

## TEST Z8154 Programování v geoinformatice (podzim 2013)

odovzdajte do: <https://is.muni.cz/auth/el/1431/podzim2013/Z8154/ode/45388234/>

1.) Navrhnite funkciu **test()**, ktorá otestuje či zadané pole obsahuje iba číselné hodnoty (tzn. dátový typ *int* alebo *float*). Ak áno, vráti hodnotu TRUE, inak vráti FALSE

```
def test(pole):
    for i in pole:
        if not isinstance(i, (int, float)):
            break
    else: return True
    return False
```

```
def test(pole):
    try:
        float(pole)
        print "TRUE"
    except:
        print "FALSE"
```

```
def test(pole):
    for i in pole:
        if not(isinstance(i, int) or isinstance(i, float)):
            return False
    return True
```

2.) Navrhnite funkciu **diff()**, ktorá spracuje pole číselných hodnôt tak, že porovná susedné záznamy a vypíše najvyšší rozdiel medzi susedmi. Napr., výsledok zo zoznamu (2, 8, 5, 7) by bol 6. Funkcia porovnáva len susedné hodnoty, teda nie všetky hodnoty navzájom.

```
def diff(pole):
    diff = 0
    for i in range(len(pole)-1):
        d = abs(pole[i] - pole[i+1])
        if (d > diff):
            diff = d
    return diff
```

3.) Jednou vetou popíšte, čím sa líši dátový typ **tuple** od klasického poľa. **To ste všetci vedeli :)**

4.) Napíšte funkciu **zet()**, ktorá na základe zadaného čísla vykreslí s použitím základných znakov "štvorec". Do štvorca bude vpísaný symbol **Z** pomocou znakov "#", ostatné znaky budú "\*", vid' príklad:

```
def zet(n):
    print n * "#"
    for i in range(n-2):
        print (n-(2+i)) * "*" + "#" + (1+i) * "*"
    print n * "#"
```

```

def zet(n):
    print n*"# "
    for i in range (n-2):
        for j in range (n):
            if (j == (n - i-2)):
                print "#",
            else: print "*",
        print
    print n*"# "

def zet(a):
    for i in range(a):
        if (i == 0):
            print a * "#"
        elif (i == a-1):
            print a * "#"
        else:
            print (a-1-i) * "*" + "#" + i * "*"

```

zet(5)

```

# # # # #
* * * # *
* * # * *
* # * * *
# # # # #

```

5.) Funkcia **area\_diff()** vypočíta na základe vstupného polygónu rozdiel medzi plochou minimálneho ohraničujúceho obdĺžnika (MBR alebo envelope) a plochou konvexnej obálky (convex hull).

```
import arcpy
```

```

def area_diff(polygon):
    mbr = arcpy.MinimumBoundingGeometry_management(polygon, output, "ENVELOPE")
    hull = arcpy.MinimumBoundingGeometry_management(polygon, output,
"CONVEX_HULL")
    return (mbr.getArea() - hull.getArea())

```

S pomocou dokumentácie knižnice *arcpy*, doplňte chýbajúci kód – nájdite funkcie ktoré v rámci *arcpy* umožňujú výpočet MBR a konvexnej obálky polygónu a syntakticky správne ich použite.

Ešte bonbónik na záver od nemenovaného autora ;)

```

if z>x:
    x=z
elif z<=x:
    x=x

```