ASGN 2 ALI SALEHI

TASK

- WRITE 4 FUNCTIONS SIN, COS, TAN, and e^x
  - USE PADÉ APPROXIMATION FOR SIN, COS, TAN AND TAYLOR FOR e^x.
- COMPARE RESULTS TO Math.h LIBRARY
  - CHECK ASGN 2.PDF FOR FORMAT
- SIN & COS $[-2\pi, 2\pi]$, TAN $[-\pi/3, \pi/3]$
- e^x FROM 0-9 WITH STEPS OF 0.1
- BE SURE TO LINK MATH LIBRARY @ COMPILE TIME USING THE -lm FLAG
- NEWTON'S Γ FUNCTION GIVEN

SIN + COS

$[-2\pi, 2\pi)$

- CENTER AROUND 0
- HORNER NORMAL FORM
  - EASIER / FASTER TO COMPUTE
  - FOUND IN LAB MANUAL

PSUEDOCODE:

```
for (x = -2π, x < 2π, x += π/16){
    -NORMALIZE BETWEEN [-π, π]
    - RUN RESPECTIVE HORNER
    - PRINT OUTPUT
}
```

TAN

$[-\pi/3, \pi/3)$

- TAN(x) = SIN(x) / COS(x) / UNDEFINED @ $x = c(\pi/2)$
  - CANNOT DO THIS BC OUR SIN & COS ARE APPROXIMATED

PADÉ FOR TAN:

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

PSUEDOCODE:

```
for (x = -π/3, x < π/3, x += π/16){
    - RUN HORNER EQ.
    - PRINT OUTPUT
}
```

HORNER NORMAL FORM OF TAN:

$$\frac{x(((((x^2-990)x^2+135135)x^2-4729725)x^2+34459425)}{45(((((x^2-308)x^2+21021)x^2-360360)x^2+765765}$$

e^x

- UNRESTRICTED DOMAIN = NO PADÉ POSSIBLE
- NOTICE THAT $\frac{x^n}{n!} = \frac{x^{n-1}}{(n-1)!} \times \frac{x}{n}$
- USE EPSILON ϵ TO HALT COMPUTATION BECAUSE $|x^n| < n!$ FOR A SUFFICIENTLY LARGE n
- $\epsilon = 10^{-9}$ IS SUFFICIENT

PRE LAB PART 1

1) WRITE PSUEDOCODE FOR APPROXIMATING e^x WITH EITHER A FOR LOOP OR WHILE LOOP

initialize variables x (as in e^x), prev, and sum

- prev = previous term / $\frac{x^{n-1}}{(n-1)!}$
- sum will update to most recent term and track final value

```
for (n=1; abs(prev) < EPSILON; ++n) {
    prev = x/n * prev;
    sum += prev;
}
```

2) WRITE PSUEDOCODE FOR PRINTING OUTPUT OF e^x

```
for (i=0; i<=9; i += 0.1) {
  - RUN SEQUENCE FROM ABOVE
  - print ("Formatting", i, sum, exp(i), exp(i) - sum);
}
```

COMMAND LINE ARGUMENTS

- USE getopt() FROM getopt.h
- argc = ARGUMENT COUNTER - DELIMITED BY WHITE SPACE
  - WILL ALWAYS BE 1 MORE THAN # OF ARGUMENTS YOU PASS BC NAME OF EXECUTABLE IS FIRST
- argv = ARGUMENT VALUES — ARRAY OF STRINGS
- getopt() RETURNS -1 WHEN THERE ARE NO MORE ARGUMENTS

PRE-LAB PART 2

1) WHAT DOES getopt() RETURN?

   - getopt() RETURNS AN INT. IT WILL RETURN THE OPTION CHARACTER FOR THE NEXT COMMAND LINE OPTION. IF THERE ARE NO MORE COMMAND LINE OPTIONS IT WILL RETURN -1.

2) IS A bool OR enum THE BEST CHOICE?

   - AN enum WOULD BE PREFERABLE IN THIS SCENARIO BECAUSE THEY PROVIDE A LIST OF POSSIBLE STATES — THAT CAN BE RELATED TO COMMAND LINE OPTIONS — OF WHICH YOU CAN CHOOSE EXACTLY ONE, SATISFYING MUTUAL EXCLUSIVITY.

3) PROVIDE PSUEDO CODE FOR YOUR MAIN FUNCTION

```
int main (int argc, char **argv) {

    op = getopt(...)

    switch (op) {

        's' : sin();
        'c' : Cos();
        't' : Tan();
        'e' : EXP();
        'q' : Sin(); Cos(); Tan(); EXP();

    if (no args) { print("NEED AN ARGUMENT"); }
}
```