Files:

sort.cpp
sort_helper.cpp
sort_helper.h

SORT_HELPER.H:

```
// Absolute value function
double abs(double val);

// Square Root Function
double Sqrt(double v, int initial_guess = 1);

// Power Function
double Power(double b, int e);


// Get Data Function
int GetData(double list[]);

// Sort Function
void Sort(double list[], int count, char order);
```

SORT_HELPER.CPP:

```
#include <iostream>

// Absolute value function
double abs(double val)
{
    if(val < 0)
    {
        return (-val);
    } else
    {
        return val;
    }
```

```
}

// Square Root Function
double Sqrt(double v, int initial_guess = 1)
{
    double new_val = 0.5*(initial_guess + v / initial_guess);
    double error;

    do
    {
        new_val = 0.5*(new_val + v / new_val);
        error = 0.5*(new_val + v/new_val) - new_val;

    } while (abs(error) > 0.005);

    return new_val;

}

// Power Function
double Power(double b, int e)
{
    double final;

    final = b;

    if(e == 0)
    {
        return 1;

    } else if(e > 0)
    {
        for (int i = 0; i < (e-1); i++)
        {
            final *= b;
        }
```

```
    } else if(e < 0)
    {
        for (int i = 0; i < ((-e)-1); i++)
        {
            final *= b;
        }
        final = 1/final;
    }

    return final;

}

// Get Data Function
int GetData(double list[])
{
    int size;
    double input;

    std::cout << "How many values would you like to process?";
    std::cin >> size;

    for(int i = 0; i < size; i++)
    {
        std::cin >> input;
        list[i] = input;
    }

    return size;

}

// Sort Function
void Sort(double list[], int count, char order)
{
```

```
    if(order == 'D' || order == 'd')
    {
        for(int i = count; i > 0; i--)
        {
            for(int j = 0; j < i - 1; j++)
            {
                if(list[j] < list[j+1])
                {
                    std::swap(list[j],list[j+1]);
                }
            }
        }
    } else
    {
        for(int i = count; i > 0; i--)
        {
            for(int j = 0; j < i - 1; j++)
            {
                if(list[j] > list[j+1])
                {
                    std::swap(list[j],list[j+1]);
                }
            }
        }
    }
}
```

SORT.CPP:

```
#include <iostream>
#include "sort_helper.h"

int main()
{

double l0[100];

int size = GetData(l0);
char order;

double l1[size];

for(int i = 0; i < size; i++)
{
    l1[i] = l0[i];

}

std::cout << std::endl <<
             "Would you like to sort in Ascending(A/
a) or Descending(D,d) order?" <<
             std::endl;
std::cin >> order;
Sort(l1, size, order);

std::cout << std::endl << "N" << "\t" << "Square" << "\t" << "Sq
uare Root" << std::endl;

for(int i = 0; i < size; i++)
{
     std::cout << l1[i] <<
                  "\t" <<
```

```
                    Power(l1[i],2) <<
                    "\t" <<
                    Sqrt(l1[i]) <<
                    std::endl;
}

    return 0;
}
```