



Palindromy

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Schemat interaktywny](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Może jutro ta dama sama da tortu jeżom, Ada biada i bada, A to kanapa pana Kota – co łączy te zdania? Brzmią tak samo czytane od lewej strony, jak i od prawej. To palindromy.

Patrząc na słowo, możemy bez większego problemu określić, czy jest ono palindromem, czy nie. Jednak w jaki sposób wytłumaczyć tok naszego rozumowania komputerowi i nauczyć go sprawdzania, czy podane słowo jest palidromem? Może okazać się to przydatne, gdy przyjdzie nam analizować palindromy wielowyrazowe tj.: *Ada raportuje, że jutro parada*, czy liczący ponad 33 tysiące liter tekst prof. Tadeusza Morawskiego *Żartem w metraż*.

Implementację programu sprawdzającego, czy dane słowo jest palindromem w wybranych językach programowania znajdziesz w e-materiałach:

- [Palindromy w języku C++](#),
- [Palindromy w języku Java](#),
- [Palindromy w języku Python](#).

Więcej zadań? Przejdź do e-materiału [Palindromy – zadania maturalne](#).

Twoje cele

- Przeanalizujesz dwa algorytmy sprawdzania, czy podane słowo jest palindromem oraz wskażesz różnice w ich działaniu.

- Zapiszesz, przy użyciu schematu blokowego, algorytm sprawdzania, czy podane słowo jest palindromem.
- Rozwiążesz kilka prostych zadań związanych z palindromami.

Przeczytaj

Zastanówmy się w jaki sposób działa oraz jak wygląda algorytm, który jednoznacznie określi, czy podane słowo jest **palindromem**, czy też nie?

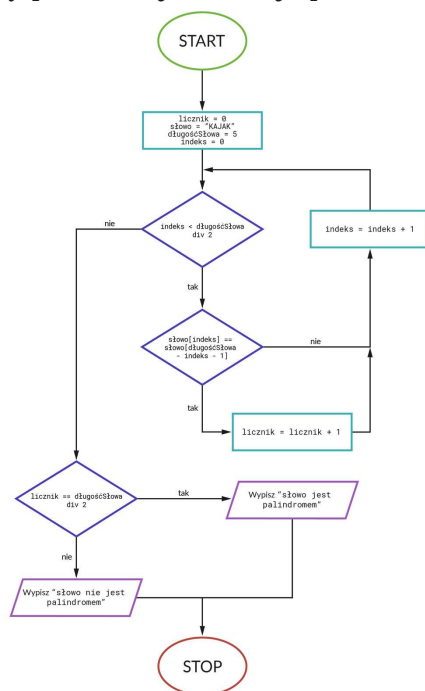
Przeanalizujemy dwa rozwiązania.

Metoda pierwsza

Aby sprawdzić, czy podane słowo jest palindromem, możemy porównywać pary liter. Pierwsza para to pierwsza i ostatnia litera w słowie. Jeżeli dwie litery są takie same, zwiększymy licznik oznaczający liczbę poprawnych par liter o 1. Następną parą będzie druga i przedostatnia litera w słowie. Jeżeli litery są sobie równe, znów zwiększamy licznik o 1. I tak będziemy wykonywać sprawdzanie, aż dojdziemy do środka wyrazu, czyli aż wykonamy: $(\text{długośćSłowa}) \div 2$ sprawdzeń.

Zwróć uwagę, że korzystając z polecenia `div`, nie ma znaczenia, czy słowo ma parzystą, czy nieparzystą liczbę znaków.

Oto schemat blokowy realizujący pierwszą metodę sprawdzania.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Zwróć uwagę na wyrażenie logiczne:

`słowo[indeks] == słowo[długośćSłowa - indeks - 1]`.

Pierwszym ważnym aspektem jest sposób odwoływania się do konkretnych liter w słowie. W tym przypadku używamy nawiasów kwadratowych. Przykładowo: `słowo[0]` oznacza

pierwszą literę w słowie.

Jak wcześniej wspomnieliśmy będziemy porównywać pary znaków. Tę operację najłatwiej wykonać wewnątrz pętli. Będzie się ona wykonywać dopóki indeks jest mniejszy od połowy długości słowa. Wewnątrz pętli sprawdzamy czy znak pobrany z pierwszej połowy słowa (`słowo[indeks]`) jest taki sam jak jego odpowiednik z drugiej połowy (`słowo[długośćSłowa - indeks - 1]`). Zwróćmy uwagę w jaki sposób wyznaczamy indeks drugiego znaku. Od `długośćSłowa` odejmujemy `indeks`, a także wartość `-1`. Operacja ta wynika z faktu, że znaki w `słowo` numerujemy od zera.

Przeanalizuj, jak działa schemat blokowy. Zapisuj ulegające zmianie wartości zmiennych. Jeżeli masz problem, w następnej sekcji dokonamy wspólnej analizy algorytmu.

Metoda druga

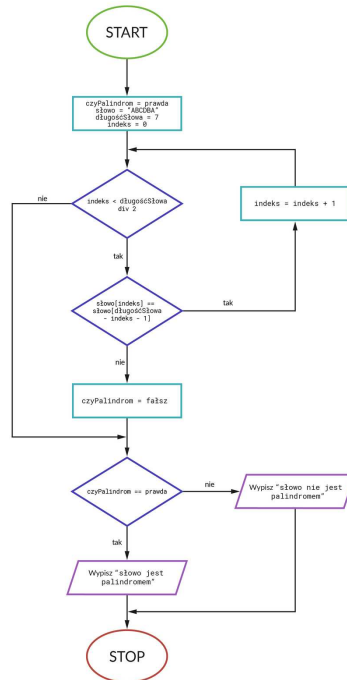
Bardziej optymalna metoda na sprawdzenie, czy podane słowo jest palindromem, wykorzystuje to, że odnalezienie jednej pary liter, które nie są sobie równe (a powinny być), automatycznie powoduje, że podane słowo na pewno nie jest palindromem.

Na początku możemy założyć, że mamy do czynienia ze słowem, które jest palindromem, a gdy natrafimy na wykluczającą się parę, zakończymy algorytm z informacją, że słowo nie jest palindromem.

Zbadajmy, czy słowo „ABCDEEDXBA” jest palidromem. Najpierw sprawdzone zostaną dwie litery 'A': pierwsza i ostatnia w słowie. Kolejną sprawdzaną parą będą litery 'B' (druga i przeostatnia w słowie). Następnie sprawdzone zostaną litery: 'C' oraz 'X' (trzecia od początku i trzecia od końca litera). Już wiemy, że nie ma potrzeby dalszego sprawdzania liter. Słowo na pewno nie jest palindromem, skoro zawiera co najmniej jedną parę liter, które powinny być takie same, a nie są. I właśnie w tym momencie algorytm zakończy porównywanie z informacją, że podane słowo nie jest palindromem.

Gdyby podane słowo było równe „ABCCBA”, program dokonałby trzech sprawdzeń i zakończył działanie z informacją, że podane słowo jest palindromem (program nie natrafił na parę różnych liter).

Poniżej przedstawiony został schemat blokowy realizujący drugą metodę sprawdzania:



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

W przedstawionej metodzie wykorzystaliśmy zmienną `czyPalindrom`, która na początku inicjowana jest wartością `prawda`. Jeżeli natrafimy na parę znaków, które nie są takie same (a powinny być), zmieniamy wartość `czyPalindrom` na `fałsz`. Następnie algorytm kończy się z informacją, że podane słowo nie jest palindromem.

Słownik

`div`

połączenie często używane jako symbol dzielenia całkowitego, np. $7 \text{ div } 2 = 3$.

`palindrom`

wyrażenie brzmiące tak samo czytane od lewej do prawej i od prawej do lewej

Schemat interaktywny

Ćwiczenie 1

Napisz program sprawdzający, czy dane słowo jest palindromem. Wykorzystaj do tego pierwszą metodę.

Zapisz algorytm, wykorzystując wybrany język programowania.

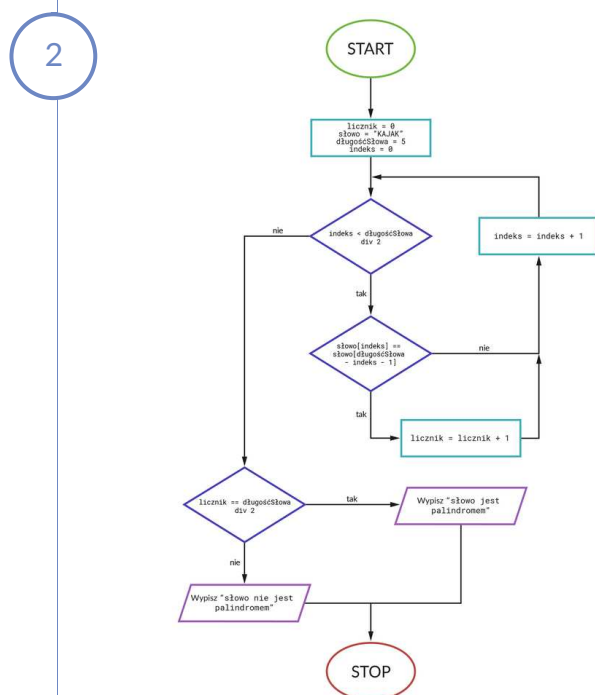
1

1

Polecenie 1

Przeanalizuj prezentację. Porównaj z nią swoje rozwiązanie.

Przeanalizujemy, w jaki sposób zachowa się pierwszy algorytm. Dla ułatwienia analizy umieściliśmy schemat blokowy w kolejnym kroku.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Schemat blokowy przedstawiający pierwszą metodę sprawdzania:

Aktualne wartości zmiennych:



1

3

```
1 licznik = 0
2 słowo = "KAJAK"
3 długośćSłowa = 5
4 indeks = 0
```

4

Podążając zgodnie z kierunkiem strzałek, kolejnym blokiem jest instrukcja warunkowa. Sprawdzamy, czy $\text{indeks} < \text{długośćSłowa} \div 2$. Podstawiamy aktualne wartości zmiennych i otrzymujemy warunek: $0 < 2$, który jest prawdziwy. Zatem przechodzimy do bloku z kolejną instrukcją warunkową.

Następny warunek to: $\text{słowo}[\text{indeks}] == \text{słowo}[\text{długośćSłowa} - \text{indeks} - 1]$. Po podstawieniu wartości zmiennych otrzymujemy: $\text{słowo}[0] == \text{słowo}[5 - 0 - 1]$, czyli sprawdzamy, czy pierwsza litera słowa jest równa ostatniej. Zgadza się, pierwsza litera słowa ('K') jest równa ostatniej ('K'), zatem warunek został spełniony.

5

6

Kolejnym blokiem jest $\text{licznik} = \text{licznik} + 1$ - aktualna wartość zmiennej `licznik` wynosi 1.

Po zwiększeniu zmiennej `licznik` przechodzimy do bloku $\text{indeks} = \text{indeks} + 1$, który zmienia wartość zmiennej `indeks` na 1.

Następnie wracamy do instrukcji warunkowej na początku pętli. Wykonujemy ją ponownie. Sprawdzamy, czy `indeks < długośćSłowa div 2`. Podstawiając aktualne wartości zmiennych, otrzymujemy warunek: $1 < 2$, który jest prawdziwy. Zatem przechodzimy po raz kolejny do drugiej instrukcji warunkowej.

7

8

Instrukcja ta sprawdza, czy: `słowo[indeks] == słowo[długośćSłowa - indeks - 1]`. Po podstawieniu wartości zmiennych otrzymujemy: `słowo[1] == słowo[5 - 1 - 1]`. Sprawdzamy drugą literę słowa ('A') i przedostatnią ('A'), które okazują się równe sobie.

Kolejnym krokiem jest ponowne przejście do instrukcji zwiększającej licznik. Po wykonaniu danego bloku wartość zmiennej `licznik` wynosi 2. Następnie wykonuje się zwiększenie zmiennej `indeks` do wartości 2.

9

10

Ponownie wracamy do pierwszej instrukcji warunkowej: `indeks < długośćSłowa div 2`. Gdy podstawimy aktualne wartości zmiennych: $2 < 5 \text{ div } 2$, widzimy, że warunek nie jest spełniony, i musimy przejść do kolejnego bloku – instrukcji warunkowej.

11

Sprawdzamy, czy `licznik == długośćSłowa div 2`, czyli w tym przypadku, czy `2 == 5 div 2`. Warunek jest spełniony, więc przechodzimy do instrukcji `wypisz "słowo jest palindromem"`. W ten sposób prawidłowo przeanalizowaliśmy działanie pierwszego algorytmu na sprawdzanie, czy podane słowo jest palindromem.

12

Teraz twoja kolej. Sprawdź, czy dowolne słowo jest palindromem. Na początek dokonaj analizy, korzystając z pierwszego algorytmu. Następnie sprawdź, czy drugi algorytm zadziała w taki sam sposób.

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Dla zainteresowanych

Jeśli chcesz zmierzyć się z trudniejszym przykładem, wykonaj Polecenie 2.

Polecenie 2

Zapisz program wyszukujący wszystkie palindromy zawarte w danym ciągu znaków. Jeśli chcesz powtórzyć ze szkoły podstawowej wiadomości na temat programowania za pomocą bloków, wykorzystaj blockly. W przeciwnym razie program zapisz, wykorzystując język programowania.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Wyjaśnij, czym jest palindrom.

Ćwiczenie 2



Oceń, czy zdanie jest prawdziwe.

Istnieje tylko jedna metoda na sprawdzanie, czy podane słowo jest palindromem.

fałsz

prawda

Ćwiczenie 3



Wskaż, które z przedstawionych słów to palindromy. Zaznacz wszystkie poprawne odpowiedzi.

Oko

Auto

Agi

Inni

Ćwiczenie 4

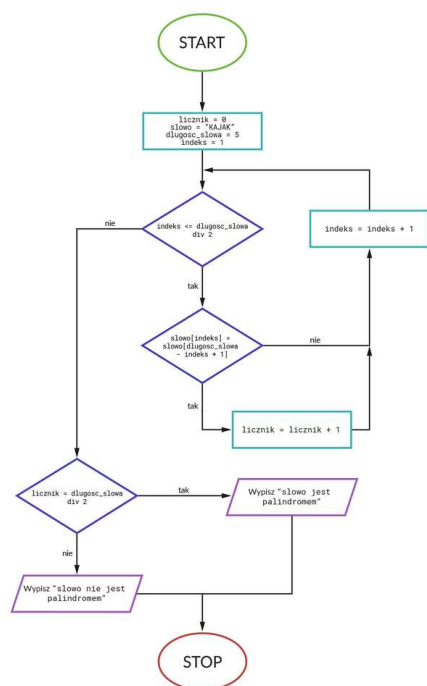


Wyjaśnij różnicę pomiędzy algorytmem I i algorytmem II.

Ćwiczenie 5



Wykorzystując pierwszy algorytm sprawdzenia, czy podane słowo jest palindromem, określ, niezależnie od początkowej wartości zmiennej `słowo`, ile razy wykona się instrukcja warunkowa `licznik = długość_słowa div 2`.



Tyle razy, ile wynosi wartość zmiennej `długość_słowa`

0

1

2

Ćwiczenie 6



Korzystając z pierwszego algorytmu (widocznego w poprzednim ćwiczeniu), określ – niezależnie od wartości zmiennej `słowo` – ile razy wykona się blok z instrukcją `licznik = licznik + 1`.

Tyle razy, ile instrukcja warunkowa `słowo[indeks] = słowo[długość_słowa - indeks + 1]` okaże się prawdziwa

Tyle razy, ile wynosi wartość zmiennej `długość_słowa`

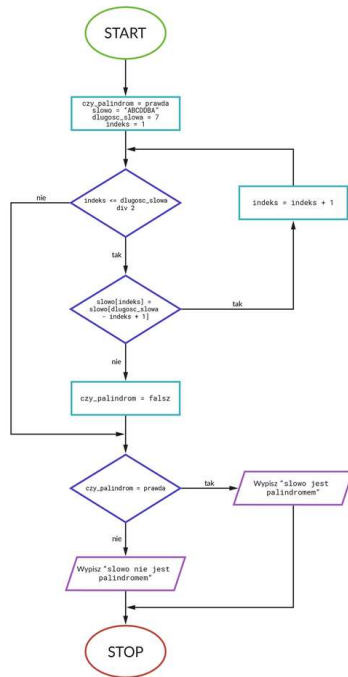
1

2

Ćwiczenie 7



Wykorzystując drugi algorytm sprawdzania, czy podane słowo jest palindromem, określ, jaka będzie wartość zmiennej `czy_palindrom` po **jednokrotnym** wykonaniu bloku z instrukcją `indeks <= długość_słowa div 2`.



prawda

fałsz

żadna z powyższych

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Ćwiczenie 8



Wykorzystując drugi algorytm sprawdzania, czy podane słowo jest palindromem (zobacz

Ćwiczenie nr 7), określ, jaką wartość będzie miała zmienna `indeks`, po przejściu przez cały algorytm.

7

4

1

3

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Palindromy

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy:

b) na tekstach: porównywania tekstów, wyszukiwania wzorca w tekście metodą naiwną, szyfrowania tekstu metodą Cezara i przestawieniową,

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Przeanalizujesz dwa algorytmy sprawdzania, czy podane słowo jest palindromem oraz wskażesz różnice w ich działaniu.
- Zapiszesz, przy użyciu schematu blokowego, algorytm sprawdzania, czy podane słowo jest palindromem.

- Rozwiążesz kilka prostych zadań związanych z palindromami.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Palindromy”. Uczniowie zapoznają się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Wyświetlenie przez nauczyciela tematu i celów zajęć, przejście do wspólnego ustalenia kryteriów sukcesu.

Faza realizacyjna:

1. Uczniowie podają przykłady prostszych i trudniejszych palindromów.
2. **Praca z multimediu.** Nauczyciel wyświetla zawartość sekcji „Schemat interaktywny”, czyta treść polecenia nr 1: „Przeanalizuj prezentację. Porównaj z nią swoje rozwiązanie (program sprawdzający, czy dane słowo jest palindromem, napisany z wykorzystaniem pierwszej metody)” i omawia kolejne kroki rozwiązania.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują ćwiczenia nr 1-8 z sekcji „Sprawdź się”. Nauczyciel sprawdza poprawność wykonanych zadań, omawiając je wraz z uczniami.

Faza podsumowująca:

1. Na koniec zajęć nauczyciel raz jeszcze wyświetla na tablicy temat lekcji i cele zawarte w sekcji „Wprowadzenie”. W odniesieniu do ich realizacji dokonuje szczegółowej oceny rozwiązania zastosowanego przez wybranego ucznia.
2. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności.

Praca domowa:

1. Uczniowie opracowują FAQ (minimum 3 pytania i odpowiedzi) do tematu lekcji („Palindromy”).
2. Napisz algorytm, który sprawdza, czy dany wyraz jest palindromem parzystym (ma parzystą liczbę liter).

Wskazówki metodyczne:

- Uczniowie mogą wykorzystać treści w sekcjach: „Przeczytaj”, „Schemat interaktywny”, „Sprawdź się” jako materiał do lekcji powtórkowej.