

# Program palindrom

w matlabie

Piotr Szyszka  
Ewa Podlowska

Politechnika Lubelska

21 maja 2021

# *Spis treści*

- ❶ Wprowadzenie
- ❷ Algorytm
- ❸ Skrypt oraz przykłady wywołania
- ❹ Zakończenie

# Wprowadzenie

**Palindrom**, z greckiego *palindromos*, czyli biegnący na powrót lub **palindromeo** – biec z powrotem – to wyrażenie brzmiące tak samo czytane od lewej do prawej i od prawej do lewej.

Współcześnie palindromy pełnią funkcję gry słownej. Prawdopodobnie tak było również i w przeszłości, choć pewne znaleziska sugerują, że palindromy mogły też mieć znaczenie magiczne.

Przykładowe wyrazy będące palindromami: **Kajak, Aga, Ala, Kamil Ślimak**. Palindromy występują również w liczbach np. **2267622, 343, 212**.

# Działanie programu sprawdzającego, czy wprowadzona liczba naturalna jest palindromem

- 1 Użytkownik wprowadza liczbę
- 2 Zostaje zdefiniowana funkcja palindrom:

- Wykonuje się instrukcja if else

```
if (x(1)~= "0" & str2num(x) > 0 & mod(str2num(x),1) == 0)
```

o warunkach sprawdzających czy:

- pierwsza cyfra nie jest zerem
- wprowadzona liczba jest liczbą naturalną
- jest liczbą parzystą

jeśli powyższe warunki nie są spełnione, następuje wyświetlenie komunikatu o niepoprawności zapisu i prośba o ponowne wprowadzenie liczby

- 3 Algorytm badania palindromu

- Zostaje wywołana pętla for, o liczniku *i*, wykonująca się tyle razy, z ilu cyfr składa się podana liczba.

- Wewnątrz pętli wywołana jest funkcja `palindrom`, która przyjmuje jako argumenty liczbę i licznik. Funkcja sprawdza, czy liczba jest palindromem. Jeśli tak, zwraca wartość 1, jeśli nie, zwraca wartość 0.
- W zależności od wartości zmiennej *wynik*, następuje wyświetlenie komunikatu tekstowego, który już bezpośrednio informuje użytkownika o rezultacie.

```
if(wynik == 1)  
    disp('Liczba jest palindromem')  
else disp('Liczba nie jest palindromem')  
end
```

- 4 Użytkownik ma możliwość ponownego wywołania programu, lub zakończenia jego pracy

# Działanie programu sprawdzającego, czy wprowadzona liczba naturalna jest palindromem

- ❶ Użytkownik wprowadza liczbę
- ❷ Zostaje zdefiniowana funkcja palindrom:
  - Wykonuje się instrukcja if else

```
if (x(1) ~= "0" & str2num(x) > 0 & mod(str2num(x),1) == 0)
```

o warunkach sprawdzających czy:

- pierwsza cyfra jest różna od 0,
- wprowadzona wartość jest liczbą dodatnią,
- jest liczbą całkowitą.

jeśli powyższe warunki nie są spełnione, następuje wyświetlenie komunikatu o niepoprawności zapisu i prośba o ponowne wprowadzenie liczby

## ❸ Algorytm badania palindromu

- Zostaje wywołana pętla *for*, o liczniku *i*, wykonująca się tyle razy, z ilu cyfr składa się podana liczba.
  - Warunkiem zerwania pętli jest nieidentyczność *i-tej* cyfry liczby z symetryczną do niej cyfrą.  
Aby móc odwoływać się do konkretnych cyfr liczby, należy zapisać ją jako *string*.  
Jeżeli odpowiednie cyfry zgadzają się ze sobą (badana liczba jest palindromem), pod zmienną *wynik* zapisywana jest wartość 1. Z kolei, jeżeli pętla się zerwie, zmienna *wynik* przyjmuje wartość równą 0.
- W zależności od wartości zmiennej *wynik*, następuje wyświetlenie komunikatu tekstowego, który już bezpośrednio informuje użytkownika o rezultacie.

```
if(wynik == 1)
    disp('Liczba jest palindromem')
else disp('Liczba nie jest palindromem')
end
```

- ❹ Użytkownik ma możliwość ponownego wywołania programu, lub zakończenia jego pracy

# Działanie programu sprawdzającego, czy wprowadzona liczba naturalna jest palindromem

- 1 Użytkownik wprowadza liczbę
- 2 Zostaje zdefiniowana funkcja palindrom:
  - Wykonuje się instrukcja if else

```
if (x(1) ~= "0" & str2num(x) > 0 & mod(str2num(x),1) == 0)
```

o warunkach sprawdzających czy:

- pierwsza cyfra jest różna od 0,
- wprowadzona wartość jest liczbą dodatnią,
- jest liczbą całkowitą.

jeśli powyższe warunki nie są spełnione, następuje wyświetlenie komunikatu o niepoprawności zapisu i prośba o ponowne wprowadzenie liczby

## 3 Algorytm badania palindromu

- Zostaje wywołana pętla *for*, o liczniku *i*, wykonująca się tyle razy, z ilu cyfr składa się podana liczba.
  - Warunkiem zerwania pętli jest nieidentyczność *i-tej* cyfry liczby z symetryczną do niej cyfrą.  
Aby móc odwoływać się do konkretnych cyfr liczby, należy zapisać ją jako *string*.  
Jeżeli odpowiednie cyfry zgadzają się ze sobą (badana liczba jest palindromem), pod zmienną *wynik* zapisywana jest wartość 1. Z kolei, jeżeli pętla się zerwie, zmienna *wynik* przyjmuje wartość równą 0.
- W zależności od wartości zmiennej *wynik*, następuje wyświetlenie komunikatu tekstowego, który już bezpośrednio informuje użytkownika o rezultacie.

```
if(wynik == 1)  
    disp('Liczba jest palindromem')  
else disp('Liczba nie jest palindromem')  
end
```

- 4 Użytkownik ma możliwość ponownego wywołania programu, lub zakończenia jego pracy

# Działanie programu sprawdzającego, czy wprowadzona liczba naturalna jest palindromem

- ① Użytkownik wprowadza liczbę
- ② Zostaje zdefiniowana funkcja palindrom:
  - Wykonuje się instrukcja if else

```
if (x(1) ~= "0" & str2num(x) > 0 & mod(str2num(x),1) == 0)
```

o warunkach sprawdzających czy:

- pierwsza cyfra jest różna od 0,
- wprowadzona wartość jest liczbą dodatnią,
- jest liczbą całkowitą.

jeśli powyższe warunki nie są spełnione, następuje wyświetlenie komunikatu o niepoprawności zapisu i prośba o ponowne wprowadzenie liczby

## ③ Algorytm badania palindromu

- Zostaje wywołana pętla *for*, o liczniku *i*, wykonująca się tyle razy, z ilu cyfr składa się podana liczba.
  - Warunkiem zerwania pętli jest nieidentyczność *i-tej* cyfry liczby z symetryczną do niej cyfrą.  
Aby móc odwoływać się do konkretnych cyfr liczby, należy zapisać ją jako *string*.  
Jeżeli odpowiednie cyfry zgadzają się ze sobą (badana liczba jest palindromem), pod zmienną *wynik* zapisywana jest wartość 1. Z kolei, jeżeli pętla się zerwie, zmienna *wynik* przyjmuje wartość równą 0.
- W zależności od wartości zmiennej *wynik*, następuje wyświetlenie komunikatu tekstowego, który już bezpośrednio informuje użytkownika o rezultacie.

```
if(wynik == 1)
    disp('Liczba jest palindromem')
else disp('Liczba nie jest palindromem')
end
```

- ④ Użytkownik ma możliwość ponownego wywołania programu, lub zakończenia jego pracy

# Działanie programu sprawdzającego, czy wprowadzona liczba naturalna jest palindromem

- 1 Użytkownik wprowadza liczbę
- 2 Zostaje zdefiniowana funkcja palindrom:
  - Wykonuje się instrukcja if else

```
if (x(1) ~= "0" & str2num(x) > 0 & mod(str2num(x),1) == 0)
```

o warunkach sprawdzających czy:

- pierwsza cyfra jest różna od 0,
- wprowadzona wartość jest liczbą dodatnią,
- jest liczbą całkowitą.

jeśli powyższe warunki nie są spełnione, następuje wyświetlenie komunikatu o niepoprawności zapisu i prośba o ponowne wprowadzenie liczby

## 3 Algorytm badania palindromu

- Zostaje wywołana pętla *for*, o liczniku *i*, wykonująca się tyle razy, z ilu cyfr składa się podana liczba.
  - Warunkiem zerwania pętli jest nieidentyczność *i-tej* cyfry liczby z symetryczną do niej cyfrą.  
Aby móc odwoływać się do konkretnych cyfr liczby, należy zapisać ją jako *string*.  
Jeżeli odpowiednie cyfry zgadzają się ze sobą (badana liczba jest palindromem), pod zmienną *wynik* zapisywana jest wartość 1. Z kolei, jeżeli pętla się zerwie, zmienna *wynik* przyjmuje wartość równą 0.
- W zależności od wartości zmiennej *wynik*, następuje wyświetlenie komunikatu tekstowego, który już bezpośrednio informuje użytkownika o rezultacie.

```
if(wynik == 1)  
    disp('Liczba jest palindromem')  
else disp('Liczba nie jest palindromem')  
end
```

- 4 Użytkownik ma możliwość ponownego wywołania programu, lub zakończenia jego pracy



# Działanie programu sprawdzającego, czy wprowadzona liczba naturalna jest palindromem

- 1 Użytkownik wprowadza liczbę
- 2 Zostaje zdefiniowana funkcja palindrom:
  - Wykonuje się instrukcja if else

```
if (x(1) ~= "0" & str2num(x) > 0 & mod(str2num(x),1) == 0)
```

o warunkach sprawdzających czy:

- pierwsza cyfra jest różna od 0,
- wprowadzona wartość jest liczbą dodatnią,
- jest liczbą całkowitą.

jeśli powyższe warunki nie są spełnione, następuje wyświetlenie komunikatu o niepoprawności zapisu i prośba o ponowne wprowadzenie liczby

## 3 Algorytm badania palindromu

- Zostaje wywołana pętla *for*, o liczniku *i*, wykonująca się tyle razy, z ilu cyfr składa się podana liczba.
  - Warunkiem zerwania pętli jest nieidentyczność *i-tej* cyfry liczby z symetryczną do niej cyfrą.  
Aby móc odwoływać się do konkretnych cyfr liczby, należy zapisać ją jako *string*.  
Jeżeli odpowiednie cyfry zgadzają się ze sobą (badana liczba jest palindromem), pod zmienną *wynik* zapisywana jest wartość 1. Z kolei, jeżeli pętla się zerwie, zmienna *wynik* przyjmuje wartość równą 0.
- W zależności od wartości zmiennej *wynik*, następuje wyświetlenie komunikatu tekstowego, który już bezpośrednio informuje użytkownika o rezultacie.

```
if(wynik == 1)  
    disp('Liczba jest palindromem')  
else disp('Liczba nie jest palindromem')  
end
```

- 4 Użytkownik ma możliwość ponownego wywołania programu, lub zakończenia jej pracy

# Działanie programu sprawdzającego, czy wprowadzona liczba naturalna jest palindromem

- ❶ Użytkownik wprowadza liczbę
- ❷ Zostaje zdefiniowana funkcja palindrom:
  - Wykonuje się instrukcja if else

```
if (x(1) ~= "0" & str2num(x) > 0 & mod(str2num(x),1) == 0)
```

o warunkach sprawdzających czy:

- pierwsza cyfra jest różna od 0,
- wprowadzona wartość jest liczbą dodatnią,
- jest liczbą całkowitą.

jeśli powyższe warunki nie są spełnione, następuje wyświetlenie komunikatu o niepoprawności zapisu i prośba o ponowne wprowadzenie liczby

## ❸ Algorytm badania palindromu

- Zostaje wywołana pętla *for*, o liczniku *i*, wykonująca się tyle razy, z ilu cyfr składa się podana liczba.
  - Warunkiem zerwania pętli jest nieidentyczność *i-tej* cyfry liczby z symetryczną do niej cyfrą.  
Aby móc odwoływać się do konkretnych cyfr liczby, należy zapisać ją jako **string**.  
Jeżeli odpowiednie cyfry zgadzają się ze sobą (badana liczba jest palindromem), pod zmienną *wynik* zapisywana jest wartość 1. Z kolei, jeżeli pętla się zerwie, zmienna *wynik* przyjmuje wartość równą 0.
- W zależności od wartości zmiennej *wynik*, następuje wyświetlenie komunikatu tekstowego, który już bezpośrednio informuje użytkownika o rezultacie.

```
if(wynik == 1)
    disp('Liczba jest palindromem')
else disp('Liczba nie jest palindromem')
end
```

- ❹ Użytkownik ma możliwość ponownego wywołania programu, lub zakończenia jego pracy

# Działanie programu sprawdzającego, czy wprowadzona liczba naturalna jest palindromem

- ① Użytkownik wprowadza liczbę
- ② Zostaje zdefiniowana funkcja palindrom:
  - Wykonuje się instrukcja if else

```
if (x(1) ~= "0" & str2num(x) > 0 & mod(str2num(x),1) == 0)
```

o warunkach sprawdzających czy:

- pierwsza cyfra jest różna od 0,
- wprowadzona wartość jest liczbą dodatnią,
- jest liczbą całkowitą.

jeśli powyższe warunki nie są spełnione, następuje wyświetlenie komunikatu o niepoprawności zapisu i prośba o ponowne wprowadzenie liczby

## ③ Algorytm badania palindromu

- Zostaje wywołana pętla *for*, o liczniku *i*, wykonująca się tyle razy, z ilu cyfr składa się podana liczba.
  - Warunkiem zerwania pętli jest nieidentyczność *i-tej* cyfry liczby z symetryczną do niej cyfrą.  
Aby móc odwoływać się do konkretnych cyfr liczby, należy zapisać ją jako **string**.  
Jeżeli odpowiednie cyfry zgadzają się ze sobą (badana liczba jest palindromem), pod zmienną *wynik* zapisywana jest wartość 1. Z kolei, jeżeli pętla się zerwie, zmienna *wynik* przyjmuje wartość równą 0.
- W zależności od wartości zmiennej *wynik*, następuje wyświetlenie komunikatu tekstowego, który już bezpośrednio informuje użytkownika o rezultacie.

```
if(wynik == 1)
    disp('Liczba jest palindromem')
else disp('Liczba nie jest palindromem')
end
```

- ④ Użytkownik ma możliwość ponownego wywołania programu, lub zakończenia jego pracy

# Działanie programu sprawdzającego, czy wprowadzona liczba naturalna jest palindromem

- 1 Użytkownik wprowadza liczbę
- 2 Zostaje zdefiniowana funkcja palindrom:
  - Wykonuje się instrukcja if else

```
if (x(1) ~= "0" & str2num(x) > 0 & mod(str2num(x),1) == 0)
```

o warunkach sprawdzających czy:

- pierwsza cyfra jest różna od 0,
- wprowadzona wartość jest liczbą dodatnią,
- jest liczbą całkowitą.

jeśli powyższe warunki nie są spełnione, następuje wyświetlenie komunikatu o niepoprawności zapisu i prośba o ponowne wprowadzenie liczby

## 3 Algorytm badania palindromu

- Zostaje wywołana pętla *for*, o liczniku *i*, wykonująca się tyle razy, z ilu cyfr składa się podana liczba.
  - Warunkiem zerwania pętli jest nieidentyczność *i-tej* cyfry liczby z symetryczną do niej cyfrą.  
Aby móc odwoływać się do konkretnych cyfr liczby, należy zapisać ją jako **string**.  
Jeżeli odpowiednie cyfry zgadzają się ze sobą (badana liczba jest palindromem), pod zmienną *wynik* zapisywana jest wartość 1. Z kolei, jeżeli pętla się zerwie, zmienna *wynik* przyjmuje wartość równą 0.
- W zależności od wartości zmiennej *wynik*, następuje wyświetlenie komunikatu tekstowego, który już bezpośrednio informuje użytkownika o rezultacie.

```
if(wynik == 1)
    disp('Liczba jest palindromem')
else disp('Liczba nie jest palindromem')
end
```

- 4 Użytkownik ma możliwość ponownego wywołania programu, lub zakończenia jej pracy

# Skrypt

```
x = input('wprowadz liczbę: ', 's')
    p(x)
function palindrom = p(x)
    if (x(1)~= "0" & str2num(x) > 0 & mod(str2num(x),1) == 0)
        for i = 1:length(x)
            if(x(i)~=x(end+1-i))
                wynik = 0;

                break;
            else
                wynik = 1;
            end
        end
        if(wynik == 1)
            disp('Liczba jest palindromem')
        else disp('Liczba nie jest palindromem')
        end
        disp('Podaj kolejną liczbę lub wpisz NIE/N, aby wyjść : ')
        d = input('odpowiedź : ', 's');
        if(d(1)~= "0" & str2num(d) > 0 & mod(str2num(d),1) == 0)
            p(d)
        elseif(lower(d) == "n" | lower(d) == "nie" )
            disp('Dziękujemy za skorzystanie z programu!')
        end

        else
            disp('Błąd - podana liczba nie jest naturalna lub jest napisem')
            z = input('Wprowadz poprawną: ', 's')
            p(z)
        end
    end
end
```

## Przykłady wywołań

```
>> palindrom
```

```
Wprowadź liczbę: 000000000
```

```
x =
```

```
'000000000'
```

```
Błąd - podana liczba nie jest naturalna lub jest napisem
```

```
Wprowadz poprawną wartość: 1234567890
```

```
z =
```

```
'1234567890'
```

```
Liczba nie jest palindromem
```

```
Podaj kolejną liczbę lub wpisz NIE/N, aby wyjść : |  
odpowiedź : 212
```

```
Liczba jest palindromem
```

```
Podaj kolejną liczbę lub wpisz NIE/N, aby wyjść :  
odpowiedź : 19391
```

```
Liczba jest palindromem
```

```
Podaj kolejną liczbę lub wpisz NIE/N, aby wyjść :  
odpowiedź : PIOTR
```

```
Błąd - podana liczba nie jest naturalna lub jest napisem
```

```
Wprowadz poprawną wartość: EWA
```

```
z =
```

```
'EWA'
```

```
Błąd - podana liczba nie jest naturalna lub jest napisem
```

```
Wprowadz poprawną wartość: 33
```

```
z =
```

```
'33'
```

```
Liczba jest palindromem
```

```
Podaj kolejną liczbę lub wpisz NIE/N, aby wyjść :  
odpowiedź : nie
```

```
Dziękujemy za skorzystanie z programu!
```

## Na zakończenie ciekawostka

Polska może się poszczycić rekordem jeśli chodzi o najdłuższy palindrom. Na prowadzonej przez siebie **stronie Tadeusz Morawski** umieścił najdłuższy na świecie palindrom, mający ponad 33 tysiące liter! Dotychczas najdłuższym był palindrom angielski mający około 17 tys. liter.

Dziękujemy serdecznie za uwagę,  
***Piotr Szyszka, Ewa Podlowska.***