

Teszteljük le a képességeid.

```
not (prime_magic 0 0 == [0])
prime_magic 0 0 == []
prime_magic 2 0 == []
not (prime_magic 1 1 == [1])
prime_magic 2 2 == [2]
prime_magic 0 10 == [2,3,5,7]
prime_magic 10 100 == [11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97]
```

REMEK

Ez a próba már kezdi kivenni a maradék erődöt is. Elkezdte hibázni. Már nem is tudod, mikor hibáztál utoljára. Nem tudod, még meddig bírod tovább, de kitartasz. Küzdened kell. Tudod alapszinten uralni a mágiádat, de ez még nem lesz elég. Ahhoz, hogy legyen esélyed a Haskell mágust legyőzni, erősebb mágiakontrollról kell tanúbizonyságot tenni.

Negyedik feladat (Harmadik próba)

A harmadik próba során meg kell mutatnod, hogy mennyire vagy képes feldarabolni a mágiát. Most rád fognak támadni. A te feladatod, hogy a támadást feldarabold az azonos részek mentén kisebb egységekre. A felbontás során uralnod kell a mágia forrását és a mágia mennyiségét is egyszerre. Erre alkossd meg a `compress` függvényt. Ennek segítségével darabold fel a beérkező mágiát az alkotóelemeire és tömörítsd össze őket az alapján, hogy egymás után mennyi ugyanolyan típusú elem jön. Segítségedre lehet a `Data.List`-ből ismert `group` függvény.

```
compress :: Eq magic => [magic] -> [(magic, MagicLevel)]
```

Rendben, most lássuk mennyire tudsz ezzel védekezni.

```

not (compress "agh" == [])
compress "covaanaa" == [('c',1),('o',1),('v',1),('a',2),('n',1),('a',2)]
compress "Abbrakkkkaa Dabbbrraaa" == [('A',1),('b',3),('r',1),('a',1),('k',4),('a',3),('
',1),('D',1),('a',1),('b',3),('r',2),('a',3)]
not (compress "Air" == [('A',1),('r',1)])
compress "Fireee" == [('F',1),('i',1),('r',1),('e',3)]
compress "HASKke1111" == [('H',1),('A',1),('S',1),('K',1),('k',1),('e',1),('l',4)]
compress "" == []

```

Ez azért már elég közel volt, de rajtad ez se fogott ki. A mágiákat így darabokban könnyebb kezelni, mint gondoltad.

Ekkor egy harmadik férfi lépett be a terembe. Fiatalabb, bár nem sokkal, mint a másik. Szemüveges, szakállas. Amikor bejött, alaposan végigmért. Észrevette a kimerültségedet. Az arcán láttad, hogy átérzi a fájdalmad. Óvatosságra intette a köpcös figurát, de az csak mosolygott ezen. Még egy próba. Már a tűréshatárodát feszegeted, de nem most fogod feladni. Túl közel van a cél.

Ötödik feladat (Negyedik próbatétel)

A feladat egyszerű. Mivel szét tudtad bontani hibátlanul a beérkező támadásokat, most a cél az, hogy összerakd őket és azzal visszatámadj. Erre alkossd meg a **decompress** függvényt, ami újra összerakja az adott mágiákat a darabjaikból. Ehhez érdemes lehet segítségül hívni a **replicate** függvényt.

```
decompress :: Eq magic => [(magic, MagicLevel)] -> [magic]
```

Gyerünk, most visszatámadhatsz. Ez a lehetőség keveseknek adatik meg ellene, használd ki!

```

decompress [('c',1),('o',1),('v',1),('a',2),('n',1),('a',2)] == "covaanaa"
decompress [('A',1),('b',3),('r',1),('a',1),('k',4),('a',3),('
',1),('D',1),('a',1),('b',3),('r',2),('a',3)] == "Abbrakkkkaa Dabbbrraaa"
decompress [('F',1),('i',1),('r',1),('e',3)] == "Fireee"
decompress [('H',1),('A',1),('S',1),('K',1),('k',1),('e',1),('l',4)] == "HASKke1111"
decompress [] == ""

```

Ezt be kell vallani, jól esett.

A férfi, aki utólag érkezett, elmosolyodott a látványon, ahogy barátja kapott egy párat a váratlanabb támadások közül. Ezek után odalépett és beszélni kezdett vele. Tőled messzebb állnak és nem tudod minden szavukat kivenni, de hallod, hogy arról van szó, hogy már nem kéne tovább kínozni téged. Egyszer csak vallatód felnevet ezen, és valami olyat hallasz, hogy ő már rég eldöntötte a sorsodat a képzésen. Ettől kicsit mérges leszel ... akkor meg miért kellett ennyit vallatni...?

A diskurzus után odalépnek mindketten hozzád, majd a szakállas megszólalt:

- Gratula, a Papival nehéz lesz, de megéri a szenvedés.

Papi? De hát annyira nem is öreg ...

4. Házi Feladat

A tiltott könyv

Típus magyarázó

A teljes olvasási élmény érdekében pár alap típust átneveztünk. Kérlek ezeket vegyétek figyelembe.

```
type Title = String
type Count = Integer
```

A tegnapi teszt kivett belőled minden erőt. Még a másnap nagyobb felében is a végkimerülés határain lépkedsz. Céltalanul bolyongsz a folyosókon, aztán elhatározod, hogy kifekszel az épület *B* szárnyában. A *B* szárnyban egy füves placc van kialakítva, hogy az ottaniak közelebb érezzék magukat a természethez és tudjanak ott relaxálni egy kicsit.

Ahogy fekszel a fűben, körülbelül 3 - 4 óra magasságában egy hang ébreszt fel téged.

- Hát te? Mit csinálsz itt a fűben? Gyere, gyakoroljunk vitszes mágikus feladatokat!

Ahogy kinyitod a szemed és látod, hogy újdonsült mestered nem hagyja, hogy a tegnapi szadizmusát feldolgozd. Tudod, hogy igaza van ... A Haskell mágus sem fog neked pihenőidőt adni. Meg kell erősödnöd. Első utatok a könyvtárba vezet.

Első feladat (Mókás könyvszámolás)

A mestered levesz egy könyvet a polcról. Első feladatnak meg kell számolnod a mágiád segítségével, hogy hány o vagy 0 betűt tartalmaznak a könyvek címei. Ehhez alkosd meg a o_o_count rekurzív függvényt, ami a segítségedre lesz ebben a feladatban.

```
o_o_count :: Title -> Count
```

Ok, ez bemelegítésnek nem is volt rossz, nézzük meg, hogy működik-e.

```
o_o_count "Morellonomikon" == 4
o_o_count "Necronomikon" == 3
o_o_count "Haskellium" == 0
o_o_count "The Song Of Ice And Fire Book 6 (draft)" == 4
```

Oh! Szóval itt rejtegeti! Vajon mennyit adnának érte a fekete piacon?

```
o_o_count "LoremIpsum" == 1
o_o_count "The Libris Mortis" == 1
o_o_count "" == 0
```

Ez túl könnyű volt. Reméled, hogy ennél azért majd kapsz nehezebb feladatot. Ha már egyszer csinálod, akkor nem akarsz ilyen badarságokkal tölteni az idődet.

Második feladat (Több is van?)

A következő feladatod nem lesz más, mint megállapítani egy listáról, hogy hosszabb-e, mint az adott szám. A trükk pedig az, hogy nem feltétlen tudhatjuk, hogy mekkora a listánk. A lista

folyamatosan növekedhet, de minket csak az érdekel, hogy nagyobb-e, mint egy megadott szám.

Erre alkossd meg a `longerThan` függvényt!

```
longerThan :: [item] -> Count -> Bool
```

Ezek a feladatok egyre unalmasabbak ...

```
not (longerThan [] 5)
longerThan ["Books"] 0
not (longerThan [] 0)
longerThan ["Ez", "A", "Feladat", "Unalmas"] 2
longerThan [1..] 10000
```

- Ugye milyen mókásak ezek a feladatok?

Erre inkább válaszolni se akarsz ... A semmiből viszont elkezded egy hangot hallani. Hívogat téged a távolból. Úgy döntesz utána mész. A hang egy könyvhöz vezet téged. Vörös bőrkötéssel rendelkezik, díszes vörös mintázattal. A könyv szól hozzád. Ki akarod nyitni. Ebben a pillanatban a mestered lép közbe.

- Azt én nem tenném. Nem éri meg.

Nem érted miről beszél. És amúgy is, mit tudhat ő. Ez a könyv igazi erőt tartalmaz. Ezzel a könyvvel erősebb lehetsz, mint valaha, érzed.

- Az a könyv nem fog neked erőt adni, csak elvenni.

Persze, mit tudhat ő. A könyv szólít téged és érzed, ahogy árad belőle az erő, de most mesterreddel mész. Az edzésnek mára vége.

Másnap reggel megint bent vagy a Képzőben. Az egyik tanuló sarokban a tegnapi könyvön merengsz. Úgy érzed a könyv téged választott. Ahogy ezen rágódsz, megint hallani kezded a tegnapi hangot. Megint szólít téged. Mit tudhat a te mestered, ő sose érezhetett ilyet. Hogyan is lehetne ennek ellent mondani?

Úgy döntesz felkeresed a könyvet. Ott találsz, ahol tegnap hagytátok. A könyv kötése egy állványon vöröslik. Felnyitod.

Harmadik feladat (Mágia összefűzés)

A könyv kinyitásával új erőre kapsz. A hang a fejedben egyre hangosabb. A hang azt a feladatot adja, hogy hozd létre a rekurzív `merge` függvényt, ami két mágialistát összefűz.

```
merge :: [magic] -> [magic] -> [magic]
```

Ez már méltóbb gyakorlás számodra.

```
merge [1,3,5,7] [2,4,6,8] == [1,2,3,4,5,6,7,8]
merge [] [1,2,3] == [1,2,3]
merge [1,2,3] [] == [1,2,3]
merge [] [] == []
merge "Hse1" "ak1" == "Haskell"
```

Ez az erő már jobban tetszik neked, de a könyv még többet kér. Ez még nem elég. Lehetsz te ennél még sokkal erősebb. Még több kell neked!

Negyedik feladat (Mágia összehasonlítás)

A könyv azt kéri, hogy hasonlíts össze két mágiaát. Létre kell hoznod a rekurzív **starter** függvényt, ami megállapítja, hogy az első mágia kezdete-e a második mágiának.

```
starter :: Eq magic => [magic] -> [magic] -> Bool
```

Ez már kezd kifogni rajtad, de még megy.

```
starter [1,2,3] [1,2,3,4]
starter [] [1]
not (starter [1] [])
starter [] [1..]
not (starter [1..] [])
not (starter [1..] [1..10])
starter [] []
starter [1..10] [1..]
starter "starter" "starterMagic"
```

Ez már kezd kicsit sok lenni, de a könyv még többet akar. A bővületébe kerített. Nem tudsz ellent mondani neki. Csinálnod kell, de nem tudod már meddig bírod. Kell neked a hatalom.

Bónusz feladat (Mágia sokszorozás)

A könyv azt kéri tőled, hogy egy mágialistát sokszorozz az alapján, hogy visszaadod az összes végződését. Erre alkossd meg a rekurzív **endings** függvényt, ami visszaadja a mágialista végződéseit.

```
endings :: [magic] -> [[magic]]
```

Ez már túl megy a limitjeiden. Nem bírod tovább.

```
endings [1,2,3] == [[1,2,3],[2,3],[3],[]]
endings [] == [[]]
endings [[]] == [[[]],[]]
endings "Haskell" == ["Haskell","askell","skell","kell","ell","ll","l",""]
endings [True] == [[True],[]]
```

...

Az egyik pillanatban még a könyvtárban gyakoroltál, a másikban pedig az ágyadban ébredtél. Barátaidtól megtudod, hogy 3 hónap telt a gyakorlásod óta. A gyakorlás során mindenedet kimerítetted és kómába estél. A barátaid aggódtak érted. Még gyengének érzed magad, de az állapotod már stabil. Hallgatnod kellett volna mesteredre.

5. Házi Feladat

Első találkozás

Típus magyarázó

A teljes olvasási élmény érdekében pár alap típust átneveztünk. Kérlek ezeket vegyétek figyelembe.

```
type Air = Int
type Fire = Integer
type Shadow = Float
type Haskell = Double
type Open = String
type Key a b = [(a,[b])]
```

3 hónap telt el az incidens óta. Az erőd napról-napra egyre több. A gyógyulásod alatt már vissza is álltál edzeni. Nem pihenhetsz, még nem jött el az ideje. Szóbeszédetek alapján a Haskell mágus a környező falvakban portyázik és új követőket gyűjt.

Már megtanultad, hogy jobb, ha mesteredre hallgatsz, így az ő módszerei alapján haladsz. Szerencsére gyorsan tanulsz és már egész komplex feladatokat is kapsz tőle. A fejlődésed egészen figyelemre méltó, de nehogy a rossz emberek figyelmét keltse fel.

A délutáni órákban mestereddal töltöd az időd. Bevezet téged szépen lassan a mágia minden rejtelmébe. A mai edzés első feladata a következő.

Első feladat (Növekvő erő)

Először meg kell állapítanod egy bejövő mágia erejét. Itt mestered folyamatos mágia listákkal fog bombázni, amiről el kell döntened, hogy a mágia ereje alapján növekvő sorrendbe talált el téged. De ne less! Erre a feladatra csak a `mind_vision` függvényt használhatod, ami eldönti, hogy a beérkező mágikus erők növekvő sorrendben vannak.

```
mind_vision :: Ord magic => [magic] -> Bool
```

Most, hogy felkészültél, lássuk mit is tudsz!

```
mind_vision ([] :: [Air])
mind_vision [1::Fire]
mind_vision [1::Air]
mind_vision [1::Shadow]
mind_vision "a"
mind_vision [5,6,9,10]
mind_vision [(-2),(-1),1,9,10,19]
mind_vision "adn"
not (mind_vision [10,9,8,7,6,5,4,3,2,1,0])
not (mind_vision "alma")
not (mind_vision [10,9..])
not (mind_vision ([1..10] ++ [9,8..]))
```

Szép munka!

```
mind_vision [99999 :: Haskell]
```

Áh, ez az utolsó már elég kemény volt! De ez mi volt?

Ezt már nem sikerült kivédened, csak eltérítened. Egy közeli fába csapódott be ez az utolsó lövedék. Ahogy utána nézel a lövedéknek, látod, hogy a becsapódás helye lilán füstöl ... Ez csak egy dolgot jelenthet. **Itt van.**

A füst egyre sűrűbb és sűrűbb lesz, majd egyszer csak azt látod, hogy egy alak kezd feléd közeledni belőle. Amint kiér, látod csapzott haját és ahogy az őrület tükröződik a szeméből. Kék köpenyébe burkolódzva halad feléd, majd egy sátáni kacaj után csak egy lila csóvát láatsz, ami feléd száguld. A csóva kiüt téged és a világ újfent elsötétül.

Második feladat (Menekülés)

Kis idő múltán egy zárt helyen ébredsz. Rácsok mögött vagy, valami cella lehet. Látod, hogy a cellán egy mágikus zár van. Alaposan megvizsgálod a zárat. Ezt a mágikus zárat veszed benne észre:

```
lock :: Key a b -> Open
lock [(x,xs):[y,ys]] = "First"
lock ([_]:[(x,[xs]):[y,ys]:[]]) = "Second"
lock [(x,y:_:[]):[]] = "Third"
```

A zárnak három fázisa van, amit három mágikus kombináció véd. Gyorsan! Ki kell találnod milyen mágiával lehet azt a három zárat feltörni. A feltöréséhez alkossd meg

a `magic_key_1`, `magic_key_2`, `magic_key_3` konstans függvényeket, amiket, ha beillesztesz a `lock` függvénybe, fel tudod törni a zárat.

```
lock magic_key_1 == "First"
lock magic_key_2 == "Second"
lock magic_key_3 == "Third"
```

Ok ez nem volt egyszerű ... még szerencse, hogy annyit gyakoroltad a mágia alapjait, hogy ki tudtál igazodni egy ilyen komplex szerkezeten.

Sajnos a szabadulásod nem ment hang nélkül. Amint kiértél a celládból, a folyosó végéről hallod, hogy valaki közeledik. Amint realizálsz, hogy ez nem más, mint a Haskell mágus, már lövedékek szárnyalnak feléd. Az első elől még éppen el tudsz ugrani, de ez nem mehet a végletekig, védekezned kell!

Harmadik feladat (Védekezés)

A tanulmányaidnak köszönhetően már a látványból is jól ki tudod szűrni a mágia természetét. Látod, hogy a mágus egyszerre több fajta mágiával is támad. Te sajnos egyszerre csak egy féle ellen tudsz védekezni. Használnod kell a `mage_armor` függvényt, ami kiszűri a lövedékek egy megadott fajtát és visszaadja az sorozat után elszenvedett sebzést.

```
mage_armor :: [Missile] -> Type -> Power
```

Ok, lássuk hogyan is működik ez élesben.

```
mage_armor [] "Shadow" == 0
mage_armor [("Shadow", 5)] "Shadow" == 0
```

```
mage_armor [("Fire", 10)] "Shadow" == 10
mage_armor [("Fire", 2), ("Air", 5)] "Fire" == 5
mage_armor [("Air", 3), ("Fire", 8), ("Fire", 100)] "Fire" == 3
mage_armor [("Shadow", 3), ("Shadow", 5), ("Fire", 8), ("Air", 3)] "Shadow" == 11
```

Ez így nem mehet tovább. Túl erős és csak védekezéssel szinte biztos a vég. **Vissza kell támadnod!**

Negyedik feladat (Vissza a feladónak)

Ezek a varázslatok túl erősek számodra. Nem tudsz ekkora erőt megidézni magadtól. Ekkor eszedbe jut. Mi történne, ha nem megidéznél ilyen mágiát, csak visszaküldenéd rá? Alkosd meg a **backfire** függvényt. A függvény a beérkező támadásokat visszaküldi a feladónak, ha annak az ereje kevesebb, mint 50. Út közben a minden feléd tartó ugyanolyan típusú támadás erejét csökkenti annyival, amennyi a visszaküldött támadás értéke. A függvény visszaadja az el nem hárított támadások listáját.

```
backfire :: [Missile] -> [Missile]
```

Ez rázós lesz.

```
backfire [] == []
backfire [("Shadow",49)] == []
backfire [("Shadow", 50)] == [("Shadow", 50)]
backfire [("Shadow", 6), ("Shadow", 56)] == [("Shadow", 50)]
backfire [("Shadow", 6), ("Shadow", 55)] == []
backfire [("Fire", 6), ("Shadow", 50)] == [("Shadow", 50)]
backfire [("Fire", 6), ("Shadow", 5), ("Fire", 55), ("Shadow", 50), ("Air", 51), ("Fire", 110), ("Shadow", 105), ("Air", 5), ("Air", 60), ("Shadow", 99)] == [("Air", 51), ("Fire", 55), ("Shadow", 55), ("Air", 55)]
```

Ahogy mennek az oda-vissza lövedékek, egy portál nyílik meg mögötted. Egy ismerős hangot hallasz belőle:

- Hát te meg mit keresel itt? Gyere már!

Ennyire még sose örültél mestered hangjának. Ahogy csak tudsz, futni kezdesz a portál felé. Szeded a lábaid, de látod, hogy a portál elkezd záródni. Úgy szaladsz, mintha csak az életed múlna rajta (mert amúgy az múlik ...). Végül egy gazellát megszégyenítő ugrással beugrasz a portálba, ami abban a pillanatban bezáródik, ahogy átérsz rajta teljesen. Ez meleg helyzet volt, de túlélted. Jó úton haladsz, hogy le tudd győzni a Haskell mágust. Mestered aggodalmában csak egy dolgot tud tőled kérdezni.

- Mókáztál egyet?

6. Házi Feladat

Bájitalok101

Típus magyarázó

A teljes olvasási élmény érdekében pár alap típust átneveztünk. Kérlek ezeket vegyétek figyelembe.

```
type Crate item = [[item]]
type Recipe item = (item -> Bool)
type Measure item = (item -> item)
type Amount = Int
```

Telnek a napok a képzőben. Nagy nap ez a mai, mivel mestered most először engedi meg neked, hogy az elkezd a tényleges tanulmányaid a bájitalok tudományában. Mindig érdekelt, hogy mit és hogyan lehet összekeverni, a különböző hatások érdekében. A mestered megadott egy terem számot. Azt mondta, hogy ott fog várni, aki majd be tud vezetni ebbe a tudományba.

Belépsz a terembe és ott egy ismerős arc fogad. A szakállas férfi fogad téged, aki a tanulmányaid elején visszafogta mestered a további "feladatoknak" álcázott kínszosztól. Ennek kicsit megörölsz. Reményeid szerint nem lesz vészesebb, mint mestered feladatai. Belépsz a terembe és már egy teljesen előkészített asztal fogad téged, ahol mindenféle mérő eszköz fogad. Látsz magad előtt egy receptes könyvet, felnyitod és elkezded böngészni a recepteket és hatásait. Az asztal mellett látsz több ládát is. Vajon mi lehet bennük?

Első feladat (Recept)

Bemelegítésként egy **recipe** függvényt kell alkotnod, ami megnézi, hogy minden ládában megtalálható-e a recept szerint minden hozzávaló.

```
recipe :: Eq item => Recipe item -> Crate item -> Bool
```

Nézzük akkor azokat a ládákat!

```
recipe (>3) []
recipe (>3) [[]]
recipe (>3) [[],[[]]]
recipe (>3) [[4,4],[5,4]]
not (recipe (>3) [[2]])
recipe (\(a,b) -> even a) [(4,5),(6,7)],[(4,8),(10,3)]
not (recipe (\xs -> length xs > 3) [[1..3],[1..8]],[],[[5..10],[[]]])
```

Remélem nem kezdjük itt is ezeket a béna számológatós feladatokat ... A férfi következőnek a mérő eszközök felé invitál téged, majd elkezd összeöntögetni az ő előtte lévő anyagokat.

Második feladat (Adagolás)

A precizitás nagyon fontos ebben a szakmában, így arra is ráfér a gyakorlás. Alkosd meg a **measure** függvényt, ami az összes ládából kiméri a megfelelő mennyiségű anyagot, majd visszaadja, hogy mennyi anyagunk maradt.

```
measure :: Measure item -> Crate item -> Crate item
```

Ok, akkor most kezdjük el a kimérést, **de óvatosan**.

```
measure (\ x -> x - 1) [] == []
```

```

measure (\ x -> x - 1) [[]] == [[]]
measure (\ x -> x - 1) [[],[[]] == [[],[[]]
measure (\ x -> x - 1) [[1,1],[1,1]] == [[0,0],[0,0]]
measure (\(a,b) -> (a-1,b-1)) [(5,6)] == [(4,5)]
measure (\(x:xs) -> xs) [[[1,2],[1]],[[1]]] == [[[2],[[]],[[]]]]

```

Ez lehet egy laza naphoz elegendő gyakorlás.

Harmadik feladat (Meddig?)

Az előző két gyakorlatból szerzett tudásod alapján, mérd ki annyszor az adagokat, ameddig a receptek engedik. De vigyázz, ne többet egy cseppet se. Erre alkosd meg

a `measure_until` függvényt, ami megadja, hogy hányszor tudod kimérni az adagokat.

```

measure_until :: Eq item => Recipe item -> Measure item -> Crate item {-not empty-} ->
Amount

```

Ok, ez már kicsit izzasztóbb melő.

```

measure_until (>1) (\ x -> x - 1) [[]] == 0
measure_until (>1) (\ x -> x - 1) [[5]] == 4
measure_until (>1) (\ x -> x - 1) [[5,2]] == 1
measure_until (>1) (\ x -> x - 1) [[5],[1]] == 0
measure_until (\(a,b) -> even a) (\(a,b) -> (a-1,b)) [(4,5),(4,1)] == 1
measure_until (\ xs -> not (null xs)) (\(x:xs) -> xs) [[[1,2,3],[1,2,3]],[[1,2,3],[1,2]]]
== 2

```

Hmm, kezdetnek nem is rossz, bár ezek se jobbak, mint mestered feladatai. Még át kell gondolnod, hogy akarod-e folytatni tanulmányaid ezen a területen. Kívülről izgalmasabbnak nézett ki, de lehet ez még csak a kezdet. Lehet adsz ennek a dolognak még egy esélyt.

7. Házi Feladat

Az Ultimatum

Típus magyarázó

A teljes olvasási élmény érdekében pár alap típust átneveztünk. Kérlek ezeket vegyétek figyelembe.

```
type Spell magic = (magic -> (Bool, magic))
type Stabilizer magic = (magic -> magic)
type Crate item = [[item]]
type Checker item = (item -> Bool)
type Amount = Int
type Cool = (Amount -> Bool)
```

- Látom visszajöttél.

A bájitalmestered hangja hallatszik, amint a folyosón feléd és a terem felé közeledik. Úgy döntöttél, adsz ennek a mesterségnek még egy esélyt, sose lehet tudni, mi jöhet ki belőle. Beléptek a terembe és meglepve látod, hogy egy teljes főző felszerelés, alapanyagokkal együtt kikészítve fogad téged. Ezek szerint várta, hogy visszagyere. Odalépsz az asztalodhoz és a felszerelést tanulmányozva észreveszed, hogy ez nem a tegnapi ősdi felszerelésed. Ez más. Ezek profi eszközök, és frissen szedett alapanyagok. Mire készülhet?

A mestertől egy régi tekercset kapsz a kezvedbe. A tekercs címe ULTIMATRUM. Már hallottál róla. Egy bájital, amit már évek óta betiltottak és minden fellelhető receptet, amit csak találtak, elpusztítottak. Vajon honnan van neki ilyenje? Miért tiltották be az elkészítését? Mit csinálhat ez az ital? Túl sok kérdés kavargó a fejedben. Emiatt még bajba is kerülhetsz, de a kíváncsiság felül kerekedik rajtad. Kinyitod a tekercset.

Első feladat (Edzés)

Először elő kell edzeni a tároló edényt. Alkalmaznod kell a megadott függvény mágiáját, de vigyázz, a tároló nem mindig tudja azt befogadni. A függvény első része teszteli ezt. Anennyiben az teljesül, a tároló be tudja fogani a mágiát. Vigyázz! Ha túl sok erő szabadul fel, abból még problémák lehetnek.

```
prepare :: [Spell container] -> container -> container
```

Ok, ha készen állsz, kezdjük is.

```
prepare [] 0 == 0
prepare [(\\a -> (a >= 0, a + 1))] 0 == 1
prepare [(\\a -> (a > 1, a + 1))] 1 == 1
prepare [(\\a -> (a > 0, a + 1)), (\\a -> (a > 2, a + 1))] 1 == 2
prepare [(\\a -> (even a, div a 2)), (\\a -> (False, a + 10000)), (\\a -> (odd a, a * 3))] 2 == 3
prepare [(\\a -> (a == 3, (a ^ 2) `div` 2)), (\\a -> (True, a + 1)), (\\a -> (False, a - 10000000))] 3 == 5
```

Ez most úgy érzem, megteszi, viszont rengeteg energia szabadult fel. Stabilizálni kell a hely mágia szintjét, különben nagy károkat tud okozni.

Második feladat (Stabilizálás)

Egy tárgy mágiáját akkor tudod stabilizálni, ha addig alkalmazod rá a megadott varázslatot, amíg az már nem változtat az erő szintjén.

```
stabilize :: Eq magic => Stabilizer magic -> magic -> magic
```

Gyorsan, mielőtt még elszabadulnak itt a dolgok.

```
stabilize id 500 == 500
stabilize (\a -> a `div` 2) 100 == 0
stabilize (\a -> if a < 100 then a * 2 else a) 49 == 196
stabilize (\a -> a + 100 `div` a) 5 == 101
stabilize (\a -> a + a `mod` 59) 59 == 59
stabilize (\a -> a + a `mod` 8) 118 == 128
```

Fuh, ez közel volt.

Most, hogy az edényt előkészítetted, ideje elkészíteni az italt.

Harmadik feladat (Ultimatum)

Az alapanyagok előtted vannak. Mestered a recept szerint kikészítette őket sorrendbe, ahogy össze kell őket keverned. Viszont vigyáznod kell! A főzet a készítés közben destabilizálódhat. Erre minden fázisban egy megadott figyelő biztosít ellenőrzést. A figyelő jelezni fog, ha a következő adag hozzávaló hozzáadásával a főzet stabilitása meggyengülne. Ebben az esetben, addig kell stabilizálnod, amíg a figyelő szerint biztonságosan hozzá nem tudod adni a következő adagot.

Sok sikert!

```
brew :: (Integral item, Num ultimatum) => Checker ultimatum -> Stabilizer ultimatum ->
Crate item -> ultimatum
```

Most, hogy előkészültél, nézzük meg, sikerül-e.

```
brew (>5) (+1) [] == 0
brew (<5) (\a -> a - 1) [[5]] == 4
brew (<10) (\a -> a - 1) [[1,3,4]] == 8
brew even (\a -> a - 1) [[1,3,4],[100,1,2]] == 110
brew (const True) id [[1,2,3],[4,5,6]] == 21
```

Huh, ez nem volt egyszerű, de kész van.

A mester elismerően bólogat az irányodba. Talán kis meglepődöttség is látszik az arcán. Lehet, nem bízott abban, hogy menni fog?

- Most pedig ideje kipróbálni! - hallatszik mester szájából.

Fenntartásaid vannak az ötlettel kapcsolatban. Azt se tudod, mit csinál ez az ital. Valamiért csak betiltották. Lehet veszélyes, sőt biztosan az. Aggodalommal a szakállas férfira nézel, aki komoran cselekvésre int téged. Ezen a ponton nem sok választásod van és egy kicsit te is ki akarod próbálni. Tudni akarod, miért is mentél végig ezen a sok hűhón.

Ahogy felhőrpinted az italt, egyből érezni kezded a hatását. A mágiád növekszik a testedben. Érzed, hogy pillanatról pillanatra erősebb vagy. A varázserő elárasztja a tested. Elkezd a körülötted lévő energiákat is elnyelni. Ebből baj lesz. Ez már kezd túl sok lenni. Nem tudni, meddig bírja a tested. Érzed, ahogy már az izmaidat feszíti belülről. Ezt valamivel le kell csillapítani.

Negyedik feladat (Mellékhatások)

Csökkentened kell a mágia mennyiségét a testedben. Erre a `cooldown` függvény pont alkalmas. A `cooldown` függvény az egymás után következő azonos energiákból tudja az ismétlődőket felszabadítani, ha azok megfelelő mértékben ismétlődnek.

```
cooldown :: Eq magic => [magic] -> Cool -> [magic]
```

Gyerünk, nem vesztegethatsz sok időt.

```
cooldown [] (>1) == []
cooldown [100] (>1) == [100]
cooldown [100,100] (>1) == [100]
cooldown [100,100,100,100,200] (>1) == [100,200]
cooldown [100,100,100,100,200,200,100,500] (const True) == [100,200,100,500]
cooldown [100,100,100,100,200,200,100,500] (const False) ==
[100,100,100,100,200,200,100,500]
cooldown [100,100,100,200,200,100,100,500] (\a -> a `mod` 3 == 0) ==
[100,200,200,100,100,500]
cooldown [100,100,100,100,200,200,100,100,500,500] even == [100,200,100,500]
```

Már nem bírod sokáig. A mágia kezdi elárasztani az elmédet. A földre rogyasz. A fájdalom elviselhetetlen. Bájjitalmestered felé fordulsz segítségért, viszont amit látsz, az nem segítségnyújtás. Mestered őrült vigyora az utolsó emléked, mielőtt elájulsz a fájdalomtól. Az a vigyor. Azok a szemek. Azok a lilán izzó szemek.

A Haskell mágus csatlósai köztünk vannak.

8. Házi Feladat

- Gondolj!

Hallatszik a hang az elmédben.

- Gondolj!

A hang egyre hangosabb és hangosabb.

- Gondolj!

Felébreds. Egy földön fekszel, legalábbis valami olyasmi lehet. Ahogy körbenézel, csak a nagy üresség fogad. Egy üres dimenzió. Amerre csak látsz, egy fehér vászon fogad. Bepánikolsz.

- Gondolj!

Hangzik el még hangosabban, mint eddig. A hang megdermed, de első gondolatod a mestered jelenléte. Amint ez a kép megjelenik elmédben, előtted terem köpcös-bajszos alakja. Mozdulatlan, élettelen, mint egy báb. Elképzeled, hogy mozog. Az elmédben kialakult mozzanatok, az előtted álló alak is leutánozza. Erre elveszíted a fókuszod és mestered is eltűnik szemed elől.

- Jó. A képesség megvan.

Első feladat (Alapok)

A hang a fejedben arra ösztönöz, hogy kezd egy kicsit egyszerűbb dolgokkal. Mestered képe elég komplex, így nehéz fenntartani a koncentrációt. Képzeletedben hozz létre egy táblát. Erre a táblára írni kell, így szükség van egy **String** paraméterre benne, amit egy **T** fog megalkotni.

```
data Table = ...
```

Most, hogy megalkottad a táblát, hozz létre egy függvényt, amivel egy **String**-ből táblát tudsz csinálni és egy másikat, amivel a tábla szövegét meg lehet változtatni.

```
to_table :: String -> Table
```

```
change :: Table -> String -> Table
```

Ezt egyszerűbb elképzelni, de működik-e?

```
to_table "Magic" == T "Magic"
```

```
to_table "Air" == T "Air"
```

```
to_table "" == T ""
```

```
change (T "Magic") "Air" == T "Air"
```

```
change (T "Air") "Magic" == T "Magic"
```

```
change (T "Magic") "" == T ""
```

- Remek!

Dicsér téged benső hangod.

Második feladat (Ütés)

Képzeleteddel hozz létre egy bábut. Ennek lehet két állapota. A bábu élhet, ebben az esetben **Alive** alkotja meg és tartalmazza az életerejének értékét, de a bábu lehet halott is, ebben az esetben csak a **Dead** fogja őt jellemezni.

```
data Dummy = ...
```

Most, hogy megvan a bábu, hozzunk létre egy függvényt, amivel meg lehet támadni a bábut. Amennyiben az életereje nullára, vagy kevesebb lenne, meghal.

```
hit :: Dummy -> Int -> Dummy
```

Támadjuk meg vele!

```
hit (Alive 5) 0 == Alive 5
hit (Alive 5) 2 == Alive 3
hit (Alive 3) 2 == Alive 1
hit (Alive 1) 1 == Dead
hit Dead 2 == Dead
```

Harmadik feladat (Üssük mindet)

Ezek a bábuk jó gyakorlásnak bizonyulnak. Hozz létre egy függvényt, ami több bábut is meg tud ütni egyszerre.

```
multi_hit :: [Dummy] -> Int -> [Dummy]
```

Teszteljük le!

```
multi_hit [] 5 == []
multi_hit [Alive 5] 5 == [Dead]
multi_hit [Alive 5, Alive 4, Alive 3] 4 == [Alive 1, Dead, Dead]
multi_hit [Alive 5, Alive 6, Dead] 3 == [Alive 2, Alive 3, Dead]
```

Negyedik feladat (Mega ütés)

Ezt már lassan még élvezed is. Csinálj egy olyan függvényt, ami megüti az első bábut. Amennyiben az ebbe belehal, az erő túlcserdülő része tovább száll a következőre.

```
megahit :: [Dummy] -> Int -> [Dummy]
```

Na még egyszer támadjuk meg őket!

```
megahit [] 5 == []
megahit [Alive 3] 2 == [Alive 1]
megahit [Alive 3] 5 == [Dead]
megahit [Alive 3, Alive 2] 5 == [Dead, Dead]
megahit [Alive 3, Alive 2] 3 == [Dead, Alive 2]
megahit [Alive 3, Alive 2] 4 == [Dead, Alive 1]
megahit [Alive 5, Alive 2, Alive 1, Alive 6] 10 == [Dead, Dead, Dead, Alive 4]
```

- Remek! Az első feladatok jól mennek. Most pihenj és készülj rá a következőkre. Azok már nem lesznek ilyen egyszerűek.