

Projekt 12 – Lista

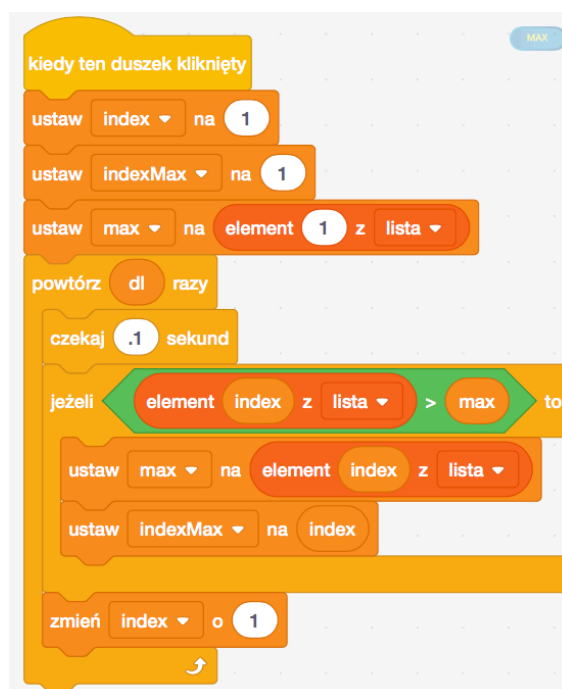
W projekcie tym zaimplementujemy proste algorytmy działające na listach.

Wstaw 6 przycisków o różnym działaniu. Każdy przycisk będzie reagował na kliknięcie:

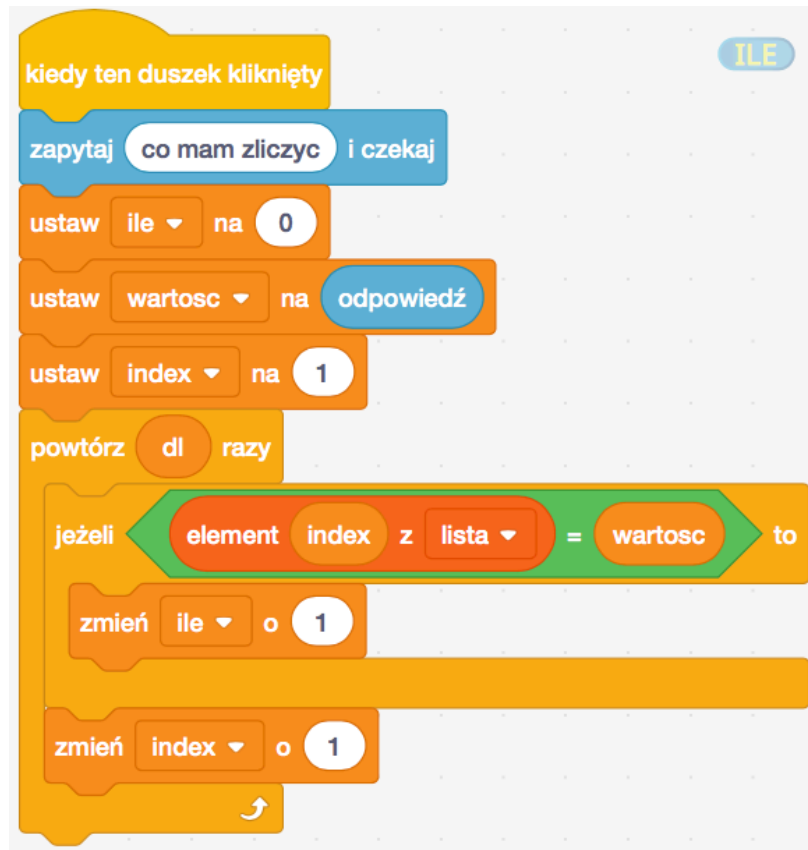
1. **Generuj** – kiedy ten duszek zostanie kliknięty, zapyta użytkownika jaka ma być długa lista. Następnie wstawi na listę podaną przez użytkownika liczbę elementów wylosowanych z zakresu od 1 do 99.



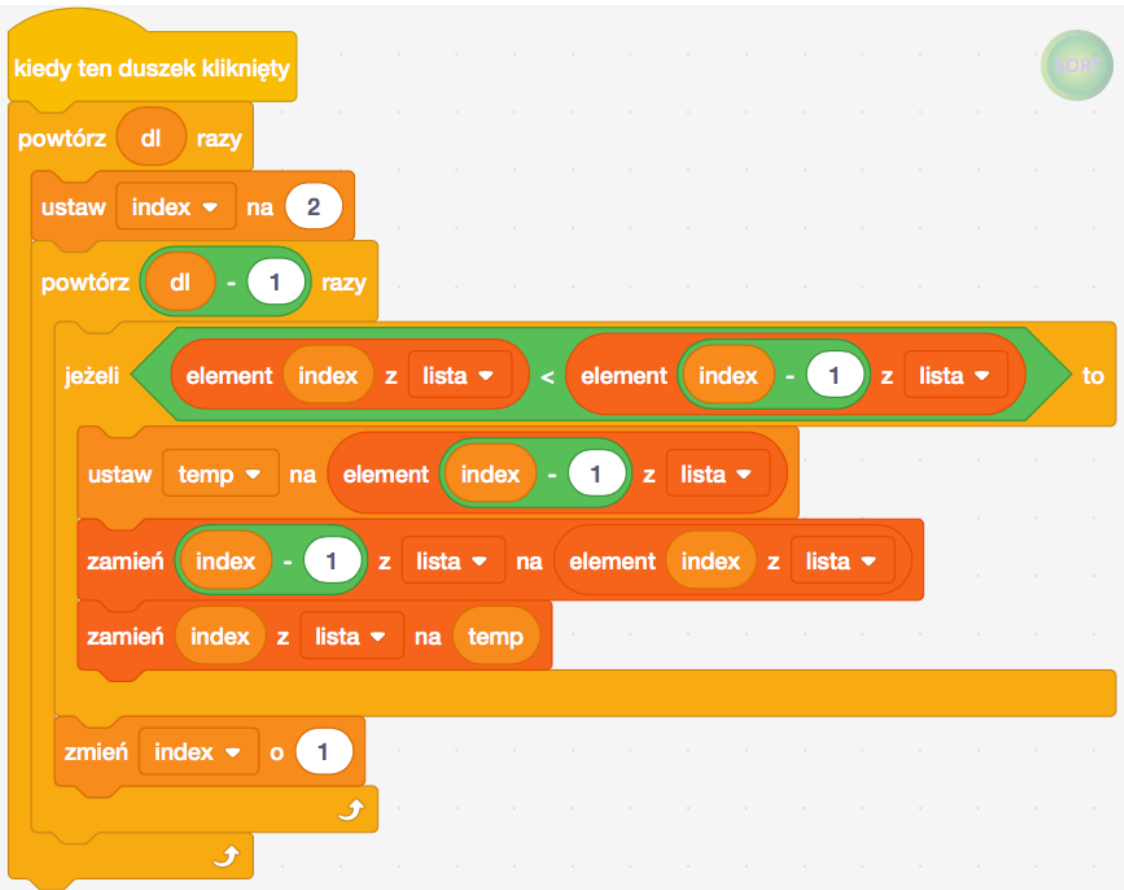
2. **MAX** – kiedy ten duszek zostanie kliknięty, zostanie wyszukany największy element na liście. Utwórz zmienną **elementMax**, w której będzie przechowywana wartość największego znalezionej do tej pory elementu z listy. Na początku zmienna **elementMax** będzie miała wartość pierwszego elementu z listy. Przejrzyj wszystkie elementy z listy. Za każdym razem, gdy napotkasz większy element, podmień na niego **elementMax**.



3. **MIN** – kiedy ten duszek kliknięty, zostanie wyszukany najmniejszy element na liście. Napisz skrypt dla tego przycisku.
4. **Czy na liście** – kiedy ten duszek kliknięty, na zostanie sprawdzone, czy podany przez użytkownika element znajduje się na liście. Napisz skrypt dla tego przycisku.
5. **ILE** – kiedy ten duszek kliknięty, zostanie policzone, ile razy podany przez użytkownika element znajduje się na liście. Utwórz zmienną **ile**. Przejrzyj całą listę. Za każdym razem, gdy napotkasz element równy wartości podanej przez użytkownika, zmień wartość **zmiennej ile o 1**.



6. **SORT** – kiedy ten duszek kliknięty, lista zostanie posortowana. Poszukaj w Internecie algorytmu sortowanie bąbelkowe. Zaimplementuj ten algorytm w Scratchu.
 - powtarzaj długość razy:
 - ustaw indeks na 2
 - powtarzaj długość – 1 razy:
 - jeżeli lista[indeks] < lista[indeks-1]:
 - zamiń lista[indeks-1] z lista[indeks]
 - zmień indeks o 1.



The image shows a Scratch script for a bubble sort algorithm. The script starts with a yellow 'when clicked' block. It then enters a 'repeat' loop with 'dl' (length) as the number of iterations. Inside the loop, the 'index' variable is set to 2. A second 'repeat' loop follows, with 'dl - 1' as the number of iterations. Within this second loop, there is an 'if' block that compares the element at 'index' with the element at 'index - 1'. If the element at 'index' is less than the element at 'index - 1', a series of blocks performs a swap: 'temp' is set to the element at 'index - 1', the element at 'index - 1' is set to the element at 'index', and the element at 'index' is set to 'temp'. After the 'if' block, the 'index' is incremented by 1. The script ends with a 'return' block.

```
kiedy ten duszek kliknięty
  powtórz dl razy
    ustaw index na 2
    powtórz dl - 1 razy
      jeżeli element index z lista < element index - 1 z lista to
        ustaw temp na element index - 1 z lista
        zamień index - 1 z lista na element index z lista
        zamień index z lista na temp
      zmień index o 1
    return
```