

Huszár

Nemes Tihamér OITV 2009. 2. forduló 2. korcsoport 3. feladat

Egy sakktáblára elhelyezünk egy huszárt. A sakktábla 8×8-as négyzet. A huszár „lógrásban” lép, azaz vagy vízszintes irányban lép egyet és függőlegesen kettőt, vagy pedig fordítva.

Készíts programot *huszar* néven, amely egy adott pozícióra elhelyezett huszár esetén megadja, hogy a huszár legfeljebb N lépés alatt mely pozíciókra juthat el!

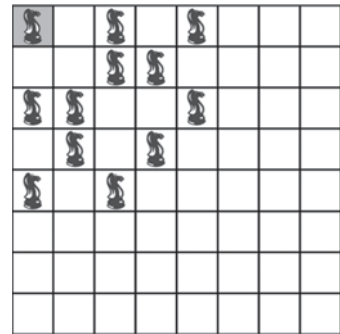
A *huszar.be* szöveges állomány egyetlen sorában a huszár sorindexe ($1 \leq \text{SOR} \leq 8$) és oszlopindexe ($1 \leq \text{OSZLOP} \leq 8$), valamint a lépések maximális száma ($1 \leq N \leq 6$) van, egy-egy szóközzel elválasztva.

A *huszar.ki* szöveges állományba 8 sort kell írni, mindegyikben pontosan 8 karakter legyen! Az i -edik sor j -edik oszlopába $*$ -ot kell kiírni, ha a huszár ezt a pozíciót legfeljebb N lépés alatt elérheti, egyébként pedig szóközt!

Példa:

```
huszar.be
1 1 2
```

```
huszar.ki
*.*.*...
..**....
**..*...
.*.*....
*.*.....
.....
.....
.....
```



Megoldás

A sakktáblát a *Tábla* tömbben tároljuk:

KONSTANS Méret MINT Egész = 8

VÁLTOZÓ Tábla(1...Méret, 1...Méret) MINT Karakter

Készítsünk rekurzív eljárást, amely az argumentumként megadott aktuális mezőre elhelyez egy csillagot, majd minden olyan mezőre meghívja saját magát, ahová a huszár továbbléphet!

Egy rekurzív hívássorozat akkor ér véget, ha a huszár kilépne a sakktábláról, illetve ha elérjük a maximális lépésszámot (báziskritériumok). A lépésszám követéséhez az eljárás paramétereire közé a *Sor* és *Oszlop* koordináták mellett felvesszük a *Lépés* paramétert is.

ELJÁRÁS Lép(Sor, Oszlop, Lépés MINT Egész)

A rekurzív hívásokat ciklusba foglalhatjuk, ha a *Dsor*, illetve *Doszlop* tömbökben feljegyezzük, hogy egy lépésnél mennyivel változhat a sor és az oszlop indexe. Az ábra alapján, az óramutató járásával megegyező irányban sorra véve a lépéseket:

VÁLTOZÓ Dsor(1...8) MINT Egész = (-2, -2, -1, 1, 2, 2, 1, -1)

VÁLTOZÓ Doszlop(1...8) MINT Egész = (-1, 1, 2, 2, 1, -1, -2, -2)

	-2		-2	
	-1		+1	
-1				-1
-2				2
		0		
		0		
1				1
-2				2
	2		2	
	-1		1	

A sor- és oszlopindexek változása a huszár lépéseinél

Az eljárás algoritmus:

ELJÁRÁS Lép(Sor, Oszlop, Lépés MINT Egész)

HA Sor \geq 1 ÉS# Sor \leq Méret ÉS# Oszlop \geq 1 ÉS# Oszlop \leq Méret ÉS# Lépés \leq N AKKOR

Tábla(Sor, Oszlop) = "*" "

CIKLUS I=1-től 8-ig

Lép(Sor+Dsor(I), Oszlop+Doszlop(I), Lépés+1)

CIKLUS VÉGE

ELÁGAZÁS VÉGE

ELJÁRÁS VÉGE

Célszerű vagy kötelező volt-e logikai rövidzárát alkalmazni az elágazás feltételében? Miért növeltük meg a *Lépés* értékét a rekurzív hívásoknál?

A főprogramban a beolvasás és a *Tábla* tömb inicializálása után meghívjuk az eljárást:

VÁLTOZÓ Sor, Oszlop, N MINT Egész

Be: Sor, Oszlop, N

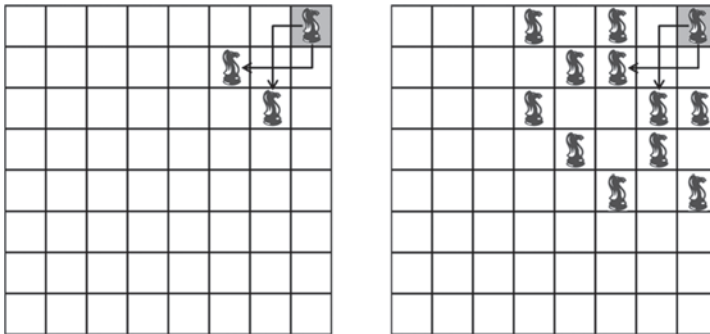
Tábla(,) = " "

Lép(Sor, Oszlop, 0)

A teljes megoldást a kiírással együtt a *huszar* program tartalmazza.

Kövessük végig papíron, ceruzával a huszár lépéseit a fenti algoritmus alapján! Jegyezzük fel, hogy az egyes mezőket hány lépésben érte el! Vegyük észre, hogy lényegében egy gráf mélységi bejárását végeztük!

Megoldásunk meglehetősen rossz hatásfokú. Nagyobb táblánál vagy a maximális lépésszám nagyobb értékénél könnyen vezet kivárthatatlan hosszúságú futásidőhöz. Ennek oka az, hogy a rekurzív hívásokat akkor is folytatnunk kell, ha esetleg már egy csillagot tartalmazó mezőhöz érkeztünk. Előfordulhat ugyanis, hogy ezt a mezőt először magasabb lépésszámmal érjük el, majd egy másik irányból alacsonyabb lépésszámmal ugyanide jutunk. Ebben az esetben több lépéssel folytathatjuk a bejárást, így innen továbbhaladva több mezőt érhetünk el.



Ha az utolsó előtti lépésben érkeztünk a szürkével jelölt mezőre, akkor legfeljebb két további mezőt érhetünk el. Ha egy lépéssel korábban érkeztünk, akkor 11 további mezőre is léphetünk.

Javítsunk az algoritmus hatékonyságán! Először jelöljük meg az egy lépésben elérhető mezőket, majd azokat, ahová két lépésben jutunk el stb. (szélességi bejárás)! A program elkészítését az Olvasóra bízunk. ☺

Javasoljuk az Olvasónak, hogy oldja meg a 2009-es OKTV 2. fordulójának nagyon hasonló, 3. feladatát (*Sakk*).