

20. LECKE / VÁLTOZÓK, ÉS LISTÁK ÖSSZEFOGLALÁS

21.) Logikai feladat:

Az arányásó át akarja juttatni a folyó túl partjára az ott várakozó feleségének az aranyrögöt. A folyón dolgozik egy révész, aki minden mozdíthatót ellop. Az arányásónak van egy ládikája, egy lakatja, és egy hozzátartozó kulcsa. És a feleségének is van egy lakatja egy hozzá tartozó kulccsal. A láda az egyetlen, ami túl nehéz ahhoz, hogy a révész ellopja. Segítsünk az arányásónak átjuttatni az aranyrögöt anélkül, hogy ő vagy a felesége átkelne a folyón.

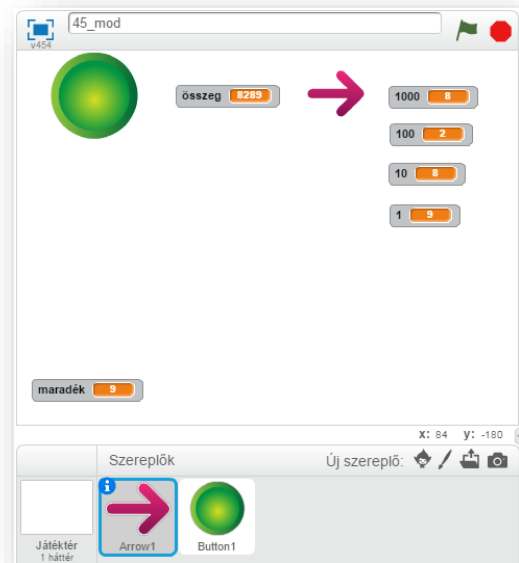


Megoldás röviden: _____

24.) Gyakorlat

A programban generálunk egy 1 és 1000 közötti véletlen számot, majd megnézzük, hogy hány darab ezres, százás tizes, és egyesből áll!

- A program neve legyen 45_mod!
- Hat változóra, és két szereplőre lesz szükségünk a feladat megoldásához!
- A Button1 és az Arrow1 szereplőket olvasd be, és hozd létre az „összeg”, „1000”, „100”, „10”, „1”, és a „maradék” nevű változókat!
- A program a zászlóra kattintással induljon, és nullázzuk le az összes változót!
- A „Button1”-el generáljuk a véletlen számot! Tehát az „összeg” változó legyen egyenlő a véletlen számmal!
- AZ „Arrow1”-re kattintva bontsa fel az „összeg” változóban lévő számot!
- Először azt kell megnéznünk, hogy hány darab ezres van a generált számban! Ezt úgy tudjuk elérni, hogy elosztjuk az „összeg”-et 1000-rel. és a kapott számot lefelé kerekítjük egészre!
Pl.: összeg: 6475 \rightarrow $6475/1000=6,475 \rightarrow$ lefelé ker.= 6
- Tehát ezt a számot beletesszük az „1000”-be!
- Viszont, hogy tovább tudjunk menni, ahhoz szükségünk van a „maradék”-ra, amit úgy kapunk meg, hogy a mod parancsot használjuk! Tehát a „maradék”-ot egyenlővé tesszük az „összeg” mod 1000-rel!
pl.: $\text{maradék} := 6475 \text{ mod } 1000 \rightarrow \text{maradék} = 465$
- Innentől fogva a logika ugyan az a változót egyenlővé tesszük az egész résszel, majd a maradékot kiszámoljuk, és azzal megyünk tovább!
A legvégén, amikor már az „1”-es változónál megkaptuk a maradékot, ott csak egyszerűen beletesszük a változóba!
- Végül teszteld, és mentsd a megadott helyre!



25.) Gyakorlat

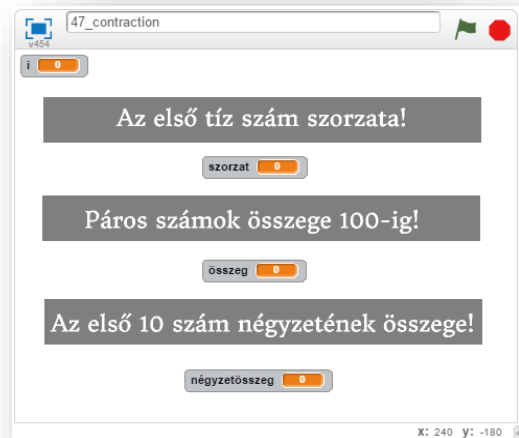
Ebben a feladatban egy változóba kell generálni egy véletlen számot 1 és 100 között, majd ha úgy gondoljuk (mondjuk páros, vagy páratlan, vagy prím szám), akkor hozzáadjuk, ha nem, akkor generálunk egy másikat!

- A program neve legyen: 46_storage_unit!
- Hozzuk létre a szükséges változókat és listát!
- A lista neve legyen „tároló”!
- Kell egy „bevitel” nevű változó, amibe generáljuk a véletlen számot!
- Szükségünk lesz a szokásos „i” változóra, melyben a lista aktuális helyét tudjuk léptetni!
- Két szereplőre is szükségünk lesz, melyek nyomógombok! A minta szerint!
- A program zászlóra kattintással induljon!
- Ürítsük ki a listát, és nullázzuk le az összes változót!
- A „Generálás” szereplőre kattintáskor tegyünk a bevitel változóba egy és száz közötti véletlen számot!
- A „Hozzáadás” szereplőre kattintáskor növeljük az „i” értékét egyel, azért, hogy utána a „tároló” lista „i”-dik helyére beszúrhasuk a „bevitel” változóban látó számot!
- (Ennek a feladatrésznek van egy egyszerűbb megoldása is, de a későbbi programok miatt itt így oldjuk ezt meg! Egyébként „i” nélkül a „Hozzáadás” gombra kattintáskor csak ezt az egy sort kell megadni!)

**21.)Önálló feladat:**

Ebben a feladatban három részfeladatot kell megoldani!

- A program neve legyen: 47_contraction!
- A három „gomb” szereplőt, és a hozzá szükséges változókat hozd létre a minta szerint!
- A változók legyenek: i, szorzat, összeg, és négyzetösszeg!
- A gombokat lásd el a feliratokkal, és írd meg a hozzájuk tartozó algoritmust!
- Minden egyéb kérdésben szabadkezet kapsz!
- Ha végeztél mentsd a megadott helyre a programfájl!



Ismertető, összegző: Egy program megírásának lépései:

- Először elolvasva a feladatot értelmezzük, és végiggondoljuk, hogy tulajdonképpen mi is a dolgunk!
- Létrehozuk a szükséges változókat, listákat, szereplőket, háttereket! (A későbbiek során bővíthetjük, ha szükséges!)
- Logikusan, lépésről lépésre, az elejétől feladatokat adunk a szereplőknek, változókat, listákat felhasználva algoritmusokat, ciklusokat készítünk!
- A program írása során teszteljük a részprogramokat, javítjuk az esetleges hibákat, mentünk!