Assignment 4.2

Lauren Lee

26th June 2022

Atof() with E Handling

```
double Atof(char s[]){
    double val, power;
    int i, sign;

    for(i = 0; isspace(s[i]); ++i);
    sign = (s[i] == '-')?-1:1;
    if (s[i] == '+' || s[i] == '-'){
        ++i;
    }
    for(val = 0.0; isdigit(s[i]); ++i){
        val = 10*val + (s[i]-'0');
    }

    if(s[i] == '.'){
        ++i;
    }

    for(power = 1.0; isdigit(s[i]); ++i){
        val = 10*val + (s[i] - '0');
        power = power * 10;
    }

    //start of assignment changes
    double coefficient = (sign * val) / power;

    //check for e
    if (s[i] == 'e' || s[i] == 'E'){
        ++i; //if e is present, check the exponent
    } else {
        return coefficient; //if e not present, return the current
    value
    }

    //check sign to see if need to divide or multiple
    int exSign;
    if (s[i] == '-'){
        exSign = -1;
        i++;
    } else if(s[i] == '+'){
        exSign = 1;
        i++;
    } else{
```

```
42          exSign = 0;
43      }
44
45      /*
46      how can I use pointers to take the rest of the s[] for the
        exponent value without using two forloops
47      // char t[counter] = *s[i];
48      //char t[sizeof(*pt)/sizeof(&pt[0])] = *pt;
49      */
50
51      //count the number of vals after e
52      int counter;
53      int position = i;
54      for(counter = 0; s[i]!='\0'; i++ )
55          counter++;
56
57      //hold the exponent val after e in another char[]
58       char t[counter];
59      int k;
60      for(k = 0; counter > 0; k++,counter--, position++){
61          t[k] = s[position];
62      }
63      t[++k] = '\0';
64
65      //convert that char[] to an integer
66      float base = Atof(t);
67
68      //divide or multiple based on sign the number of times the
        exponent val is
69      if(exSign != -1){
70         for(int j = 0; j < base; j++){
71              coefficient = coefficient * 10.0;
72          }
73      } else{
74          for(int j = 0; j < base; j++){
75              coefficient = coefficient / 10.0;
76          }
77      }
78
79      return coefficient;
80  }
81
82  int main(){
83      char in[] = "123.45e6";
84      printf("%f ...should return 123450000\n", Atof(in));
85
86      char in_one[] = "123.45e-6";
87      printf("%f ...should return 0.000123\n", Atof(in_one));
88
89      return 0;
90  }
```

```
123450000.000000 ...should return 123450000
0.000123 ...should return 0.000123
```