

## Temat: Podstawy algorytmiki.

Przeanalizuj następujące zagadnienia:

Podstawy algorytmiki to temat obejmujący teorię algorytmów oraz praktyczne aspekty ich implementacji i analizy. Oto kilka kluczowych punktów:

### 1. Definicja algorytmu

Algorytm to skończony zbiór kroków prowadzących do rozwiązania określonego problemu. Algorytmy mogą być przedstawione w różnych formach, takich jak:

- Pseudokod.
- Diagramy przepływu.
- Programy w określonym języku programowania.

### 2. Cechy dobrego algorytmu

Dobry algorytm powinien wyróżniać się następującymi cechami:

- Skończoność: powinien zakończyć się po skończonej liczbie kroków.
- Jednoznaczność: każdy krok powinien być jasno określony.
- Efektywność: powinien być zaprojektowany w taki sposób, aby zużywał minimalne zasoby (czas, pamięć).
- Poprawność: wykonuje określone zadania.
- Wejście i wyjście: powinien mieć co najmniej jedno wejście i co najmniej jedno wyjście.

### 3. Typy algorytmów

- Algorytmy sortowania (np. sortowanie bąbelkowe, szybkie sortowanie, sortowanie przez wstawianie).
- Algorytmy wyszukiwania (np. wyszukiwanie liniowe, wyszukiwanie binarne).
- Algorytmy grafowe (np. algorytm Dijkstry, DFS, BFS).
- Algorytmy dynamiczne (np. problem plecakowy, Fibonacciego).

### 4. Złożoność algorytmów

Złożoność algorytmu określa, jak jego czas wykonania lub użycie pamięci wzrasta w zależności od wielkości danych wejściowych. Dzielimy ją na:

- Złożoność czasową: czas potrzebny do wykonania algorytmu.
- Złożoność pamięciową: ilość pamięci potrzebnej do wykonania algorytmu.

Złożoność jest często wyrażana za pomocą notacji asymptotycznej, takiej jak  $O(n)$ ,  $\Theta(n)$  czy  $\Omega(n)$ .

### 5. Analiza algorytmu

Analiza algorytmu polega na badaniu jego złożoności czasowej i pamięciowej, co pozwala ocenić jego efektywność. Zazwyczaj przeprowadza się analizę najlepszego, najgorszego oraz przeciętnego przypadku.

### 6. Implementacja algorytmu

Implementacja algorytmu polega na napisaniu go w konkretnym języku programowania. Ważne jest, aby być świadomym specyfiki danego języka, ponieważ może to wpływać na wydajność i złożoność algorytmu.

### 7. Przykłady praktyczne

Rozwiązywanie rzeczywistych problemów przy użyciu algorytmów, np.:

- Jak zoptymalizować trasę dostaw (problem najkrótszej drogi).
- Jak zorganizować dane do efektywnego wyszukiwania.

### 8. Metody zapisu algorytmów:

- Opis słowny.
- Lista kroków.
- Schemat blokowy.
- Drzewo algorytmiczne.
- Pseudokod programowy.
- Program w danym języku programowania.

### 9. Narzędzia i zasoby

Istnieje wiele narzędzi i bibliotek ułatwiających pracę z algorytmami, jak na przykład:

- Scratch – edukacyjny wizualny język programowania.
- Draw.io – narzędzie do tworzenia diagramów dostępne online.
- Jupyter Notebook dla Pythona.
- Biblioteki C++ (STL).
- Algorytmy gotowe do użycia w różnych językach programowania.

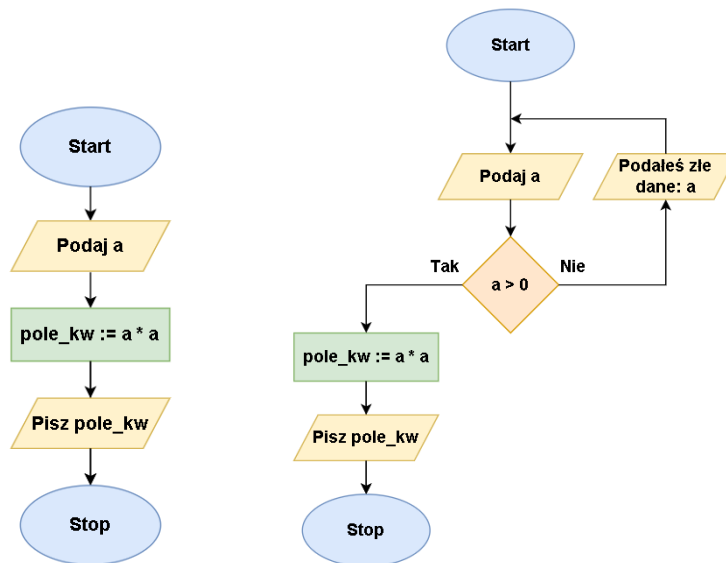
Wykonaj zadanie:

Zapoznaj się z programem **scratch**: <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

Wykonaj zadanie:

Zapoznaj się z oprogramowaniem dostępnym na stronie: <https://app.diagrams.net>

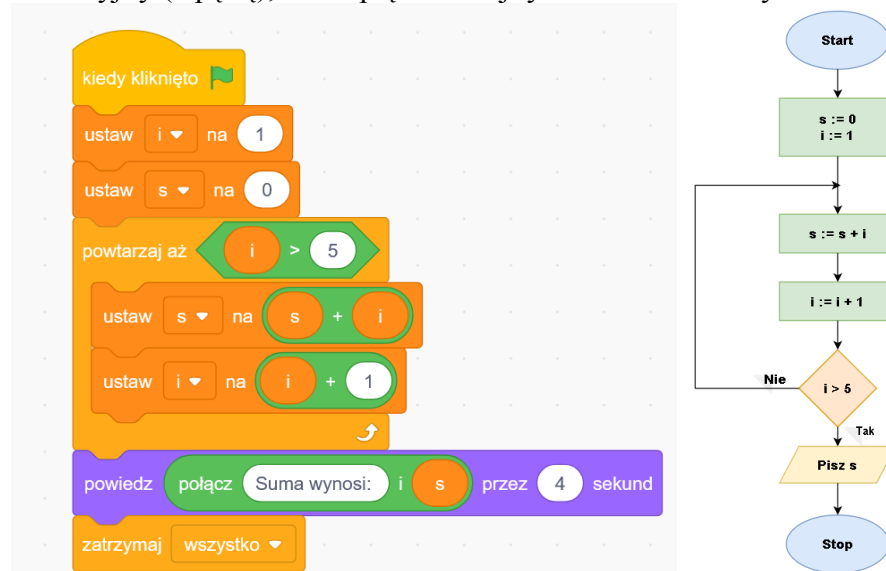
Przykład 1: Algorytm sekwencyjny i z rozgałęzieniem, do obliczania pola powierzchni kwadratu.



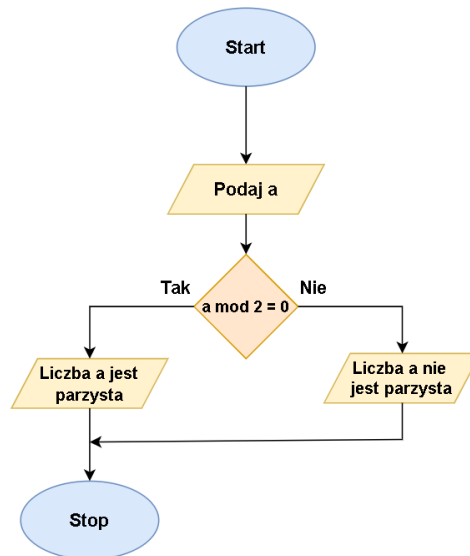
Przykład 2: Algorytm sekwencyjny, suma dwóch liczb całkowitych podanych przez użytkownika.



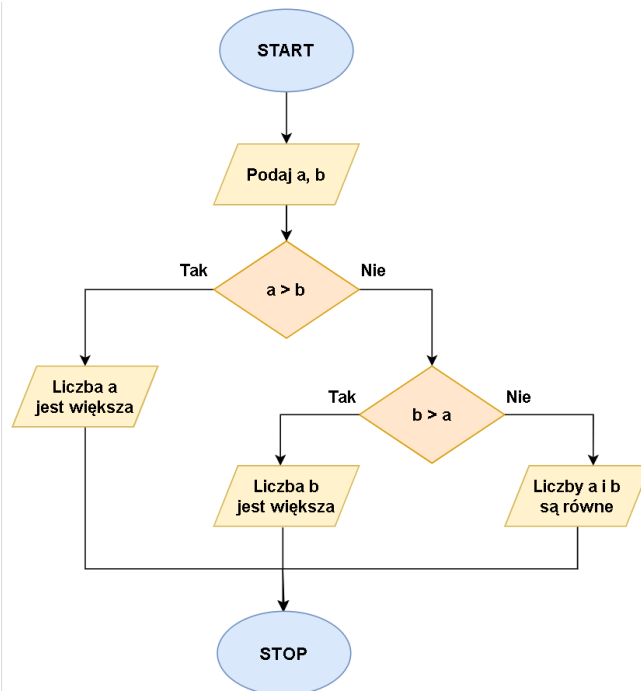
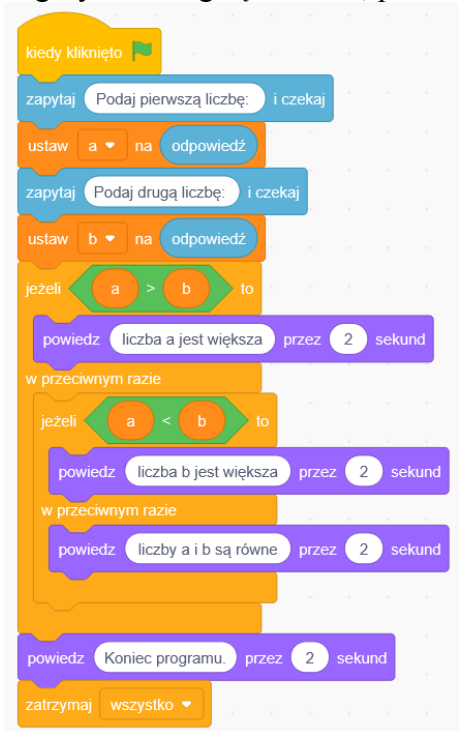
Przykład 3: Algorytm iteracyjny (z pętlą), suma pięciu kolejnych liczb całkowitych.



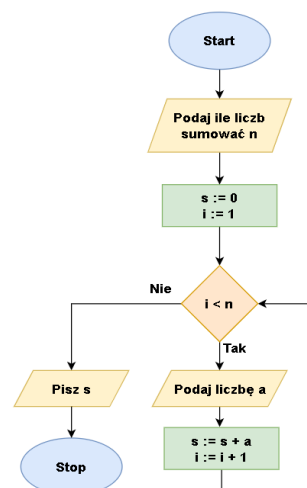
Przykład 4: Algorytm z rozgałęzieniem, sprawdzający, czy podana przez użytkownika liczba jest parzysta.



Przykład 5: Algorytm z rozgałęzieniami, porównanie dwóch liczb podanych przez użytkownika.

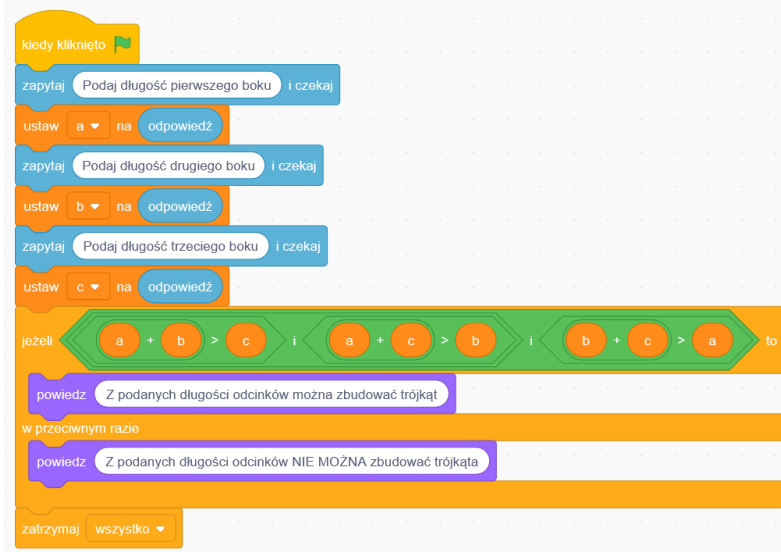


Przykład 4: Suma tulu liczb i takich, jakie poda użytkownik.



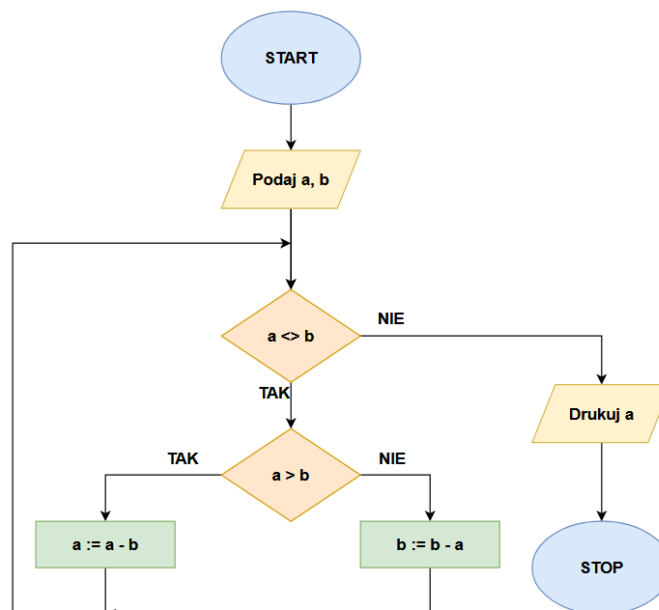
Wykonaj zadanie:

Wykorzystując serwis **scratch.mit.edu** utwórz algorytm sprawdzający, czy z podanych trzech długości odcinków można zbudować trójkąt. Pracę zapisz w pliku pod nazwą **\$nazwisko\_\$klasa\_\$gr\_trojkat** i prześlij do nauczyciela w postaci załącznika na adres [greszata@zs9elektronik.pl](mailto:greszata@zs9elektronik.pl).



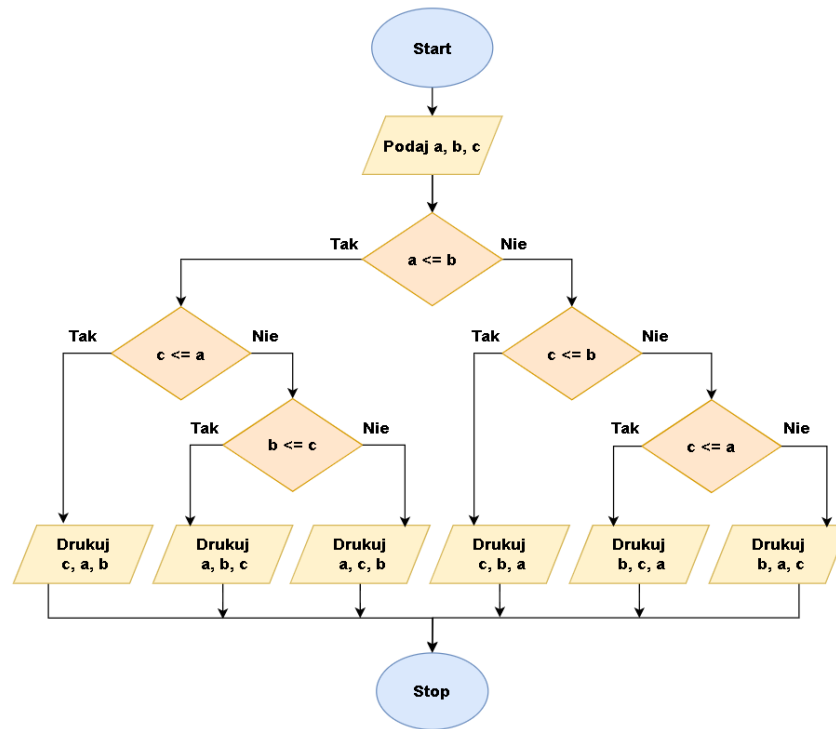
Wykonaj zadanie:

Wykorzystując serwis **diagrams.net** utwórz algorytm Euklidesa, znajdujący największy wspólny dzielnik (NWD) dwóch liczb naturalnych. Pracę zapisz w pliku pod nazwą **\$nazwisko\_\$klasa\_\$gr\_euklides.html** i prześlij do nauczyciela w postaci załącznika na adres [greszata@zs9elektronik.pl](mailto:greszata@zs9elektronik.pl).



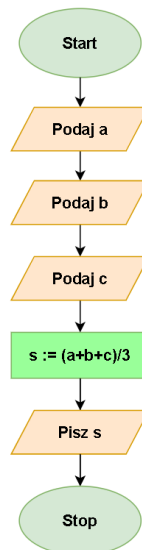
Wykonaj zadanie:

Wykorzystując serwis **diagrams.net** utwórz algorytm sortujący rosnąco trzy liczby podane przez użytkownika. Pracę zapisz w pliku pod nazwą **\$nazwisko\_\$klasa\_\$gr\_sortowanie.html** i prześlij do nauczyciela w postaci załącznika na adres [greszata@zs9elektronik.pl](mailto:greszata@zs9elektronik.pl).

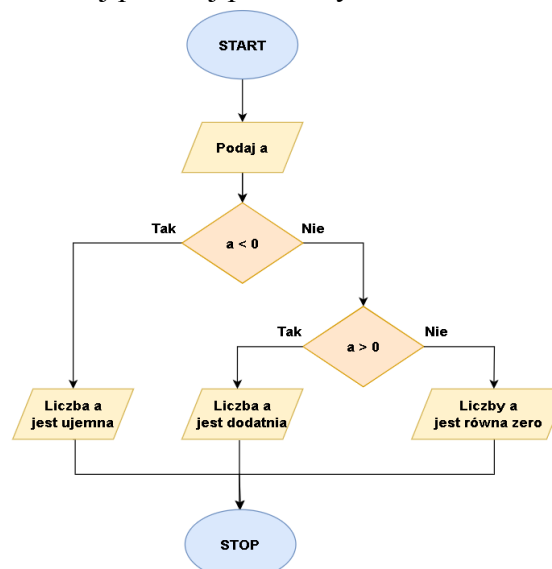


Inne przykłady algorytmów.

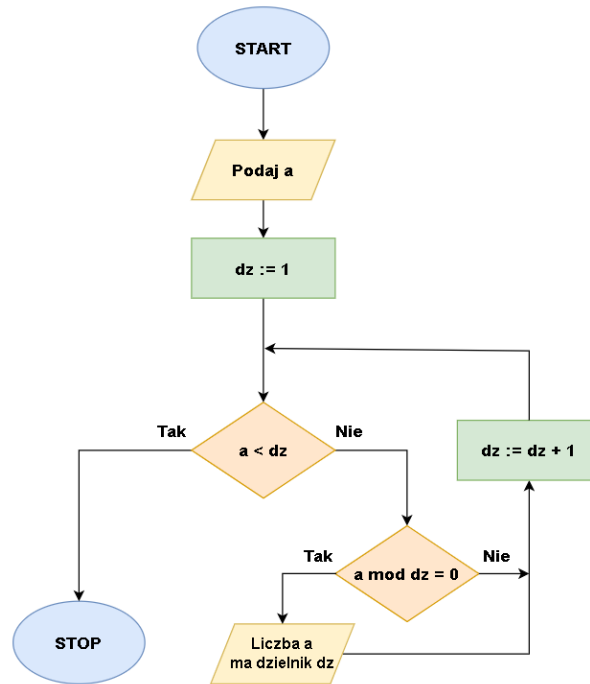
Algorytm sekwencyjny, obliczający średnią arytmetyczną trzech liczb podanych przez użytkownika:



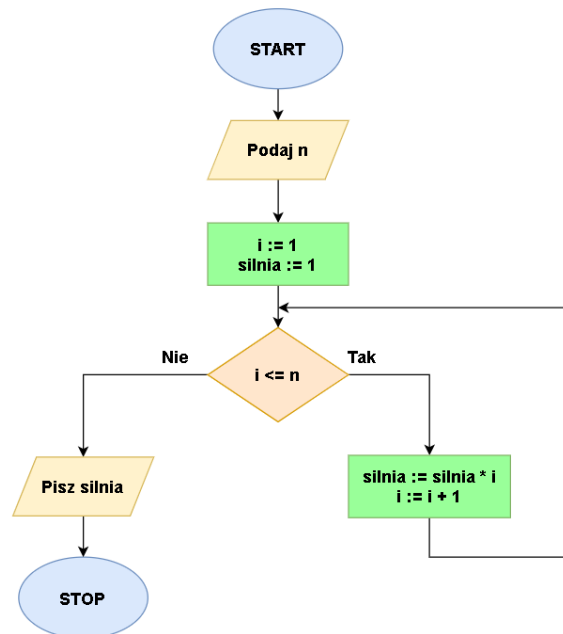
Algorytm sprawdzający wartość zmiennej podanej przez użytkownika:



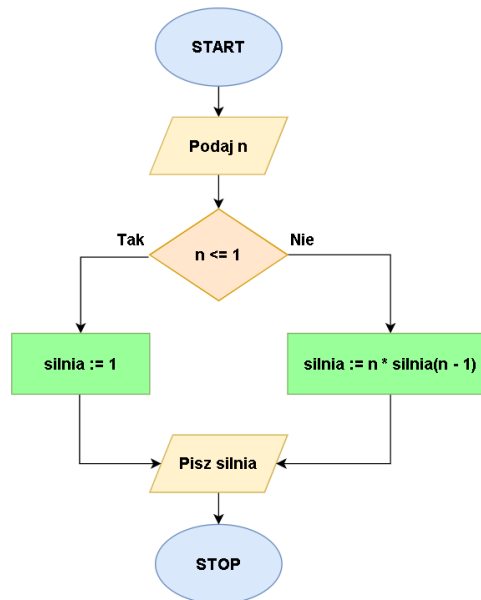
Algorytm iteracyjny obliczający silnię liczby podanej przez użytkownika:



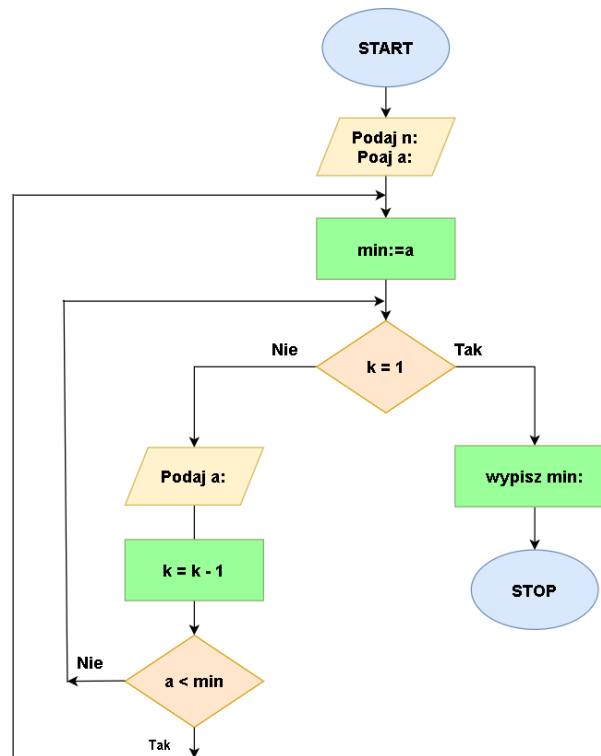
Algorytm iteracyjny obliczający silnię liczby podanej przez użytkownika:



Algorytm rekurencyjny obliczający silnię liczby podanej przez użytkownika:



Algorytm wyszukujący najmniejszą liczbę z podanych przez użytkownika:



Zgadywanie przez użytkownika liczby całkowitej wylosowanej przez algorytm z zakresu 1..100:

```
whenClickedFlagClicked
  sayHello "Cześć!" 2 seconds
  hideVariable los
  set los to random 1 100
  set odp to 0
  set ile to 0
  think "Losowanie liczby..." 2 seconds
  repeatUntil (odp = los)
    ask "Zgadnij, jaką wylosowano liczbę?" and wait
    set odp to answer
    set ile to ile + 1
    if (odp > los) then
      say "Wylosowana liczba jest mniejsza." 2 seconds
    otherwise
      if (odp < los) then
        say "Wylosowana liczba jest większa." 2 seconds
      otherwise
        say "Połączył Podales wylosowaną liczbę. Zgadywałeś i połączył ile i razy." 3 seconds
  stopAllSounds
```

The image shows a Scratch script for a number-guessing game. The script starts with a 'when clicked' event, followed by a 'say' block with the text 'Cześć!' for 2 seconds. It then hides a variable named 'los', sets 'los' to a random number between 1 and 100, sets 'odp' to 0, and sets 'ile' to 0. A 'think' block with the text 'Losowanie liczby...' for 2 seconds follows. The main loop is a 'repeat until' block with the condition 'odp = los'. Inside this loop, there is an 'ask' block with the text 'Zgadnij, jaką wylosowano liczbę?' and a 'wait' block. The 'ask' block is connected to a 'set' block that sets 'odp' to the 'answer' variable. This is followed by a 'set' block that increments 'ile' by 1. There are two 'if' blocks: one for 'odp > los' which says 'Wylosowana liczba jest mniejsza.' for 2 seconds, and another for 'odp < los' which says 'Wylosowana liczba jest większa.' for 2 seconds. A 'w przeciwnym razie' (otherwise) block contains a 'say' block with the text 'Połączył Podales wylosowaną liczbę. Zgadywałeś i połączył ile i razy.' for 3 seconds. The script ends with a 'stop all sounds' block.