

1	2	3	4	5	6	7	8	9	10

Neprajem si zverejniť moje výsledky testu na webe:

Záverečný test z Programovania (2) pre 1. ročník

1. Turingov stroj je zadaný tabuľkou:

	start	jeden	dva
x	> jeden	> dva	> start
y	> start	> jeden	> dva
_		= stop	

Stavy sú: **start**, **jeden**, **dva**, **stop** (koncový) a symboly na páske: **x**, **y**, **_** (prázdny).

Zistite, ktoré z uvedených reťazcov budú týmto turingovým strojom akceptované?

- a) xyxyxyx
- b) yyxyxyx
- c) xyyyyyx
- d) xxxxxxx

2. Zistite, čo vypíše táto funkcia:

```
def test(slovo):
    queue = Queue()
    for znak in slovo:
        try:
            p = queue.dequeue()
            queue.enqueue(znak)
            queue.enqueue(p)
        except EmptyError:
            queue.enqueue(znak)
    while not queue.empty():
        print(queue.dequeue(), end='')
    print()
test('abcdef')
```

Výpis:

3. Zadaný aritmetický výraz:

$$1 + (2 + 3) * 4 / (5 - 6) + 7 * 8$$

prepíšte do

- a) prefixového zápisu
- b) postfixového zápisu

Operácie s rovnakou prioritou sa vyhodnocujú zľava doprava.

4. Napíšte funkciu `vyhod_posledny(zoznam)`, v ktorej parametrom je referencia na začiatok zoznamu typu `Vrchol` (alebo `None`). Funkcia vráti referenciu na pôvodný zoznam, ktorý už bude bez posledného prvku:

```
class Vrchol:
    def __init__(self, data, next=None):
        self.data, self.next = data, next

def vyhod_posledny(zoznam):

    return zoznam
```

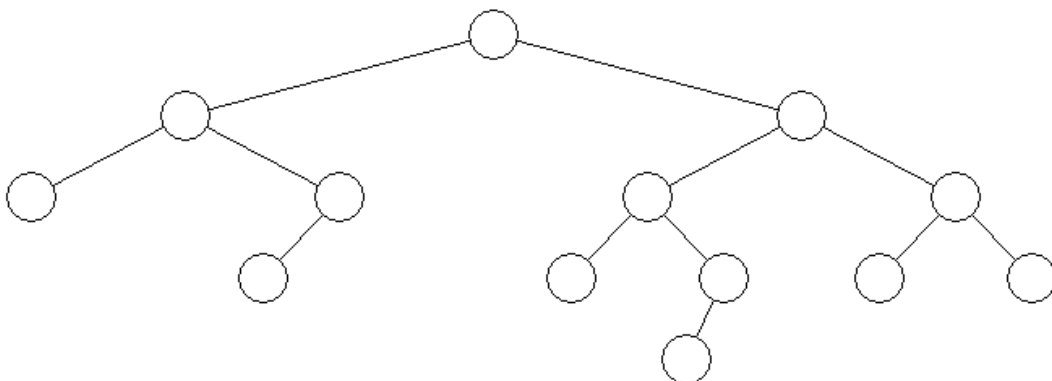
5. Rekurzívna funkcia `vytvor_uplny()` by mala vrátiť úplný binárny strom, v ktorom v listoch budú `0`, v ich predkoch budú `1`, v každej úrovni bližšie ku koreňu bude o 1 väčšie číslo, až v koreni by teda mala byť výška celého stromu - teda `n`:

```
class Vrchol:
    def __init__(self, data, left=None, right=None):
        self.data, self.left, self.right = data, left, right

def vytvor_uplny(n, cislo=0):
    if cislo > n:
        return 0
    lavy = pravy = vytvor_uplny(n-1, cislo+1)
    koren = Vrchol(n-cislo, lavy, pravy)
    return koren
```

Opravte všetky (štyri) chyby v tele funkcie.

6. Pre daný strom vpíšte do vrcholov hodnoty tak, aby výpisom pre postupnosť **postorder** bolo `'programovanie'`.

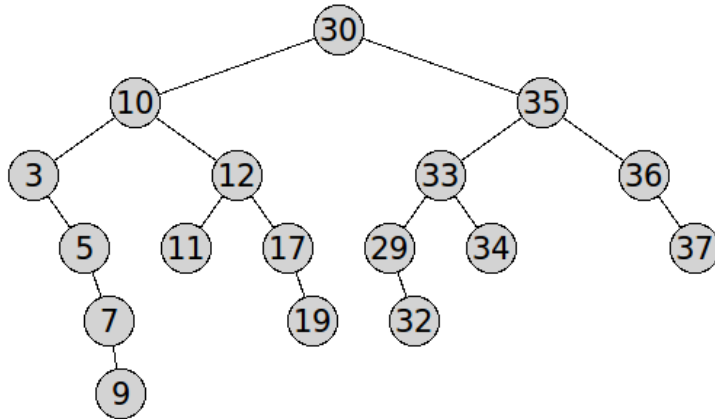


7. Pre daný prefixový zápis

`* + / a b - c d + e f`

nakreslite príslušný **aritmetický strom**.

8. Tento strom možno nespĺňa podmienky BVS (binárny vyhľadávací strom). Označte minimálny počet vrcholov stromu, ktorým keby sme navzájom vymenili ich hodnoty, dostaneme BVS.



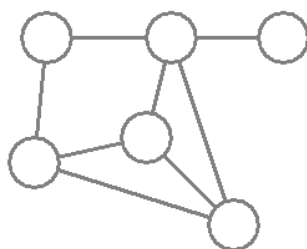
9. Táto verzia triedenia vkladaním v niektorých situáciách vypisuje momentálny obsah celého poľa. Vypíšte tieto kontrolné výpisy:

```
def insert_sort(pole):
    for i in range(len(pole)-1):
        prvok = pole.pop(i+1)
        while i >= 0 and pole[i] > prvok:
            i -= 1
        pole.insert(i+1, prvok)
        print(pole)
```

```
p = [5, 9, 4, 3, 6, 10, 1, 8, 2, 7]
insert_sort(p)
```

riadky výpisu:

10. Označte písmenami vrcholy grafu na obrázku



ak poznáte túto jeho reprezentáciu.

	A	B	C	D	E	F
A	0	1	0	1	0	1
B	1	0	0	1	1	0
C	0	0	0	0	0	1
D	1	1	0	0	0	1
E	0	1	0	0	0	1
F	1	0	1	1	1	0