

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.

```
n = int(input())
p = 2
while n > 1:
    while n % p == 0:
        n //= p
        print(p)
    p += 1
```

```
28  # vstup
2   # výstup
2
7
```

Další vhledy do možných řešení může poskytnout řešení úlohy o převodu do dvojkové soustavy.

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.

```

n = int(input())
rozklad = str(n) + "="
p = 2
while n > 1:
    while n % p == 0:
        n //= p
        rozklad += str(p) + "*"
    p += 1

print(rozklad[:-1])

```

Alternativní řešení

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

```

n = int(input())
rozklad = str(n) + "="
p = 2
sep = "*"
while n > 1:
    while n % p == 0:
        n //= p
        if n == 1:
            sep = ""
        rozklad += str(p) + sep
    p += 1

print(rozklad)

```

Také můžeme seskládat výsledný řetězec ze seznamu prvočinitelů.

```

n = int(input())
print(str(n) + "=", end="")
primes = []
p = 2
sep = "*"
while n > 1:
    while n % p == 0:
        n //= p
        primes.append(p)
    p += 1

print(*primes, sep=sep)

```