

Programarea Algoritmilor

– SEMINAR NR. 2 –

(grupa 131)

1. Modificarea/ștergerea caracterului aflat pe o poziție p într-un șir de caractere s .

Modificarea: $s = s[:p] + \text{caracter_nou} + s[p+1:]$

Ștergerea: $s = s[:p] + s[p+1:]$

2. Modificarea/ștergerea tuturor aparițiilor unui caracter (sau subșir) dintr-un șir de caractere s .

Modificarea: $s = s.\text{replace}(\text{șir_inițial}, \text{șir_nou})$

Ștergerea: $s = s.\text{replace}(\text{șir_inițial}, "")$

3. Se citește un cuvânt w , un număr natural nenul p și un șir format din n cuvinte. Să se afișeze toate cuvintele care sunt p -rime cu w , respectiv ultimele p caractere coincid.

```
w = input("Cuvântul w: ")
p = int(input("Numărul p: "))
n = int(input("Numărul de cuvinte: "))
r = ""
for i in range(n):
    cuv = input("Cuvânt: ")
    if len(cuv) >= p and w[-p:] == cuv[-p:]:
        r = r + cuv + " "
r = r.strip()
print(str(p) + "-rimele cuvântului " + w + ": " + r)
```

4. Se citește o propoziție/frază în care cuvintele sunt despărțite între ele prin spații și semnele de punctuație uzuale. Să se afișeze toate cuvintele distincte de lungime maximă.

```
text = input("Propoziția/fraza: ")

separatori = ",.;;?!"
for x in separatori:
    text = text.replace(x, " ")

rez = " "
lmax = 0
for cuv in text.split():
    if len(cuv) > lmax:
        rez = " " + cuv + " "
        lmax = len(cuv)
    elif len(cuv) == lmax:
        if " " + cuv + " " not in rez:
            rez += cuv + " "
rez = rez.strip()

print("Cuvintele distincte de lungime maximă: " + rez)
```

5. Se citește un șir de caractere s . Să se verifice dacă există un șir t , diferit de s , astfel încât s să se poată obține prin concatenarea de un număr arbitrar de ori k a șirului t .

Exemplu: pentru $s = "abcabc" \Rightarrow t = "abc", k = 2$.

```

s = input("Șirul s: ")
n = len(s)
for d in range(1, n//2 + 1):
    if n % d == 0:
        t = s[:d] * (n//d)
        if t == s:
            print("t = ", s[:d], "\nk = ", n//d)
            break
else:
    print("Imposibil!")

```

Observație: Programul de mai sus va determina cel mai scurt subșir t având proprietatea cerută! Astfel, pentru $s = "abababab" \Rightarrow t = "ab", k = 4$. Dacă dorim să determinăm cel mai lung subșir t cu proprietatea cerută vom înlocui $\text{range}(1, n//2 + 1)$ cu $\text{range}(n//2, 0, -1)$. În acest caz, pentru $s = "abababab" \Rightarrow t = "abab", k = 2$.