Lab 10 – Features

Exercise 1. User defined literals

Define user literals for time (ms, s, h) and distance (m, km, cm) they should convert literal to seconds and meters correspondingly.

```
/**
    Computes velocity in meters per seconds.
    @param distance  distance in meters
    @param time      time in seconds
    @return velocity in meters per seconds.
*/
double computeVelocity(double distance, double time){
    return distance/time;
}

int main(){
    cout << computeVelocity(100_m, 5_s) << endl;     //20
    cout << computeVelocity(360_km, 2.0_h) << endl;  //50
    cout << computeVelocity(3.6_km, 0.02_h) << endl; //50
    cout << computeVelocity(250_cm, 2.5_ms) << endl; //1000
    return 0;
}
```

Literature:
https://en.cppreference.com/w/cpp/language/user_literal

Exercise 3. Chrono timer

Implement class **Timer** that measures life span (time from creation to destruction) of its elements. On destruction it should print timer name and life span in seconds.
Provide also method duratioInNanoseconds() which will return life span of an object in nanoseconds (from construction to current moment).

Use: library std::chrono, steady_clock.

Exercise 3. Regex - Mail and links extractor

Write program findLinks that for HTML file (given on standard input) lists all valid emails and non local hyperlinks (starting with http or https inside an a tag).
Use regular expressions from STL: regex, sregex_token_iterator.

Example:

```
curl -s http://ww2.ii.uj.edu.pl/~kapela/test.php | ./findLinks
```

```
Links
BaCa                        : https://baca.ii.uj.edu.pl
Solar System                : https://www.solarsystemscope.com/
USOS UJ                     : http://www.usosweb.uj.edu.pl/
Donald Duck                 : https://en.wikipedia.org/wiki/Donald_Duck

Emails:
 > admin@helpdesk.eu
 > donald@duck.co
 > XX.xx-xx.XX@XXX.xx.pl
```