

1. Wprowadzenie

Silniowy system pozycyjny to pozycyjny sposób zapisu liczb naturalnych, dla którego mnożniki następnych pozycji czytanych od prawej strony są silnią kolejnych liczb naturalnych, począwszy od 1. Tak oto cyfra na pozycji pierwszej od prawej posiada mnożnik 1, druga – 2, trzecia – 6, czwarta – 24, piąta - 120, itd. Liczby w silniowym systemie pozycyjnym zapisuje się w następujący sposób: “(54321)!” , gdzie za 54321 podstawiamy liczbę wyrażoną w systemie silniowym.

Przykład:

$$(1220)! = 1 \cdot 4! + 2 \cdot 3! + 2 \cdot 2! + 0 \cdot 1! = 24 + 12 + 4 + 0 = 40$$

2. Opis działania algorytmu zamiany systemu silniowego na dziesiętny

Algorytm polega na przeiterowaniu od prawej do lewej strony kolejnych cyfr składających się na liczbę w systemie silniowym i pomnożeniu każdorazowo tej cyfry przez silnię pozycji danej cyfry od prawej strony, licząc od 1 (tj. Pierwsza cyfra od prawej * 1, druga * 2, trzecia * 6, czwarta * 24, itd.). Następnie algorytm zwraca sumę tych iloczynów, jako reprezentację tej cyfry w systemie dziesiętnym.

3. Przykładowy kod w języku Python3

```
# Zamiana z silniowego systemu pozycyjnego na dziesiętny
def siln_na_dzies(liczba):
    wynik = 0 # Zmienna przechowująca wynik
    liczba = str(liczba) # Potrzebujemy str jako typ zmiennej, gdyż będziemy po niej iterować
    silnia = 1 # Zmienna przechwywająca wartości kolejnych pozycji systemu silniowego

    for i in range(1, len(liczba)+1):
        silnia *= i # Obliczanie wartości silni z i
        wynik += (int(liczba[-i]) * silnia) # Dodawanie do wyniku iloczynu wartości pozycji w systemie pozycyjnym i liczby na danej pozycji

    return wynik # Zwracanie wyniku działania funkcji
```

4. Opis działania algorytmu zamiany systemu dziesiętnego na silniowy

Algorytm opiera swoje działanie na określeniu największej silni z liczby n , która jest mniejsza lub równa liczbie w systemie dziesiętnym, którą pragniemy zamienić na system silniowy. Po ustaleniu takiej liczby do zmiennej tekstowej zawierającej wynik działania algorytmu konkatenujemy tekstową reprezentację wyniku dzielenia całkowitego liczby do zmiany przez wartość $n!$. Następnie liczbę tę zastępujemy resztą z dzielenia owej liczby przez $n!$, a następnie zmienną przechowującą wartość największej silni, dzielimy przez n , aby od teraz zmienna ta przechowywała wartość silni z $(n-1)$. Następnie zmienną n zmniejszamy o jeden, aby przejść do określenia cyfry znajdującej się na kolejnej pozycji zapisu. Następnie algorytm powraca do konkatencji tekstowej reprezentacji wyniku dzielenia całkowitego liczby do zmiany, przez wartość $n!$. Całość jest powtarzana, dopóki $n > 1$.

5. Przykładowy kod w języku Python3

```
# Zamiana z dziesiętnego na silniowy
def dzies_na_siln(liczba):
    silnia = 1 # Zmienna przechowująca wartość silni
    n = 1 # Zmienna przechowująca liczbę z której policzona jest silnia

    # Pętla służąca do znalezienia największej silni mieszczącej się w liczbie
    while silnia <= liczba:
        n += 1
        silnia *= n
    silnia //= n
    n -= 1

    wynik = "" # Zmienna przechowująca wynik
    while n > 0:
        wynik += str(int(liczba // silnia) ) # Dodawanie do wyniku całkowitej ilości wartości
        silni możliwych do zmieszczenia w liczbie
        liczba %= silnia # Przypisywanie do zmiennej liczba reszty z dzielenia przez silnię
        silnia //= n # Dekrementacja wartości silni z n! do (n-1)!
        n -= 1 # Zmniejszanie n, aby odpowiadała wartości silni
```

return wynik # Zwracanie wyniku