



Nazwa implementacji:

**Sortowanie Autor: agata Opis implementacji:**

Zostajecie podzieleni na 3-5 osobowe zespoły. Otrzymujecie od nauczyciela 5 kartek z różnymi liczbami ułożonymi w przypadkowej kolejności. Ułóżcie w grupie 5 kartek w porządku rosnącym. Wykorzystajcie dowolną ilość ruchów, ale pamiętajcie, że w każdym ruchu można zmienić tylko dwie kartki.

Ponowie pomieszajcie kartki. Spróbujcie teraz wykonać ćwiczenie jak najmniejszą liczbą ruchów kartek. Ostatni raz ułóżcie kartki w przypadkowej kolejności. Kiedy skończycie, spróbujcie jeszcze raz wykonać zadanie, w każdym ruchu zmieniając tylko dwie sąsiednie kartki. Z pomocą nauczyciela stwórzcie aplikację w programie Scratch realizującą sortowanie metodą „bąbelkową”.

```

kiedy kliknięto Duzek1
ustaw na na 0
ustaw na na 0
ustaw temp na 0
usuń wszystko z liczby

powiedz Sortowanie bąbelkowe ("bubble sort") polega na porównywaniu dwóch kolejnych elementów zbioru. Jeżeli porównywane elementy znajdują się w niekłaśowej kolejności, są zamieniane miejscami. przez 10
powiedz Podczas sortowania n-elementów zbioru, algorytm wykonuje n przejść, a w każdym przejściu n-1 porównań. przez 5
powiedz teraz wylosuj 5 liczb do posortowania przez 2
powtórz 5 razy
czekaj 1
dodaj losuj liczbę pomiędzy 1 = 100 do liczby
powiedz teraz posortuj wylosowane liczby przez 3
ustaw na na 1
powtórz 4 razy
powiedz połącz i i przejście algorytmu przez 1
ustaw na na 1
powtórz 4 razy
powiedz połącz połącz porównania i połącz i + 1 i połącz i + 1 i elementu listy przez 1
jeżeli element i z liczby > element i + 1 z liczby
powiedz połącz połącz zamiana i połącz i i połącz i + 1 i elementu listy przez 1
ustaw temp na element i z liczby
zamień element i z liczby na element i + 1 z liczby
zamień element i + 1 z liczby na temp
zmień o 1
czekaj 1
zmień o 1
czekaj 1
powiedz Gotowe! przez 2
zatrzymaj wszystko

```

Przetestuj działanie programu.

Następnie stwórz aplikację w programie Scratch realizującą sortowanie metodą „przez wybór”.





```

kiedy kliknięto Duszek1.
usun wszystko z lista
powiedz: Sortowanie "przez wstawianie" polega na porównaniu kolejnego elementu ze zbioru z już uporządkowanymi i wstawieniu go na właściwej pozycji. przez 10 s
powiedz: Wylosuję do posortowania 5 liczb. przez 2 s
powtórz 5 razy
  dodaj losuj liczbę pomiędzy 1 a 100 do lista
  powiedz: Elementy zbioru ustawię rosnąco. przez 2 s
  powiedz: Jeżeli wybrany do porównania element będzie mniejszy od poprzednika, zamienię je miejscami. przez 5 s
  powiedz: Proces porównywania będzie trwał tak długo, aż wybrany element trafi na właściwe miejsce. przez 5 s
  powiedz: Porównywanie zaczynam od drugiego elementu listy. przez 3 s
  ustaw i na 2
  powtarzaj aż i > 5
    ustaw k na i
    powiedz: połącz połącz wybieram i i i element. przez 2 s
    powiedz: połącz i porównam go z i k - 1. przez 3 s
    powtarzaj aż k = 1 lub element k z lista > element k - 1 z lista
      powiedz: połącz Zamiana elementu na pozycji i połącz k i połącz 2 i k - 1. przez 2 s
      ustaw temp na element k z lista
      zamień element k z lista na element k - 1 z lista
      zamień element k - 1 z lista na temp
      powiedz: Porównuję z poprzednimi, stosując obrane wcześniej kryterium. przez 5 s
      zmień k o -1
    zmień i o 1
  czekaj 2 s
  powiedz: Gotowe! przez 2 s
  zatrzymaj wszystko

```

Przetestuj działanie programu.

A teraz stwórz aplikację w programie Scratch realizującą sortowanie metodą „przez wstawianie”.





```

kiedy kliknięto Duszek1
ustaw n na 5
usuń wszystko z liczby
powiedz Sortowanie "przez wybór" polega na wyznaczeniu najmniejszego elementu w nieuporządkowanym zbiorze i ustawieniu go na początku. przez 5
powiedz Procedura jest powtarzana dla pozostałych nieuporządkowanych elementów, z tym, że zamiana odbywa się z pierwszym elementem podzbioru jeszcze nieuporządkowanego. przez 7
powiedz Teraz wylosuję 5 liczb do posortowania. przez 2
powtórz 5 razy
  czekaj 1 s
  dodaj losuj liczbę pomiędzy 1 = 100 do liczby
powiedz Teraz posortuję wylosowane liczby w kolejności rosnącej. przez 3
ustaw i na 1
powtarzaj aż i = 5
  ustaw j na i + 1
  powtarzaj aż j = n + 1
    powiedz połącz połącz porównanie i i połącz i i połącz j i elementu przez 3
    jeżeli element j z liczby < element i z liczby
      powiedz połącz połącz zamiana i i połącz na i połącz j i element przez 2
      ustaw temp na element i z liczby
      zamień element i z liczby na element j z liczby
      zamień element j z liczby na temp
      czekaj 1 s
    zmień j o 1
  czekaj 2 s
  zmień i o 1
  czekaj 1 s
powiedz Gotowa! przez 2
zatrzymaj wszystko
  
```

Przetestuj działanie programu.

Po skończeniu pracy zmodyfikuj kod każdej implementacji, zmieniając liczbę elementów zbioru na 7,10, 15; przetestuj działanie aplikacji. Na końcu rozbuduj kod każdej implementacji o możliwość wyboru przez użytkownika liczby elementów zbioru.

