

1. Program zlicza powtarzające się słowa:

/ zadanie bez punktów /

```
int main() {
    int liczba_slow = 0;
    string poprzedni = " ";           // nie słowo
    string biezacy;
    while (cin>>biezacy && biezacy!="koniec") {
        ++liczba_slow;
        if (poprzedni == biezacy)
            cout << "Słowo numer " << liczba_slow
                << " powtarza się: " << biezacy << "\n";
        poprzedni = biezacy;
    }
}
```

- Uzupełnij go o nagłówki bibliotek, instrukcję zatrzymującą okno oraz zwracanie wartości.
 - Prześledź działanie programu na karce papieru dla zdania „*Ten kot kot skoczył.*”
 - Zapisz program w pseudokodzie.
 - Narysuj jego schemat blokowy.
 - Przetestuj działanie programu dla zdania „*Ona ona zaśmiała się he he he ponieważ to co, on zrobił, nie wyglądało dobrze dobrze.*”
- Napisz program sprawdzający, czy dana liczba całkowita jest parzysta. Użyj operatora modulo (%). Zwracaj jasne i kompletne komunikaty, zrozumiałe nie tylko dla Ciebie, np. „*Liczba 4 jest parzysta*” zamiast „*tak*” albo „*nie*”.
 - Napisz program, który prosi użytkownika o podanie dwóch liczb
 - całkowitych
 - zmiennoprzecinkowych.
 Następnie wyświetl kolejno: liczbę większą, mniejszą, sumę, różnicę, iloczyn oraz stosunek dwóch liczb. Porównaj działanie wersji (a) i (b) programu.
 - Napisz program generujący prosty list na podstawie wprowadzonych danych. Niech zaczyna się od „*Witaj <imie>*”. Dodaj jakąś prostą treść. Poproś o podanie płci ('*m*'/'*k*') i zbuduj instrukcję warunkową, która dostosuje formę zdań do płci, np. „*Proszę, byś pamiętał o mnie*” / „*Proszę byś pamiętała o mnie*”. Poproś o podanie wieku, a następnie zmodyfikuj powitanie i zakończenie stosownie do wieku, np. „*Cześć*” do dziecka lub osoby młodej, a „*Dzień dobry*” do dojrzałej, a „*Szanowny Panie*” / „*Szanowna Pani*”, do starszej.

Dla ambitnych

- Napisz program, który prosi użytkownika o podanie trzech liczb całkowitych, a następnie wyświetla je w kolejności od najmniejszej do największej, oddzielone przecinkami. Możesz wykorzystać kod programu porównującego dwie liczby.
 - Zmodyfikuj program z poprzedniego zadania, tak by ustawiał łańcuchy znaków w kolejności alfabetycznej.
- Napisz program przyjmujący operator z dwoma operandami i wyświetlający wynik działania. Np.


```
+ 100 3.14
* 4 5
```

Wczytaj wyrażenie do zmiennej łańcuchowej o nazwie **operacja** i za pomocą instrukcji **if** sprawdź, co użytkownik chce zrobić, np. **if (operacja=="+")**. Operandy wczytaj do zmiennych typu **double**. Zaimplementuj obsługę operatorów **+**, **-**, *****, **/**, *plus*, *minus*, *mnoz*, *dziel* zgodnie z ich konwencjonalnym znaczeniem.