Four test.

Sin ()

Cos ()

Tan ()

Exp ()

10^{-9}

define EPSILON 0.0000000001

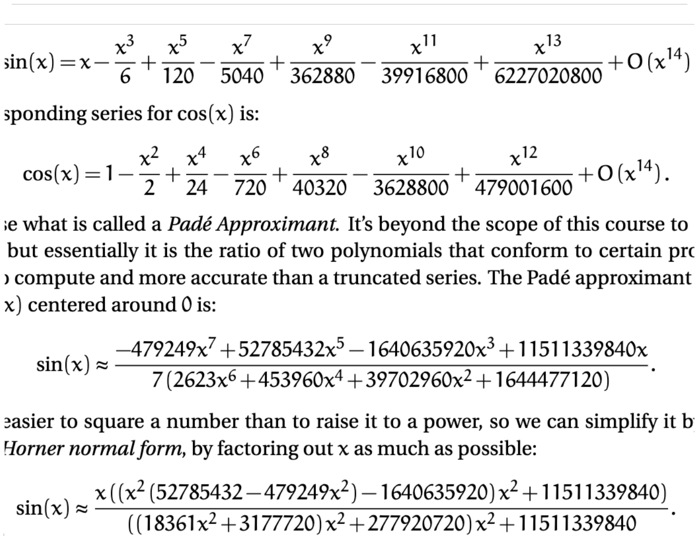
Loop format.

```
while (abs (x - Target) > EPSILON)
        ⋮
```

Save x^2 may save some time.

Restrict domain from $[-2\pi, 2\pi]$ for cos, sin.

Padé approximation for Sin and Cos



$$\sin(x) = x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362880} - \frac{x^{11}}{39916800} + \frac{x^{13}}{6227020800} + O(x^{14})$$

sponding series for $\cos(x)$ is:

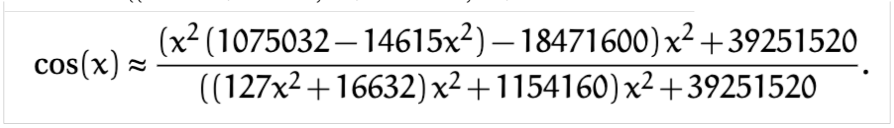
$$\cos(x) = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \frac{x^8}{40320} - \frac{x^{10}}{3628800} + \frac{x^{12}}{479001600} + O(x^{14}).$$

se what is called a *Padé Approximant.* It's beyond the scope of this course to but essentially it is the ratio of two polynomials that conform to certain pro compute and more accurate than a truncated series. The Padé approximant $x)$ centered around 0 is:

$$\sin(x) \approx \frac{-479249x^7 + 52785432x^5 - 1640635920x^3 + 11511339840x}{7(2623x^6 + 453960x^4 + 39702960x^2 + 1644477120)}.$$

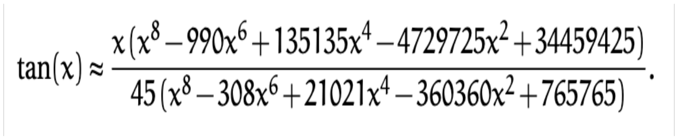
easier to square a number than to raise it to a power, so we can simplify it b *Horner normal form*, by factoring out x as much as possible:

$$\sin(x) \approx \frac{x((x^2(52785432 - 479249x^2) - 1640635920)x^2 + 11511339840)}{((18361x^2 + 3177720)x^2 + 277920720)x^2 + 11511339840}.$$



$$\cos(x) \approx \frac{(x^2(1075032 - 14615x^2) - 18471600)x^2 + 39251520}{((127x^2 + 16632)x^2 + 1154160)x^2 + 39251520}.$$

For tan



$$\tan(x) \approx \frac{x\,(x^8 - 990x^6 + 135135x^4 - 4729725x^2 + 34459425)}{45\,(x^8 - 308x^6 + 21021x^4 - 360360x^2 + 765765)}.$$

undefine when $\cos(x) = 0.$

$$x = \frac{\pi}{2} \cdot n$$

For Exp()

$$\frac{x^n}{n!} = \frac{x^{n-1}}{(n-1)!} \times \frac{x}{n}$$

x = input

n = number keep going.

Pre Lab 1

1

While loop for e^x

```
n = 1
double old = 1
                    //old
while ( abs ( n! - $x^n$ > $\epsilon$ ) {
            //end if the diff
             is < Epilon
old = old × $\frac{++n}{n.}$ }
```

② prinf("%f", old)

Pre Lab 2

① return an integer.

② Bool, because bool only take 1. Byte.

```
3  main() {
    case (command) .{
   if  s
        run   sin  Test
   if  c
        run   cos  Test
   if  t
        run   tan  Test
   if  e
        run   exp  test
if all
    run all test.}
return 0 // exit
  }
```