

REMIGIUSZ NOWAKOWSKI 82225

Zad. 3

Kod python:

class MaszynaTuringa:

def __init__(self):

self.tasma = []

self.glowica = 0

self.stan = 'q0'

def wczytaj_tasme(self, tekst):

self.tasma = list(tekst)

def krok(self):

symbol = self.tasma[self.glowica] if self.glowica < len(self.tasma) else ''

if self.stan == 'q0':

if symbol == 'A':

self.tasma[self.glowica] = 'C'

self.glowica += 1

elif symbol in 'BCD':

self.glowica += 1

elif symbol == '':

self.stan = 'qf'

def uruchom(self):

while self.stan != 'qf':

self.krok()

def pobierz_wynik(self):

return ''.join(self.tasma)

maszyna = MaszynaTuringa()

REMIGIUSZ NOWAKOWSKI 82225

Użytkownik podaje tekst

```
tekst = input("Podaj tekst z literami A, B, C i D: ")
```

```
maszyna.wczytaj_tasme(tekst)
```

```
maszyna.uruchom()
```

```
nowy_tekst = maszyna.pobierz_wynik()
```

```
print(f"Tekst przed zamianą: {tekst}")
```

```
print(f"Tekst po zamianie liter A na C: {nowy_tekst}")
```

Opis:

Definicja klasy **MaszynaTuringa**:

- Program rozpoczyna od zdefiniowania klasy **MaszynaTuringa**, która reprezentuje maszynę Turinga do zamiany liter "A" na "C" w tekście.
2. Inicjalizacja klasy **MaszynaTuringa**:
- Tworzenie konstruktora klasy **MaszynaTuringa**, który inicjalizuje zmienne i ustawia początkowy stan na 'q0'.
3. Metoda **wczytaj_tasme(self, tekst)**:
- Metoda, która przyjmuje tekst jako argument i konwertuje go na listę symboli, reprezentujących taśmę maszyny Turinga.
4. Metoda **krok(self)**:
- Metoda, która reprezentuje jeden krok działania maszyny Turinga.
 - Odczytuje symbol spod głowicy.
 - Jeśli stan jest 'q0' i symbol to 'A', zamienia go na 'C' i przesuwą głowicę w prawo.
 - Jeśli symbol to 'B', 'C', 'D' lub spacja, to przesuwą głowicę w prawo.
 - Jeśli symbol to spacja (koniec tekstu), to zmienia stan na 'qf', oznaczając zakończenie.
5. Metoda **uruchom(self)**:
- Metoda, która uruchamia maszynę Turinga, wywołując kroki, aż osiągnie stan 'qf' (zakończenie).

REMIGIUSZ NOWAKOWSKI 82225

6. Metoda **pobierz_wynik(self)**:

- Metoda, która konwertuje taśmę z powrotem na tekst i zwraca go jako wynik.

7. Tworzenie instancji klasy **MaszynaTuringa**:

- Program tworzy instancję klasy **MaszynaTuringa**.

8. Użytkownik podaje tekst:

- Program prosi użytkownika o podanie tekstu, który zawiera litery A, B, C, D.

9. Wywołanie metody **wczytaj_tasme**:

- Tekst wprowadzony przez użytkownika jest przekazywany do metody **wczytaj_tasme** maszyny Turinga, która konwertuje go na taśmę.

10. Uruchomienie maszyny Turinga:

- Program uruchamia maszynę Turinga za pomocą metody **uruchom**.

11. Wywołanie metody **pobierz_wynik** i wyświetlenie wyniku:

- Program pobiera wynik w postaci przetworzonego tekstu, w którym litery "A" zostały zamienione na "C" i wyświetla wynik na konsoli.

12. Zakończenie programu:

- Program kończy się, pozostawiając użytkownika z wynikami zamiany liter w tekście.

Zad.4

Kod python:

```
def przesun_pierwsza_literke(tekst):
```

```
    if not tekst:
```

```
        return tekst # Zwróć pusty tekst, jeśli wejściowy tekst jest pusty.
```

```
    tekst_lista = list(tekst) # Konwertuj tekst na listę znaków.
```

```
    pierwsza_literka = tekst_lista[0] # Zapisz pierwszą literkę.
```

```
    for i in range(1, len(tekst_lista)):
```

```
        tekst_lista[i - 1] = tekst_lista[i] # Przesuń każdą literkę o jedno miejsce w lewo.
```

REMIGIUSZ NOWAKOWSKI 82225

```
tekst_lista[-1] = pierwsza_literka # Wstaw zapisaną pierwszą literkę na koniec.
```

```
nowy_tekst = ''.join(tekst_lista) # Połącz listę w nowy tekst.
```

```
return nowy_tekst
```

Przykład użycia:

```
tekst = input("Podaj wyraz zbudowany z liter A, B, C i D: ")
```

```
nowy_tekst = przesun_pierwsza_literke(tekst)
```

```
print(f"Wyraz przed przesunięciem: {tekst}")
```

```
print(f"Wyraz po przesunięciu: {nowy_tekst}")
```

Opis:

1. Program rozpoczyna się od definicji funkcji "przesun_pierwsza_literke" z jednym parametrem, którym jest tekst.
2. Sprawdza, czy tekst jest pusty. Jeśli tak, zwraca go bez zmian, ponieważ nie ma potrzeby przesuwania literki w pustym tekście.
3. Następnie konwertuje tekst na listę znaków, co pozwoli na manipulację jego zawartością.
4. Zapisuje pierwszą literkę (element o indeksie 0) w zmiennej "pierwsza_literka".
5. Rozpoczyna pętlę "for", która przechodzi przez wszystkie pozostałe litery w tekście, zaczynając od drugiej litery (indeks 1).
6. W każdym kroku pętli przesuwa literkę o jedno miejsce w lewo, nadpisując literkę poprzedzającą ją w liście.
7. Po zakończeniu pętli, na ostatniej pozycji w liście zostaje pusta przestrzeń, gdzie wcześniej była pierwsza literka.
8. Wstawia zapisaną wcześniej pierwszą literkę na koniec listy, zastępując pustą przestrzeń.
9. Łączy listę znaków w jedną linię, tworząc nowy tekst.
10. Zwraca nowy tekst jako wynik działania funkcji.
11. Program wczytuje wyraz od użytkownika.
12. Wywołuje funkcję "przesun_pierwsza_literke" z podanym wyrazem.
13. Wyświetla pierwotny wyraz przed przesunięciem oraz wyraz po przesunięciu na ekranie.

REMIGIUSZ NOWAKOWSKI 82225

Zad. 2

Kod python:

class MaszynaTuringa:

def __init__(self, tabela_przejsc):

self.tabela_przejsc = tabela_przejsc

self.stan = 'q0'

self.pozycja_glowicy = 0

self.taśma = []

def wczytaj_tasme(self, wejscowe_slowo):

self.taśma = list(wejscowe_slowo + 'B') # Dodajemy pole B na końcu taśmy

def uruchom(self):

while self.stan != 'qf':

symbol = self.taśma[self.pozycja_glowicy]

if (self.stan, symbol) in self.tabela_przejsc:

nowy_stan, nowy_symbol, kierunek = self.tabela_przejsc[(self.stan, symbol)]

self.taśma[self.pozycja_glowicy] = nowy_symbol

if kierunek == 'R':

self.pozycja_glowicy += 1

elif kierunek == 'L':

self.pozycja_glowicy -= 1

self.stan = nowy_stan

else:

break

REMIGIUSZ NOWAKOWSKI 82225

Obliczenie ilości "1" w słowie

```
ilosc_jedynek = self.taśma.count('1')
```

```
return ilosc_jedynek
```

Przykład użycia:

```
tabela_przejsc = {  
    ('q0', '1'): ('q0', '1', 'R'),  
    ('q0', '0'): ('q0', '0', 'R'),  
    ('q0', 'B'): ('qf', 'B', 'S'), # Zakończenie obliczeń  
}
```

```
maszyna = MaszynaTuringa(tabela_przejsc)
```

```
wejscowe_slowo = input("Podaj słowo binarne: ")
```

```
maszyna.wczytaj_tasme(wejscowe_slowo)
```

```
ilosc_jedynek = maszyna.uruchom()
```

```
print(f"Ilość wystąpień '1' w słowie: {ilosc_jedynek}")
```

Opis:

1. Maszyna Turinga rozpoczyna działanie w stanie początkowym na pierwszej komórce taśmy (pierwszym znaku w słowie).
2. Głowica odczytuje znak na taśmie i sprawdza, czy jest to "1" czy "0".
3. Jeśli odczytany znak to "1", maszyna zapisuje ten fakt, na przykład inkrementując licznik (ewentualnie zapisując wynik na innej części taśmy).
4. Głowica przemieszcza się w prawo, aby przejść do kolejnej komórki taśmy.
5. Proces powtarza się, aż głowica dotrze do końca słowa (gdzie znajduje się znak "B" oznaczający puste pole).
6. Wówczas maszyna Turinga zakończy działanie i zwróci wynik obliczeń, którym jest ilość wystąpień "1" w słowie.

Ta maszyna Turinga realizuje zadanie obliczania ilości "1" w danym słowie binarnym, iterując przez wszystkie znaki w słowie i zliczając "1". Jeśli nie ma już znaków "1" do zliczenia, zakończy swoje działanie.