Zestaw 1

1. Napisz program, który narysuje kwadrat o zadanym boku (liczbie gwiazdek) oraz pusty w środku (też o zadanym boku). O parametry zapytaj w programie, sprawdzając ich poprawność. Przykładowy obraz dla zewnętrznego rozmiaru 10 i wewnętrznego 6.

```
* * * * * * * * * *
* * * * * * * * * *
* *           * *
* *           * *
* *           * *
* *           * *
* *           * *
* *           * *
* *           * *
* *           * *
* * * * * * * * * *
* * * * * * * * * *
```

   Uwaga: należy napisać dwie wersje programu. Pierwsza, do rysowania będzie używać manipulatorów `std::setfill` oraz `set::setw` (szczegóły patrz cppreference.com). Druga, do rysowania użyje nowe możliwości formatowania za pomocą `std::format`. Ponieważ druga opcja nie jest jeszcze obsługiwana przez kompilator, polecam program napisać i testować na godbolt.org (edytor w wersji trunk).
2. Napisz program, który zapyta o dwa ciągi znaków oraz je porówna, nie zwracając uwagi na wielkość liter (tzn. żeby traktował literę A i a jako takie same). W tym programie proszę nie używać jeszcze typu `std::string`, tylko ciągi znaków na początku wpisać do odpowiednio pojemnej tablicy `char`.
3. Napisz program, który wczytuje zdanie, a potem sprawdza, czy jest palindromem (tzn. czytane wspak brzmi tak samo). Przykład: *Kobyła ma mały bok*. Należy ignorować białe znaki oraz nie zwracać uwagi na wielkość liter.
4. Napisz funkcję wyliczającą kolejne wyrażenia ciągu Fibonacciego
   a. w wersji rekurencyjnej (czyli funkcja wywołuje samą siebie)
   b. w wersji z jedną pętlą for

   Niech przykładowy program wygląda tak:

```
int main() {
    unsigned long long k = 80;
    for (unsigned long long i=1; i<=k; ++i) {
        cout << fib(i) << endl;
    }
}
```

5. Zmodyfikuj program z poprzedniego zadania tak, żeby korzystał z wyliczonych wcześniej wartości (nie powtarzał ich wyliczania) do wyznaczenia następnych. Na przykład, żeby pytał, który element ciągu ma wyliczyć i jeśli program już wcześniej wyliczył niższe wartości, to miał je zapamiętane i wykorzystał. Można użyć prostą tablicę lub jeśli ktoś potrafi (chce) to jakiś kontener np. `std::vector`.