

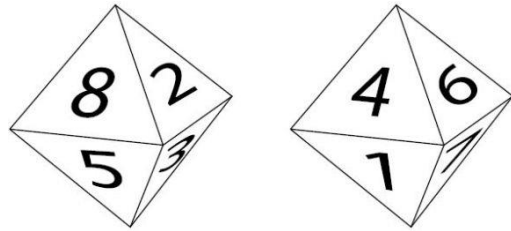
## 29. LECKE / SEGÉD- ÉS ALKALMAZÓI PROGRAMOK KÉSZÍTÉSE SCRATCH-BEN

## 30.) Logikai feladat:

A képen egy test elejét és hátulját látod.


Melyik szám van az 5-el szemben átlósan? \_\_\_\_\_

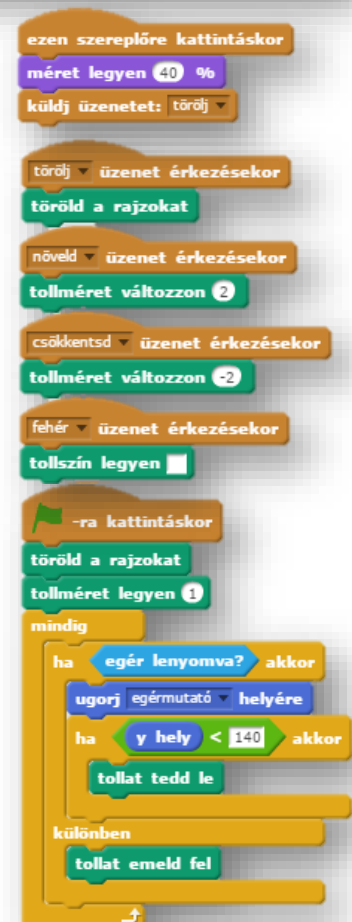
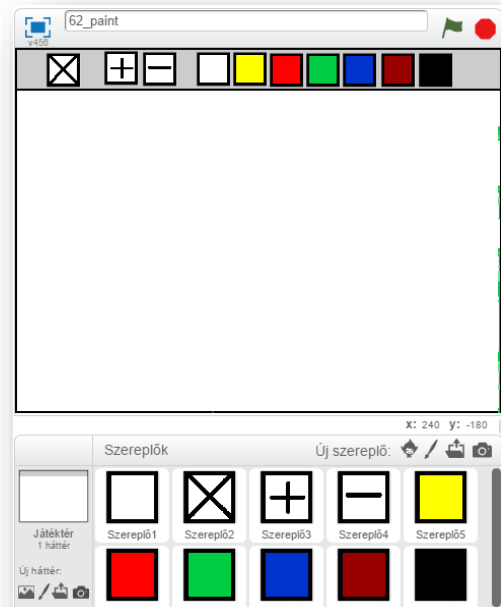
Melyik szám van az 2-vel szemben átlósan? \_\_\_\_\_



## 35.) Gyakorlati feladat:

Ebben a feladatban egy egyszerű rajzoló programot fogunk készíteni! A funkciók, amelyet tudnia kell a programnak: törölni az előző rajzokat; a rajzoló „toll” vastagságát növelni, illetve csökkenteni; különböző alapszínek kiválaszthatósága!

- A program neve legyen: 62\_paint!
- Egy háttérre lesz szükségünk. Rajzoljunk egy vékony fekete keretet a játéktérnek, és a felső sávra húzzunk egy fekete egyenes vonalat (kb. y:220 kp)! Színezzük be világos szürkére a felső sávot! A háttér neve legyen: alap!
- 12 szereplőt kell létrehozunk! Először „alapgomb” néven egy fehér nézetet, fekete szegéllyel hozunk létre! Ezt fogjuk duplikálni tízszer!  Mindegyiknek külön nevet adunk pl.: törlés, növel, csökkent, sárga, piros, ..., fekete! Aztán megrajzoljuk a „gombok” egyéb vonalaikat, színeit! (Ha végeztünk a szereplőkkel, akkor beállítjuk, hogy zászlóra kattintáskor az „alapgomb” szereplő tűnjön el, hiszen erre nem lesz szükségünk!)
- A tizenkettedik szereplő, egy kis fekete kör, amivel rajzolni fogunk! A neve legyen: tollhegy! Ehhez a szereplőhöz fogjuk majd a program törzsét elkészíteni!
- De előtte az összes többi szereplőnek megadjuk, hogy ha rákattintunk a „gombra”, mekkora legyen a mérete %-ban, hogy elérjen a felső sávra! Majd felhelyezzük a helyére!
- Ebben a programrészletben küldjön egy gombra utaló üzenetet! Például: törölj, növelj, csökkent, fehér, sárga, ... , fekete! Ezt tízszer megtesszük!
- A tollhegy szereplőnél a zászlóra kattintással először töröljük az eddigi rajzokat, és a toll méretét állítsuk 1 –re!
- Az összes üzenet érkezésekor beállítjuk a teendőket! „Törölj” üzenet érkezésekor, letisztítja a rajfelületet! A „növel” üzenet érkezésekor a tollméret változzon 2 kp-al, stb.!
- Majd a program fő részénél „mindig” algoritmus futása közben vizsgálja, ha az egér le van nyomva, ugorjon az egérmutató helyére és tegye le a tollat! Ha felengedjük az egeret, akkor emelje fel a tollat! Csak akkor tudjunk rajzolni, ha a felső sáv alatt van az egér!
- Programozás közben teszteljük és mentjük!



**30.) Önálló feladat:**

Ebben a feladatban egy egyszerű „számológép”-et kell készíteni a következő leírás alapján! Csak két számmal végzett műveletet kell megoldani a programmal!

- A program neve legyen 63\_calculator!
- Először a háttérrel kell elkészítened, amely egy egyszerű sötétszürke téglalap, melyen egy világosszürke kisebb téglalap van a tetején a minta szerint!
- 18 darab szereplőt kell létrehoznod! (1-9, a 0 számokat; a műveleti jeleket -,+,\*,/; az egyenlőség jelet; és a „C” törlő gombot)
- Szükség lesz legalább 4 változóra! („1\_szám”; „2\_szám”; „művelet”; és az „eredmény” változókra)
- A számokat úgy kell bevinni, hogy ha rákattintunk egy szereplőre (pl.: 2), akkor küldjön üzenetet!
- Ha érkezik az előbbi üzenet, akkor vizsgálja, hogy melyik számnál vagyunk, és az „együtt \_\_\_” parancsot használjuk az „összefűzéshez”! Ezt minden gombra el kell készítenünk! (Ezeket elkészítheted a háttér feladatainál például!)
- A „C” gombra kattintásnál minden változót töröljön, illetve nullázzon le!
- Az egyenlőség jelre kattintással hozza működésbe a törzsprogramot, melyben „ha / akkor / különben” parancsokkal vizsgálja a műveleti jeleket, és számolja ki a megoldásokat!
- A hibajelzéseket is kezelve, ha túl nagy számot, számokat nem tud kezelni, és hibajelzést küld a program (Infinity), akkor írja ki, hogy: „Próbáld újra!”
- Minden kérdéses egyéb esetben szabadkezet kapsz!
- A programot folyamatos teszteléssel készítsd, és közben ments, nehogy elveszzen a munkád, ha probléma adódik!

