



XII. Robotprogramozó Országos Csapatverseny

1. forduló – II. kategória (6. évfolyam)

A feladat sorszáma után zárójelben szerepel a megszerezhető pontszám, amely két részből áll. Az első szám a tesztelés során a robot látható működése alapján adott pontszám, a második szám a forráskód alapján a zsűri által adott pontszám. Ha a feladat megoldására kapott pontszám eléri a maximális 50%-át, akkor a forráskód hossza alapján további plusz pontok is kaphatók. Ennek részletei a versenykiírásban található.

1. feladat (6+12 pont)

Írj programot, amelyet a robot végrehajtva fájlban tárolt adatok alapján megjeleníti a képernyőn a szükséges információkat!

A feladat megoldásához két fájl áll rendelkezésre. Az *eur.rtf* (*eur.txt*) fájlban egy év havi Forint-Euró átlag árfolyamai szerepelnek. Összesen 12 adat, a hónapok szerinti sorrendben. A *honapok.rtf* (*honapok.txt*) fájlban pedig a hónapok nevei (ékezetmentes formában). A fájlok *rtf* és *txt* változatainak ugyanaz a tartalma.

A feladat, hogy a fájlok adatait felhasználva, jelenítsd meg a képernyőn a legmagasabb Forint-Euró árfolyamot és azt is, hogy ez melyik hónapban volt. A két információ a minta szerint jelenjen meg a képernyőn. Az adatok ütközésérzékelő megnyomásáig látszódnak!

A program megírásakor a képernyőn megjelenő változó tartalmú adatokat a fájlokból kell venni, tehát mindkét fájlt használni kell a feladat megoldásához.

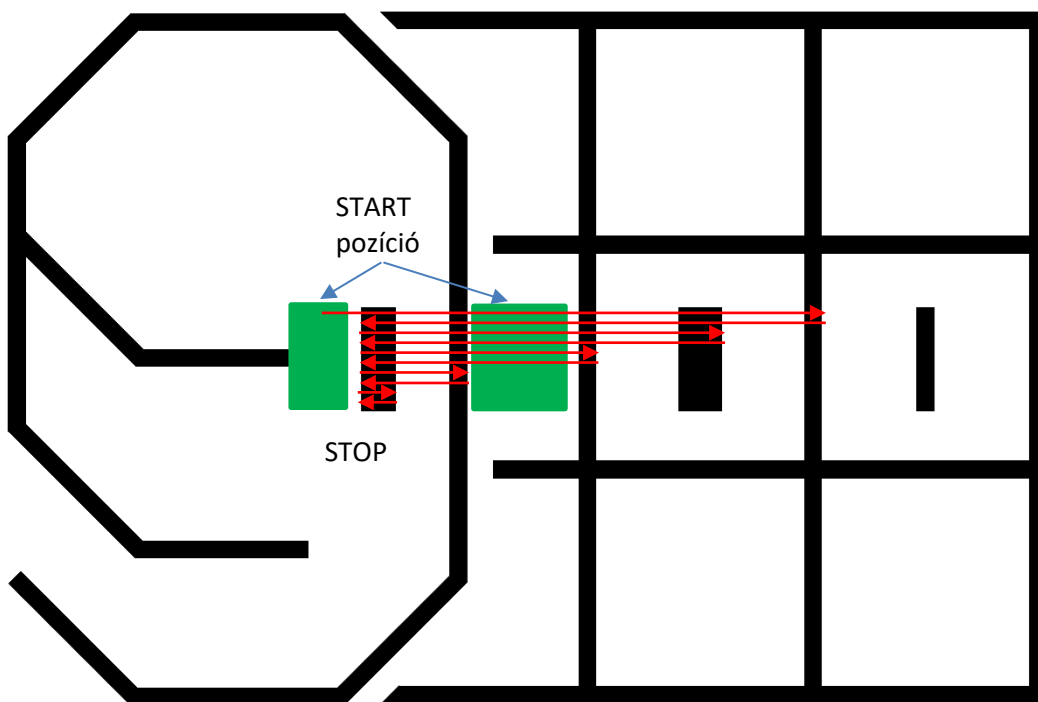
Minta (a mintán szereplő adatok nem a fájlban tárolt adatok alapján készültek)

```
Max.: 345.12  
Honap: junius
```

2. feladat (17+10 pont)

Írj programot, amelyet a robot végrehajtva megadott feltételek szerint mozog! A robot a startpozícióból indul és egyenesen előre halad az útvonalát merőlegesen keresztező fekete színű vonalak fölött. Az 5. vonalon történő áthaladás után megáll, majd tolatni kezd. Hátrafelé haladva is áthalad 5 vonalon, majd az ötödik után ismét előre indul, de most csak 4 vonalon halad keresztül. A negyedik vonalon történő áthaladás után ismét tolatni kezd. 4 vonalat tolat hátra és ismét előre indul. Mindezt ismétli 3 majd 2 és végül 1 vonalra. (Lásd az ábrát!) Az utolsó lépésben, miután egy vonalnyi tolatott megáll és befejezi programját.

A robot indulási pozíciója a zölddel jelzett területen belül bárhol lehet. (A robot 3. portra kötött fény szenzora van a zöld színű terület fölött.) A robot keleti irányba néz. A robot indulási irányát a nyíl jelzi. Induláskor a robot fény szenzora fehér színt érzékel.



3. feladat (19+10 pont)

Írj programot, amelyet a robot végrehajtva a koordináta-rendszer egy pontjához mozog, majd visszatér az origóhoz!

A robot startpozícióból indul egyenesen előre és 3-as portra kötött fény szenzorával megszámlolja a haladási útvonalára merőleges fekete színű vonalakat, amelyeken áthalad. A mozgását akadálytól 15 cm-re fejezi be. A képernyőre írja ki, hogy hány fekete vonalon haladt át!

Ezután a robotot át kell helyezni a koordináta-rendszer kezdőpontjába (lásd ábra), majd ütközésérzékelő megnyomására a robotnak abba a koordinátájú pontba kell mozognia, amennyi vonalat számolt az első mozgása során. Tehát például, ha a robot 3 fekete vonal fölött haladt át, amíg az akadályt nem észlelte, akkor most a koordináta-rendszer (3; 3) koordinátájú pontjába kell útvonalkövetéssel mozognia. (Ha két vonalon haladt át, akkor a (2; 2) koordinátájú pontba, ha pedig 1 vonalon haladt át, akkor az (1; 1) koordinátájú pontba.) A pontot elérve álljon meg, adjon tetszőleges hangjelzést 1 másodpercig, majd útvonalkövetéssel mozogjon vissza a koordináta-rendszer origójába (kb. az indulási pozíciójába: ± 8 cm pontossággal)!

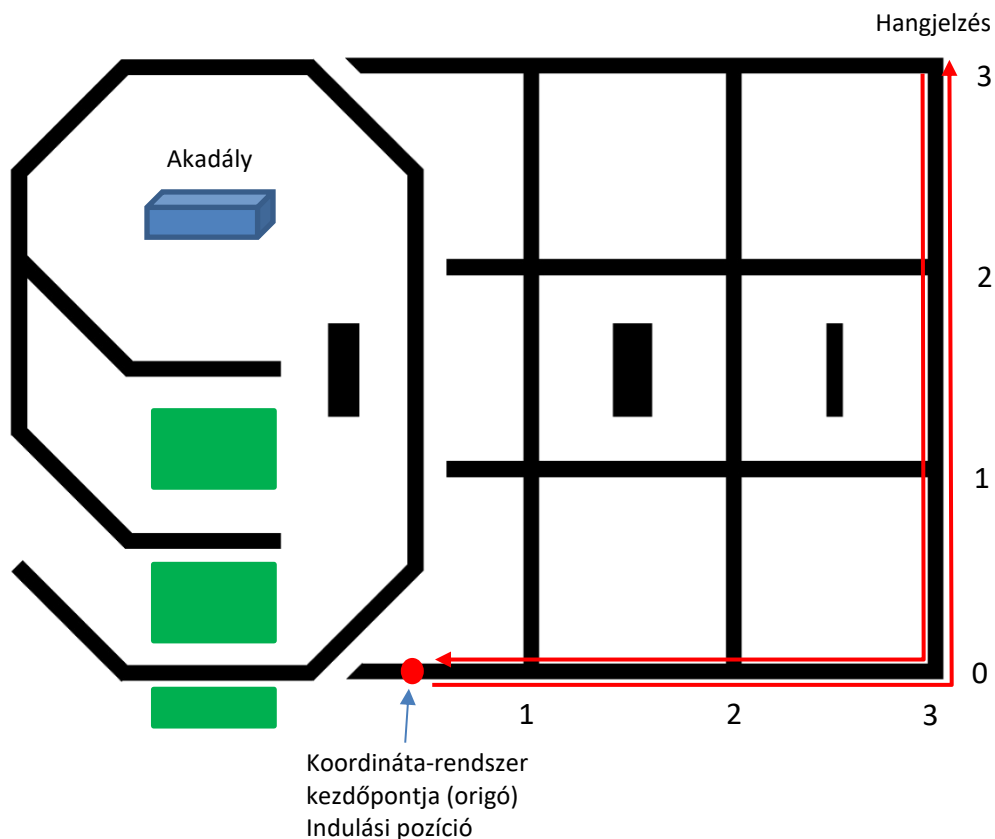
A koordináta-rendszerben a robot útvonalkövetéssel mozoghat. A célhoz és a visszafelé történő mozgása során nem kell ugyanazon az útvonalon haladnia. Az indulási pozíciót elérve nem kell megállnia a robotnak.

A programot kétszer kell bemutatni, más-más leolvasott vonalszámok esetére.

Az ábrán a startpozíciók zölddel van jelölve. A robot 3-a fény szenzora van a zölddel jelölt terület fölött és a robot észak felé néz. Induláskor a robot fény szenzora fehér színt érzékel.

A koordináta-rendszeren belüli mozgás indulási pozícióját piros kör jelzi az ábrán. A robotot úgy kell lehelyezni a pályára, hogy a két fény szenzora a fekete vonal két oldalán helyezkedjen el (fehér felületek fölött) és a két fény szenzor között legyen az origó (az ábrán piros kör). A robot kelet felé néz.

Pl.: Ha a robot 3 vonal fölött haladt át, akkor a cél koordinátája (3; 3). A robot egy lehetséges mozgását a piros nyilak jelzik.



4. feladat (16+10 pont)

Írj programot, amelyet a robot végrehajtva a robot véletlen bolyongással megkeres egy fekete vonalat, majd amentén kezd mozogni!

A robot a starterületen belül kezdi meg mozgást, sorsolt számoknak megfelelő sebességgel. Sorsoljon a robot két véletlen számot 0 és +30 között. A két véletlen számot használja fel a motorjai sebességparamétereként! Tehát az egyik sorsolt szám legyen a B portra csatlakozó motor, a másik véletlen szám a C portra csatlakozó motor sebessége. A sorsolt számokkal a mozgást 0,5 másodpercig végezze, majd az idő letelte után sorsoljon újra és az új számokat használja sebességként! Mindezt addig ismétlje, amíg valamelyik fényszenzora fekete vonalat nem érzékel! Ekkor álljon meg és várakozzon 1 másodpercig!

A várakozás után kezdje el követni a fekete vonalat, abba az irányba, amelyik fényszenzora először meglátta azt. Tehát, ha a véletlen bolyongás során a robot bal oldali szenzorával találta meg a vonalat, akkor balra kezdje azt követni, ha a jobb oldali szenzorával látta meg először, akkor jobbra kezdje követni. (Ha a két szenzora egyszerre látná meg a vonalat (kizárt 😊), akkor mindegy, hogy melyik irányban követi.)

Az útvonalkövetést kikapcsolásig ismétli a robot (nem kell külön programból megállítani).

A robot az ábrán zölddel jelzett területen belülről indul (fényszenzora a zölddel jelzett területen belül van) és lehetséges indulási irányait az ábrán pirossal jelzett nyilak szemléltetik.

A programot kétszer kell bemutatni, ugyanolyan indulási pozícióból, de eltérő indulási iránnyal. A véletlen bolyongás miatt az útvonala eltérő lesz.

