

1. Uveďte příklad víceznačnosti přirozeného jazyka na rovině fonetické, morfologické, syntaktické a sémantické (pro každou rovinu stačí jeden příklad).

- fonetická: bít/být, soutok oněmí/jsou to koně mí, led/let, plod/plot atp.
- morfologická: ženu, zvěra, želé (ano, i víceznačnost v pádě, tedy téměř cokoli!)
- syntaktická: Karel mluvil o sexu s B. S.
- sémantická: raketa, oko, ...

2. Vyberte tvrzení, které nejlépe vystihuje činnost příkazu `echo`:

Možnosti:

- příkaz vypíše na standardní výstup všechny své argumenty a znak nového řádku,
- příkaz vypíše na standardní výstup všechny své argumenty,
- příkaz vypíše na standardní výstup to, co přečte ze standardního vstupu nebo souborů zadaných argumenty,
- Příkaz vypíše na standardní výstup to, co přečte ze standardního vstupu nebo souborů zadaných argumenty a znak nového řádku.

3. Mějme soubor `zdroj.vert` s vertikálním textem, který na každém řádku obsahuje tabulátorem oddělená pole obsahující slovní tvar, lemma a morfologickou značku (v tomto pořadí). Doplňte vytečkované části tak, aby výstupem celé posloupnosti příkazů byl právě seznam tzv. *hapax legomen*, tedy slov (lemmat) vyskytujících se v souboru právě jednou:

```
cut -f 2 zdroj.vert | sort | uniq -c | grep "^ *1 "
```

4. Vyberte tvrzení, které nejlépe vystihuje činnost následujícího programu:

```

1 import sys
2 i = 0
3 for line in sys.stdin:
4     if (2 * i) % 2 == 0:
5         print line
6         i += 1

```

- program vypíše na standardní výstup všechny řádky, které přečte ze standardního vstupu – protože $2 \cdot n$ (pro $n \in \mathbb{Z}$) je dělitelné dvěma vždy!,
- program nevypíše na standardní žádný z řádků, které přečte ze standardního vstupu,
- program vypíše na standardní výstup právě liché řádky ze všech, které přečte ze standardního vstupu,
- program vypíše na standardní výstup právě sudé řádky ze všech, které přečte ze standardního vstupu.

5. Kolik iterací provede v následujícím programu *vnější* cyklus `while`?

```

1 i = 0
2 j = 5
3 while i <= 2 * j:
4     while j > 0:
5         j -= 1

```

Možnosti:

- 0
- 1
- 2
- 3
- více než 3 – protože zacyklí

6. Doplňte do následujícího programu právě jeden příkaz na řádek 5 tak, aby cyklus `while` provedl právě 4 iterace:

```

1 x = 1
2 y = 1
3 while 2 * x < 5 * y:
4     y += 1
5     x *= 2

```

Uznával jsem i $x += 3$, jakkoli to správně není – v tom případě proběhne pouze tři iterace, iterací se rozumí průběh **těla** cyklu. Toto jsem ale příliš nezdůrazňoval a z řešení většiny z Vás bylo zřejmé, že víte, jak si s úlohou poradit.