1. **Bisection method.** Make a program that finds the roots of the function

$$f(x) = e^x - 5 \tag{1}$$

with the bisection method in the range $x \in [x_1, x_2]$.

2. **Brent's method.** Calculate the zeros of the function

$$f(x) = \cos(x - \pi/4) + \frac{\sin(x - \pi/4)}{8x} \tag{2}$$

with accuracy $\epsilon = 10^{-6}$ in the range $1 < x < 20$ using the Brent method. Use the Numerical Recipes subroutine `zbrent`.

3. **General iteration formula.** Find the roots of the equation

$$x = \tan(x), \tag{3}$$

where $0 < x < 10$ using the general iteration formula.

4. **Roots of Bessel J_0.** Find the roots of the Bessel J_0 function. Use Numerical Recipes `rtsafe` subroutine, that uses the Newton–Raphson method with bisection method as a safety fallback.

`float zbrent(float (*func)(float), float x1, float x2, float tol)` Finds the root of function `func` using the Brent's method.

- `float func(float x)`, input, function whose roots are searched.
- `float x1`, input, start point of the interval where the root is known to be
- `float x2`, input,, end point of the interval where the root is known to be
- `float tol`, input, tolerance, or the accuracy of the moethod

`void zbrak(float (*fx)(float), float x1, float x2, int n, float xb1[], float xb2[], int *nb)` Subdivides a given interval and returns those subintervals, where function `fx` changes its sign

- `float fx(float x)`, input, function, whose roots are searched.
- `float x1, x2`, input, start and end points of the interval
- `int n`, input, number of subdivisions of the given interval
- `float xb1[nb], xb2[nb]`, output, start and end points of the subintervals where the function changes its sign.
- `int nb`, in/out On input, the maximum number of roots sought, on output, the number actually found.

`float rtsafe(void (*funcd)(float, float *, float *), float x1, float x2, float xacc)` Using a combination of Newton–Raphson and bisection methods, finds the root of the given function

- `void funcd(float x, float *y, float *dy)`, input, subroutine, that calculates the value `y` and derivative `dy` of the function.
- `float x1,x2`, input, start and end points of the interval where the roots are known to be.
- `float xacc`, input, desired accuracy