

1. Zdefiniuj nowe znaczenie dla operatora `*`, tak aby dla dowolnych obiektów `v` oraz `w` klasy `std::vector<double>` wyrażenie `v * w` zwracało iloczyn skalarny `v` i `w`. Przetestuj ten operator. Wskazówka: chodzi o zwykły iloczyn skalarny, czyli pierwszy wzór na stronie [https://en.wikipedia.org/wiki/Dot\\_product](https://en.wikipedia.org/wiki/Dot_product). Możesz założyć, że `v` i `w` mają tę samą liczbę elementów.
2. Zaimplementuj funkcję  

```
double find_root(double (*f)(double), double a, double b,  
                 double tolerance = 1e-10)
```

która za pomocą metody bisekcji ([https://pl.wikipedia.org/wiki/Metoda\\_r%C3%B3wnego\\_podzia%C5%82u](https://pl.wikipedia.org/wiki/Metoda_r%C3%B3wnego_podzia%C5%82u)) znajduje pierwiastek funkcji `f` w przedziale `[a,b]` z dokładnością do `tolerance`. Przetestuj ją, znajdując przy jej pomocy pierwiastki równań  
 $\cos(x) = x$  w przedziale `[0, 2]`  
oraz  
 $\cos(x) = 1/2$  w przedziale `[0, 1.5]`,  
oba z tolerancją nieprzekraczającą  $10^{-6}$ . Tolerancja to maksymalne odchylenie otrzymanej wartości od wartości dokładnej.
3. Napisz program, który będzie wczytywał argumenty wiersza poleceń, a następnie:
  - a. jeżeli wśród nich znajduje się napis `--help` lub `-h`, to wyświetli (dowolny) komunikat o przeznaczeniu programu
  - b. w przeciwnym wypadku będzie wyświetlał sumę swoich argumentów.Przykład:

```
> sumuj.exe 0.5 1 2  
3.5
```