Lab # 1 -> Recursion
Group: Kevin Williams, Krystal Maughan
CSIS 211

recursivePower
1. Write a recursive function, recursivePower that computes a^n where a is a real number and n is a non-negative integer. This does not have to be a function template. Write pre-and post-conditions if needed

```cpp
//Kevin Williams, Krystal Maughan
//CSIS 211
//Lab #1

//RecursivePower

#include <iostream>
#include <cmath>
#include <vector>
int recursivePower(int, int);
using namespace std;

int main(){
    // recursive power: Write a recursive function recursivePower that computers a^n where a is a real number
    // and n is a non-negative integer. This does not have to be a function template. Make sure and write pre and post
    // conditions if they are needed

    // Pre condition -> a is a real number but can be negative
    // n is a non-negative number (ie 0 and above)
    // Post-condition -> returns an int that is the power a ^ n for the values entered
    int a;
    int n;
    int ans;
    cout << "Please enter a number, a, which is real" << endl;
    cin >> a;
    cout << "Please enter a number, n, which is non-negative (ie 0 and above) " << endl;
    cin >> n;
    cout << recursivePower(a, n) << endl;
    system("PAUSE");
    return 0;
}
int recursivePower(int x, int y)
{
    int ans;
    if (y == 0){
        return 1;
    }
    else if (y == 1)
    {
        return x;
    }
    else{
        return x * recursivePower(x, y - 1);

    }
}
```

```
c:\users\kmaughan\documents\visual studio 2013\Projects\Lab1_\Debug\Lab1_.exe
Please enter a number, a, which is real
5
Please enter a number, n, which is non-negative (ie 0 and above)
3
125
Press any key to continue . . .
```

```cpp
#include <iostream>
#include <cmath>
#include <vector>

int recursivePower(int, int);

using namespace std;

int main(){
    // recursive power: Write a recursive function
    // recursivePower that computers a^n where a is a real number
    // and n is a non-negative integer. This does not have to be a
    // function template. Make sure and write pre and post
    // conditions if they are needed

    // Pre condition -> a is a real number but can be negative
    // n is a non-negative number (ie 0 and above)
    // Post-condition -> returns an int that is the power a ^ n for the values entered
    int a;
    int n;
    int ans;
    cout << "Please enter a number, a, which is real" << endl;
    cin >> a;
    cout << "Please enter a number, n, which is non-negative (ie 0 and above) " << endl;
    cin >> n;
    cout << recursivePower(a, n) << endl;
    system("PAUSE");
    return 0;
}
int recursivePower(int x, int y)
{
    int ans;


    if (y == 0){
        return 1;
    }
    else if (y == 1)
    {
        return x;
    }
    else{
        return x * recursivePower(x, y - 1);

    }
}
```

```
c:\users\kmaughan\documents\visual studio 2013\Projects\Lab1_\Debug\Lab1_.exe
Please enter a number, a, which is real
7
Please enter a number, n, which is non-negative (ie 0 and above)
3
343
Press any key to continue . . .
```

```cpp
//Kevin Williams, Krystal Maughan
//CSIS 211
//Lab #1

//RecursivePower

#include <iostream>
#include <cmath>
#include <vector>
int recursivePower(int, int);
using namespace std;

int main(){
    // recursive power: Write a recursive function recursivePower that computers a^n where a is a real number
    // and n is a non-negative integer. This does not have to be a function template. Make sure and write pre and post
    // conditions if they are needed

    // Pre condition -> a is a real number but can be negative
    // n is a non-negative number (ie 0 and above)
    // Post-condition -> returns an int that is the power a ^ n for the values entered
    int a;
    int n;
    int ans;
    cout << "Please enter a number, a, which is real" << endl;
    cin >> a;
    cout << "Please enter a number, n, which is non-negative (ie 0 and above) " << endl;
    cin >> n;
    cout << recursivePower(a, n) << endl;
    system("PAUSE");
    return 0;
}
int recursivePower(int x, int y)
{
    int ans;
    if (y == 0){
        return 1;
    }
    else if (y == 1)
    {
        return x;
    }
    else{
        return x * recursivePower(x, y - 1);

    }
}
```

```
c:\users\kmaughan\documents\visual studio 2013\Projects\Lab1_\Debug\Lab1_.exe
Please enter a number, a, which is real
3
Please enter a number, n, which is non-negative (ie 0 and above)
3
27
Press any key to continue . . .
```

Lab # 1 -> Recursion
Group: Kevin Williams, Krystal Maughan
CSIS 211

maxArray
2. Implement the Recursive maxArray function. It must be a function template. This should find maximum of int arrays, double arrays, char arrays.

```cpp
//Kevin Williams, Krystal Maughan
//CSIS 211
//Lab #1
//i hope you dont find this program as annoying as i did...
//Pre-conditions -> input can be int, double, char
//Post -> output must return maximum value of each array
#include <iostream>
#include <string>
using namespace std;
template<typename Hex>
Hex maximum(Hex arr[], int SIZE)
{
    Hex maximum = arr[0];
    for (int k = 0; k < SIZE; k++)
    {
        if (arr[k] > maximum)
        {
            maximum = arr[k];
        }
    }
    return maximum;
}
int main()
{
    const int SIZE = 5;
    int arr[SIZE];
    char arr2[SIZE];
    double arr3[SIZE];

    cout << "Input 5 ints." << endl;
    for (int i = 0; i < SIZE; i++)
    {
        cin >> arr[i];
    }
    cout << "Input 5 chars." << endl;
    for (int i = 0; i < SIZE; i++)
    {
        cin >> arr2[i];
    }
    cout << "Input 5 doubles." << endl;
    for (int i = 0; i < SIZE; i++)
    {
        cin >> arr3[i];
    }
    cout << "The maximum number in arr is: " << maximum(arr, SIZE) << endl;
    cout << "The maximum char(ascii) in arr2 is: " << maximum(arr2, SIZE) << endl;
    cout << "The maximum double in arr3 is: " << maximum(arr3, SIZE) << endl;
    system("PAUSE");
    return 0;
}
```

```
c:\users\kmaughan\documents\visual studio 2013\Projects\Lab1_\Debug\Lab1_.exe
Input 5 ints.
1
2
3
4
5
Input 5 chars.
a
b
c
d
e
Input 5 doubles.
1.0
2.0
3.0
4.0
5.0
The maximum number in arr is: 5
The maximum char(ascii) in arr2 is: e
The maximum double in arr3 is: 5
Press any key to continue . . .
```

Lab # 1 -> Recursion
Group: Kevin Williams, Krystal Maughan
CSIS 211

binarySearch
3. Rewrite the binary search algorithm as a template. It should search int arrays, double arrays and char arrays.

1. Asks the user to pick five ints, chars, doubles
2. Asks user to find an element (int, char, double respectively).
3. Returns index of element if found (or -1 otherwise)

```cpp
//Kevin Williams, Krystal Maughan
//CSIS 211
//Lab #1
//Pre-conditions -> input can be int, double, char based on array type (ie of same type as array)
//Post -> output must return index of input or else -1
#include <iostream>
#include <cmath>
#include <vector>
#include <string>
using namespace std;
template<typename T>
int binarySearch(const T list[], T key, int arraySize)
{
    int low = 0;
    int high = arraySize - 1;
    while (high >= low)
    {
        int mid = (low + high) / 2;
        if (key < list[mid])
            high = mid - 1;
        else if (key == list[mid])
            return mid;
        else
            low = mid + 1;
    }
    return -1;
}
int main()
{
    const int SIZE = 5;
    int arr[SIZE];
    char arr2[SIZE];
    double double3[SIZE];
    int key1;
    double key2;
    char key3;
    cout << "Please pick five integers. " << endl;
    for (int i = 0; i < SIZE; i++){
        cin >> arr[i];
    }
    cout << "Enter the integer you would like to find" << endl;
    cin >> key1;
    cout << "Please pick five chars. " << endl;
    for (int i = 0; i < SIZE; i++){
        cin >> arr2[i];
    }
    cout << "Enter the char you would like to find " << endl;
    cin >> key3;
    cout << "Please pick five doubles. " << endl;
    for (int i = 0; i < SIZE; i++){
        cin >> double3[i];
    }
    cout << "Enter the double you would like to find" << endl;
```

```
c:\users\kmaughan\documents\visual studio 2013\Projects\Lab1_\Debug\Lab1_.exe
Please pick five integers.
1
2
3
4
5
Enter the integer you would like to find
3
Please pick five chars.
a
b
c
d
e
Enter the char you would like to find
b
Please pick five doubles.
1.0
2.0
3.0
4.0
5.0
Enter the double you would like to find
2.0
2
1
1
Press any key to continue . . .
```

```cpp
    cin >> key2;
    cout << binarySearch(arr, key1, SIZE) << endl;
    cout << binarySearch(arr2, key3, SIZE) << endl;
    cout << binarySearch(double3, key2, SIZE) << endl;
    system("PAUSE");
    return 0;
}
```

Lab # 1 -> Recursion
Group: Kevin Williams, Krystal Maughan
CSIS 211

reverseDigits
4. Write a recursive function that takes an integer as an argument and returns an integer with the digits reversed. The function is required to take an integer as an argument. You do not need and should not have a template for this one.

Asks the user to input a number
Returns reverse of number

```cpp
//Kevin Williams, Krystal Maughan
//CSIS 211
//Lab #1
//Pre-conditions -> input must be a string type, but a number input
//Post -> prints reverse type as string
#include <iostream>
#include <string>
using namespace std;

void printReverse(string, int);

int main()
{
    string num;

    cout << "Please type in a number with multiple digits." << endl;
    cin >> num;

    printReverse(num, num.length());

    //cout << "Your number was " << num << " and the reverse is " << reverse << endl;

    system("PAUSE");
    return 0;
}

void printReverse(string n, int length)
{
    int len = length - 1;
    if (len == 0)
    {
        cout << n[len] << endl;
    }
    else
    {
        cout << n[len];
        printReverse(n, len);
    }
}
```

```
c:\users\kmaughan\documents\visual studio 2013\Projects\Lab1_\Debug\Lab1_.exe
Please type in a number with multiple digits.
12345667834343
34343876654321
Press any key to continue . . .
```

```cpp
//Kevin Williams, Krystal Maughan
//CSIS 211
//Lab #1
//Pre-conditions -> input must be a string type, but a number input
//Post -> prints reverse

#include <iostream>
#include <string>
using namespace std;

void printReverse(string, int);

int main()
{
    string num;

    cout << "Please type in a number with multiple digits." << endl;
    cin >> num;

    printReverse(num, num.length());

    //cout << "Your number was " << num << " and the reverse is " << reverse << endl;

    system("PAUSE");
    return 0;
}

void printReverse(string n, int length)
{
    int len = length - 1;
    if (len == 0)
    {
        cout << n[len] << endl;
    }
    else
    {
        cout << n[len];
        printReverse(n, len);
    }
}
```

```
c:\users\kmaughan\documents\visual studio 2013\Projects\Lab1_\Debug\Lab1_.exe
Please type in a number with multiple digits.
232325343423
324343523232
Press any key to continue . . .
```

```cpp
//Kevin Williams, Krystal Maughan
//CSIS 211
//Lab #1
//Pre-conditions -> input must be a string type, but a number input
//Post -> prints reverse

#include <iostream>
#include <string>
using namespace std;

void printReverse(string, int);

int main()
{
    string num;

    cout << "Please type in a number with multiple digits." << endl;
    cin >> num;

    printReverse(num, num.length());

    //cout << "Your number was " << num << " and the reverse is " << reverse << endl;

    system("PAUSE");
    return 0;
}

void printReverse(string n, int length)
{
    int len = length - 1;
    if (len == 0)
    {
        cout << n[len] << endl;
    }
    else
    {
        cout << n[len];
        printReverse(n, len);
    }
}
```

```
c:\users\kmaughan\documents\visual studio 2013\Projects\Lab1_\Debug\Lab1_.exe
Please type in a number with multiple digits.
3433454443431239
9321343444543343
Press any key to continue . . .
```