

1. Napisz program, który prosi użytkownika o wprowadzenie wyrazu. Następnie wyświetl wprowadzony wyraz na konsoli.

We wnętrzu programu wpisz później dodatkowo:

```
setlocale( LC_ALL, "" );  
setlocale( LC_ALL, "polish" );  
setlocale( LC_ALL, "pl_PL" );
```

i przetestuj jego działanie pod kątem wyświetlania i wczytywania polskich znaków.

2. Napisz program wczytujący ciąg 7 liczb rzeczywistych (typ float). Do każdej podanej liczby należy dodać jeden. Tak otrzymane liczby należy wydrukować na ekranie.

3. Napisz program wczytujący ciąg 12 liczb całkowitych (typ int). Liczby ujemne należy zastąpić zerami. Tak otrzymane liczby drukujemy na ekranie.

4. Napisz program generujący n początkowych liczb ciągu Fibonacciego.

a) zadeklaruj zmienną n typu int

b) poproś użytkownika o wprowadzenie liczby całkowitej większej niż 1 i zapisz ją jako n

c) jeśli n mniejsze niż 2, za pomocą while poproś jeszcze raz o podanie n

d) zadeklaruj tablice fib typu int o liczbie elementów n

e) uzupełnij tablice zgodnie z zasadami tworzenia ciągu (użyj for od trzeciego elementu), potem wypisz jej elementy na konsoli

f)\* rozbuduj program tak, aby przy n równym 0 – 1 wyświetlało poprawnie elementy ciągu

g) uruchom program i wpisz za n 100, czy może warto zmienić typ w tablicy? jeśli tak na jaki?

5. Napisz program, który poprosi użytkownika o podanie jednej liczby.

Następnie stwórz tablicę o podanej liczbie elementów i wypełnij ją losowymi liczbami całkowitymi z przedziału [-4,12].

a) zadeklaruj zmienną n typu int

b) poproś użytkownika o podanie liczby, wprowadzoną wartość zapisz jako zmienną n

c) jeśli  $n$  jest ujemne, za pomocą `while` przekaz informacje o błędzie, i wczytaj liczbę jeszcze raz

d) uzupełnij program o odpowiednie biblioteki (`#include...`) i dodaj funkcję `srand()`

e) za pomocą instrukcji `for`  $n$ -razy wylosuj liczbę z podanego przedziału, potem liczby zapisz jako kolejne elementy tablicy