

UMETNA INTELIGENCA, SKLAD IN VRSTA

Izbrala sem si nalogo, ki se jo rešuje s poznavanjem skladov. Naloga je sledeča in je povzeta iz (Janez Demšar):

Rešimo podobno nalogo kot so bili Gnezdeni oklepaji, a vzemimo namesto oklepajev črke: velike črke naj predstavljajo "oklepaje", male črke pripadajoče "zaklepaje". Tokrat moramo poleg tega, koliko oklepajev in zaklepajev imamo, paziti tudi na to, za katero črko gre: veliki A "zapremo" le z malim a, ne pa tudi z malim b ali velikim A. Na primer, niz "ABba" je pravilen, a "ABab" pa ne, ker bi se moral B zapreti pred A. Pravilni oklepaji so tudi, "Aa", "AaBb", "AaBbACBbDdcDda", primeri nepravilnih pa "ABC", "aB", "AA", "aA", "ABCabc". (Kdor pozna zapis HTML, se je morda spomnil na oznake v njem. Oznake v pravilnem HTMLju se tudi pravilno zaključujejo.)

Napiši funkcijo, ki prejme takle niz in pove, ali je pravilen ali ne.

Za iskanje rešitve sem uporabila Chat GPT. Moje izpraševanje je izgledalo tako:

Jaz:

Bi mi lahko rešil sledečo nalogo: Rešimo podobno nalogo kot so bili Gnezdeni oklepaji, a vzemimo namesto oklepajev črke: velike črke naj predstavljajo "oklepaje", male črke pripadajoče "zaklepaje". Tokrat moramo poleg tega, koliko oklepajev in zaklepajev imamo, paziti tudi na to, za katero črko gre: veliki A "zapremo" le z malim a, ne pa tudi z malim b ali velikim A. Na primer, niz "ABba" je pravilen, a "ABab" pa ne, ker bi se moral B zapreti pred A. Pravilni oklepaji so tudi, "Aa", "AaBb", "AaBbACBbDdcDda", primeri nepravilnih pa "ABC", "aB", "AA", "aA", "ABCabc". (Kdor pozna zapis HTML, se je morda spomnil na oznake v njem. Oznake v pravilnem HTMLju se tudi pravilno zaključujejo.) Napiši funkcijo, ki prejme takle niz in pove, ali je pravilen ali ne.

Chat GPT:

Seveda, lahko vam pomagam z rešitvijo te naloge. Spodaj je Python koda za funkcijo, ki preverja, ali je podani niz pravilen glede na dana pravila. Uporabimo enostaven pristop s pomočjo sklada ('stack').

python

 Copy code

```
def preveri_pravilnost_niza(niz):  
    stack = []  
    for crka in niz:  
        if crka.isupper():  
            if stack and stack[-1] == crka.lower():  
                stack.pop()  
            else:  
                return False  
        else:  
            stack.append(crka)  
  
    return len(stack) == 0  
  
# Primeri uporabe funkcije:  
niz1 = "AaBbACBbDdcDda"  
niz2 = "ABCabc"  
niz3 = "ABab"  
niz4 = "ABba"  
  
print(preveri_pravilnost_niza(niz1)) # Izhod: True  
print(preveri_pravilnost_niza(niz2)) # Izhod: False  
print(preveri_pravilnost_niza(niz3)) # Izhod: False  
print(preveri_pravilnost_niza(niz4)) # Izhod: True
```

KOMENTAR: Ko sem kodo pognala v VS Code, sem dobila za vse primere izhod False.


Jaz:

Žal v vseh primerih dobim odgovor False.

Chat GPT:

Oprostite za zmedo. Tukaj je posodobljena različica funkcije:

python

 Copy code

```
def preveri_pravilnost_niza(niz):
    stack = []
    for crka in niz:
        if crka.isupper():
            if stack and crka.lower() == stack[-1]:
                stack.pop()
            else:
                return False
        elif crka.islower():
            if stack and crka == stack[-1].lower():
                stack.pop()
            else:
                return False
        else:
            return False

    return len(stack) == 0

# Primeri uporabe funkcije:
niz1 = "AaBbACBbDdcDda"
niz2 = "ABCabc"
niz3 = "ABab"
niz4 = "ABba"

print(preveri_pravilnost_niza(niz1)) # Izhod: True
print(preveri_pravilnost_niza(niz2)) # Izhod: False
print(preveri_pravilnost_niza(niz3)) # Izhod: False
print(preveri_pravilnost_niza(niz4)) # Izhod: True
```



KOMENTAR: Opazila sem, da nikoli nič ne daje v sklad stack.


Jaz:

Mislím, da je problem, da ničesar ne daješ v sklad in je zato vedno prazen.

Chat GPT:

Oprostite za napako. Res je, pozabil sem dodati črke v sklad v primeru, ko preverjam niz. Tukaj je posodobljena verzija funkcije, ki pravilno upravlja s skladiščenjem črk:

python

 Copy code

```
def preveri_pravilnost_niza(niz):
    stack = []
    for crka in niz:
        if crka.isupper():
            if stack and crka.lower() == stack[-1]:
                stack.pop()
            else:
                return False
        elif crka.islower():
            stack.append(crka)
        else:
            return False

    return len(stack) == 0

# Primeri uporabe funkcije:
niz1 = "AaBbACBbDdcDda"
niz2 = "ABCabc"
niz3 = "ABab"
niz4 = "ABba"

print(preveri_pravilnost_niza(niz1)) # Izhod: True
print(preveri_pravilnost_niza(niz2)) # Izhod: False
print(preveri_pravilnost_niza(niz3)) # Izhod: False
print(preveri_pravilnost_niza(niz4)) # Izhod: True
```

KOMENTAR: Še vedno rešitev ni pravilna, zato bom Chat GPT ponovno izprašala, pri tem pa mu bom podala namig.


Jaz:

Vendar tudi v primeru, ko je črka velika bi moral dati v sklad, da potem skupaj z malo črko primerjaš ali je enaka tisti veliki, ki je trenutno na vrhu sklada.

Chat GPT:

Imate prav, moje opravičilo za napako. Potrebno je dodati velike črke v sklad, da lahko preverimo, ali se pravilno zaprejo z ustreznimi malimi črkami. Tukaj je pravilna verzija funkcije:

python

 Copy code

```
def preveri_pravilnost_niza(niz):
    stack = []
    for crka in niz:
        if crka.isupper():
            stack.append(crka)
        elif crka.islower():
            if stack and crka == stack[-1].lower():
                stack.pop()
            else:
                return False
        else:
            return False

    return len(stack) == 0

# Primeri uporabe funkcije:
niz1 = "AaBbACBbDdcDda"
niz2 = "ABCabc"
niz3 = "ABab"
niz4 = "ABba"

print(preveri_pravilnost_niza(niz1)) # Izhod: True
print(preveri_pravilnost_niza(niz2)) # Izhod: False
print(preveri_pravilnost_niza(niz3)) # Izhod: False
print(preveri_pravilnost_niza(niz4)) # Izhod: True
```

KOMENTAR: Šele sedaj sem dobila pravilno rešitev.

Morala sem Chat GPT kar nekajkrat popraviti. Če sama ne bi videla rešitve iz knjige (Janez Demšar), potem mislim, da sama ne bi znala podati namigov. Je pa res, da nikoli nisem imela težave, da Chat GPT ne bi že od začetka uporabljal skalda. Še vseeno pa se mi zdi rešitev iz knjige lepša kot ta, ki mi jo je podal Chat GPT.

Bibliography

Janez Demšar, J. Ž. (n.d.). *Rešene naloge iz programiranja v Pythonu*. Fakulteta za računalništvo in informatiko.