

Tartalomjegyzék

- 1 Feladatok
 - ◆ 1.1
Pascal-háromszög
 - ◆ 1.2
Huszárok
 - ◆ 1.3 Zárt terület
kifestése

Feladatok

Pascal-háromszög

Írjunk egy függvényt! Bemenete egy egész szám, **n**, kimenete pedig a Pascal-háromszög n-edik sora (lista).

```
def pascal(n):
    if n == 1:
        return [1]
    if n == 2:
        return [1,1]
    else:
        previous = [1,1]
        for i in range(2,n):
            actual = [1]
            for i in range(1,len(previous)):
                actual.append(previous[i]+previous[i-1])
            actual.append(1)
            previous = actual
        return previous
print pascal(14)
```

Huszárok

Áll a sakktáblán egy huszár. Számoljuk ki a sakktábla minden mez?jére, hogy legkevesebb hány lépéssel tudunk eljutni oda az el?bbi huszárral! A feladat megoldásához írjunk egy függvényt, **knight(x,y)**, amelynek két bemenete a huszár táblán elfoglalt helyének koordinátái. A visszatérési érték legyen egy nyolcszor nyolcas lista! Hsználjunk dinamikus programozást!

```
board = [[-1]* 8 for i in range(8)]

def fill(x,y):
    ret = [[x+dx,y+dy] for dx in [2,-2] for dy in [1,-1]]
    ret += [[x+dx,y+dy] for dx in [1,-1] for dy in [2,-2]]
    ret = [r for r in ret if r[0] >= 0 and r[1] >= 0 and r[0] < 8 and r[1] < 8]
    return ret

print fill(1,4)

found = True
n = 0
board[0][0] = 0

while found:
    found = False
    for y in range(len(board)):
        for x in range(len(board[y])):
```

```

    if board[y][x] == -1:
        found = True
    if board[y][x] == n:
        for [yf,xf] in fill(y,x):
            if board[yf][xf] == -1:
                board[yf][xf] = n + 1

    n += 1
for i in range(7,-1,-1):
    print board[i]

```

Zárt terület kifestése

Olvassuk be az alábbi "http://wiki.math.bme.huszöveget"http://wiki.math.bme.hu listák listájába (minden karakter egy elem): mentsük le egy fájlba kézzel, majd nyissuk meg a fájlt Pythonnal! Írjunk egy fill(x,y) függvényt, ami ugyanazt csinálja, mint a Paint kitölt? funkciója! Az (x,y) pontból kiindulva a . helyére # jelet tesz, amíg a # jel által jelölt falba nem ütközik! A módszer rekurzív: kifestjük az (x,y) pontot, majd a szomszédait, ha azok nem # jelek. Hívjuk meg a szomszédokra (akik nem # jelek) a függvényt rekurzívan. Ha nincs kit kiszínezni, akkor álljunk meg!

```

.....
...#####.....
...#.....#.....
...#.....#.....
...#.....#.....
...#.....#.....
...#.....#.....
...#.....#####.....
...###.....##.....#.....
...#...##.....##.....#.....
...#.....##.....##.....#.....
...#.....##.....##.....#.....
...#.....#####.....
...#.....#.....#.....
...#.....##.....#.....
...#.....##.....#.....
...#.....##.....#.....
...#.....##.....#.....
...#####.....
.....
.....
.....
.....
.....

```

```

picture = []
with open('abra.txt') as f:
    for l in f:
        L = []
        for c in l:
            L.append(c)
        picture.append(L)

def color(x,y,picture):
    if y < len(picture):
        if x < len(picture[y]):
            if picture[y][x] == ".":
                picture[y][x] = "#"
                color(x+1,y,picture)
                color(x-1,y,picture)
                color(x,y+1,picture)
                color(x,y-1,picture)
color(2,2,picture)
for p in picture:

```

```
print "".join(p),
```