

# Řešení domácích úkolů - 2. týden

## Rozklad na prvočinitele

Napište program, který přečte ze vstupu celé číslo větší než 1 a vytiskne jeho rozklad na součin prvočinitelů, oddělených hvězdičkami, v rostoucím pořadí, např. pro vstup

28

vytiskne výstup

28=2\*2\*7

## Řešení

**Analýza** Načítáme jediné číslo, které máme rozložit na prvočinitele a vypsát rozklad v požadovaném tvaru.

Začneme s kódem, který vypíše prvočinitele, a pak budeme zkoumat, jak výstup upravit do požadovaného tvaru.

```
1  n = int(input())
2  p = 2
3  while n > 1:
4      while n % p == 0:
5          n //= p
6          print(p)
7      p += 1
8
9  -----
10 28  # vstup
11 2   # výstup
12 2
13 7
```

## Vzorové řešení

V této úloze máme namísto tisku prvočinitelů na standardní výstup sformovat vstupní číslo a nalezené prvočinitele do požadovaného tvaru. Musíme zejména nějak ošetřit tisk hvězdiček. Například můžeme tisknout hvězdičku za každým prvočinitelem, a pak odstranit poslední, přebývající hvězdičku.

```

1  n = int(input())
2  rozklad = str(n) + "="
3  p = 2
4  while n > 1:
5      while n % p == 0:
6          n //= p
7          rozklad += str(p) + "*"
8      p += 1
9
10 print(rozklad[:-1])

```

## Alternativní řešení

Můžeme také sledovat, zda jsme na konci rozkladu, a poslední hvězdičku nepřidat:

```

1  n = int(input())
2  rozklad = str(n) + "="
3  p = 2
4  sep = "*"
5  while n > 1:
6      while n % p == 0:
7          n //= p
8          if n == 1:
9              sep = ""
10         rozklad += str(p) + sep
11     p += 1
12
13 print(rozklad)

```

Také můžeme seskládat výsledný řetězec ze seznamu prvočinitelů.

```

1  n = int(input())
2  print(str(n) + "=", end="")
3  primes = []
4  p = 2
5  sep = "*"
6  while n > 1:
7      while n % p == 0:
8          n //= p
9          primes.append(p)
10     p += 1
11
12 print(*primes, sep=sep)

```