Alejandra Briones

Assignment 2 - Design.pdf

newton.c

#define Counter

int sqrt_newton(){ — python (will have to switch)

z = 0.0

y = 1.0

while abs(y-z) > epsilon:

z = y

y = 0.5 * (z + x/z)

return y

}

sqrt_newton_iters(){ * all term/iters counters will follow this

counter += 1

return counter

e.c

Static Counter

ec(){

k = 1.0 $\frac{x^{k-1}}{(k-1)!} \cdot \frac{x}{k}$

prevt = 0.0

while (abs(k-prevt) > Epsilon){ * no loops so pt * k

prevt = (k * prevt)

y = 1/prevt

k += 1

e_terms()

return y //12 / 2k+1 ... 3^k

madhava.c

t=sqrt_newton(12); 2k+1 k=0 = 1 + 2

//finding 0th term $3^0 = 1 \cdot 3$

bt = 1

nt = tt/bt

while (nt-pt) > Epsilon{

pt = nt

temp1 = (2*k)+1;

for (int i=1; i)k; i++){

temp2 = -3*3

} * no for loops

bt = temp1 * temp 2

nt = tt/bt

k += 1

}

return nt;

}

euler.c * the most complete one I have.

```
k=1   * beginning
pt = 0   * previous term
while k> epsilon{
   bottomFact = k*k   * k^2
   nt= 1/bottom fact   * 1/k^2
   x= pt + nt   * x = 0 + 1/k^2
   pt=x   * storing previous term
   k+=1   * incrementing k
  x = x*6   * mult by 6 the summation
  sqrt(x)   * sqrt(6 * sum 1/k^2)
return x
```

bbp.c

```
pi_bbp{
  K=1.0   * Set to first
  pt=0.0   * no previous
  while (fabs(k-0.0) > epsilon{
     tt=(k(120*k+151)+47)   * top term
     bt= (k(k(k(512*k+1024)+712)+194)+15)   * bottom term
     pt= tt/bt   * tt/bt
      x= 1/16   * setting multiple 16^-i
     for (int i = 1; i<k; i++){
          x= 1/16 * x   * 16^-i
     }
     pt= x*pt   16^-i * tt/bt
  }
  return pt
}
```

viete.c

```
K= 1.0
pt=0.0
while (abs(k-pt) > =ps){
     for (int i=0; i>k, i++){
          pt = 2 + pt
          pt= sqrt(pt)
     }
      pt= pt/2
}
 double x=0.0
 x= 2/pt;
 return x
```