

Understanding

1. I need to create a program called randFun
 1. It needs to use pass by reference in order to calculate a random number
2. I need to create a program called refFun
 1. I needs to take two strings, pass them by reference to a function, and return whether they are the same or different.
3. I need to create a program called sortFun
 1. It needs to take three ints and sort them
 1. once in order a, b, c
 2. once in order c, a , b
4. I need to create a program called recFun
 1. It needs to calculate appropriate fibonacci numbers
5. I need to create a program called hailstone
 1. It needs to see how many recursions are needed to calculate the hailstone sequence for a given number.
6. I need to create a program called reconvert
 1. It needs to convert decimal to binary when asked.
 2. It needs to convert binary to decimal when asked.
 3. It needs to exit when asked.
7. I need to create this reflection.
8. I need to discuss.

Design

hailstone.cpp

using and includes statements

```
void hailstone(int n, int &timesRun)
{
```

```
    if(n == 1)
    {
        cout << n << endl;
        //break out of recursion
    }
    else
    {
        if((n % 2) == 0)
        {
            cout << n << endl;
            hailstone((n / 2), timesRun);
            timesRun++;
        }
    }
}
```

```

        else
        {
            cout << n << endl;
            hailstone(((n * 3) + 1), timesRun);
            timesRun++;
        }
    }
}

int main()
{

    int initialNum;
    int numRun = 0;

    cout << "Enter an initial hailstone integer: ";
    cin >> initialNum;

    hailstone(initialNum, numRun);

    cout << "Ran " << numRun << " times.";

}

```

randfun.cpp

using and include statements

```

void rand_int(const int &min, const int &max, int &val)
{

    val = rand() % ((max - min) + 1) + min;
}

int main()
{
    //seed the random number generator
    srand(time(NULL));

    int lowerBound;
    int upperBound;
    int result;

    cout << "Please enter two integers." << endl;
    do {
        do {

```

```

        if (!cin) {
            cin.clear();
            cin.ignore(INT_MAX, '\n');
        }

        cout << "Number 1: ";
        cin >> lowerBound;
        cin.ignore(INT_MAX, '\n');
    }
    while (!cin);

    do {

        if (!cin) {
            cin.clear();
            cin.ignore(INT_MAX, '\n');
        }
        cout << "number 2: ";
        cin >> upperBound;
        cin.ignore(INT_MAX, '\n');
    }
    while (!cin);
}
while(lowerBound >= upperBound);

rand_int(lowerBound, upperBound, result);

cout << result << endl;

return 0;
}

```

recFun.cpp

using and includes

```

int fib(int n)
{
    if(n > 1)
    {
        return fib(n - 1) + fib(n - 2);
    }
    else if(n == 1)
    {
        return 1;
    }
    else if(n == 0);
}

```

```

    {
        return 0;
    }
}

int fibEfficient(int n, int* sequence)
{
    if(sequence[n-1] == 0 && sequence[n-2] == 0)
    {
        fibEfficient(n-1, sequence);
        sequence[n-1] = sequence[n-2] + sequence[n-3];
    }
    else
    {
        sequence[n-1] = sequence[n-2] + sequence [n-3];
    }

    return sequence[n-1];
}

int main()
{
    int fibNumber;

    //over 46 causes overflow
    cout << "Please enter the number of the sequence that you want (1-46 only): ";
    cin >> fibNumber;

    cout << "Starting inefficient calculation... (you will see the difference as n gets higher) " <<
endl;
    cout << fib(fibNumber) << endl;

    int sequence[fibNumber];
    for(int x = 0; x < fibNumber; x++)
    {
        sequence[x] = 0;
    }

    sequence[0] = 1;
    sequence[1] = 1;

    cout << "Starting efficient calculation... " << endl;
    cout << fibEfficient(fibNumber, sequence) << endl;

    return 0;
}

```

refFun.cpp

using and includes

```
bool StrComp(string &string1, string &string2)
{

    if(string1.length() != string2.length())
    {

        return false;
    }
    else
    {
        for(int i = 0; i < string1.length(); i++)
        {
            if(string1.at(i) == string2.at(i))
            {

                continue;
            }
            else
            {
                return false;
            }
        }
    }

    return true;
}

int main()
{
    string string1;
    string string2;

    cout << "Enter two strings." << endl;
    cout << "String 1: ";
    getline(cin, string1);
    cout << "String 2: ";
    getline(cin, string2);

    bool same = StrComp(string1, string2);
```

```

    if(same)
    {
        cout << "The strings are the same." << endl;
    }
    else
    {
        cout << "The strings are different." << endl;
    }

    return 0;
}

```

sortFun.cpp

```

void sort3(int &val1, int &val2, int &val3)
{
    do
    {
        if(val1 > val2)
        {
            int tmpVar;

            tmpVar = val2;
            val2 = val1;
            val1 = tmpVar;
        }
        if(val2 > val3)
        {
            int tmpVar;
            tmpVar = val3;
            val3 = val2;
            val2 = tmpVar;
        }
    }
    while(val1 > val2 || val2 > val3 || val1 > val3);
}

int main()
{
    int a;
    int b;
    int c;

    cout << "Please enter int a: ";
    cin >> a;
    cout << "Please enter int b: ";
    cin >> b;
    cout << "Please enter int c: ";
}

```

```

cin >> c;

sort3(a, b, c);

cout << "Sorted values: " << a << " " << b << " " << c << endl;

sort3(c, a, b);

cout << "Sorted values (c, a, b): " << c << " " << a << " " << b << endl;

}

```

recConvert.cpp

```

using namespace std;

int convertBinaryToDecimal(string binary)
{
    int decimal = 0;

    if(binary.at(0) == '0')
    {
        if(binary.length() == 1)
        {
            return 0;
        }
        else
        {
            binary.erase(binary.begin());
            return convertBinaryToDecimal(binary);
        }
    }
    else
    {
        if(binary.length() == 1)
        {
            return 1;
        }
        else
        {
            int len = binary.length();
            binary.erase(binary.begin());
            decimal = (pow(2, (len - 1))) + convertBinaryToDecimal(binary);

            return decimal;
        }
    }
}

```

```
}  
}
```

```
void convertDecimalToBinary(int decimal, string &appendString)  
{
```

```
    if ((decimal / 2) != 0)  
    {  
        convertDecimalToBinary(decimal / 2, appendString);  
    }
```

```
    char digit = '0' + (decimal % 2);  
    string tmpString(1, digit);  
    appendString.append(tmpString);  
}
```

```
bool validateBinaryNumber(string validate)
```

```
{  
    for(int x = 0; x < validate.length(); x++)  
    {  
        if(validate.at(x) == '0' || validate.at(x) == '1')  
        {  
            continue;  
        }  
        else  
        {  
            return false;  
        }  
    }  
}
```

```
    return true;  
}
```

```
bool validatePositiveDecimalNumber(string validate)
```

```
{  
    for(int x = 0; x < validate.length(); x++)  
    {  
        if(isalpha(validate.at(x)))  
        {  
            return false;  
        }  
    }  
}
```

```
    if(atoi(validate.c_str()) < 0)  
    {  
        return false;  
    }
```



```

    }
    else
    {
        return true;
    }
}

int main()
{
    int choice;
    string userInput;
    string appendString = "";

    cout << "Welcome to number converter!" << endl;

    do
    {
        do
        {
            cout << "Please choose from the following menu options:" << endl;
            cout << "1) Convert binary number to decimal." << endl;
            cout << "2) Convert decimal number to binary." << endl;
            cout << "3) Exit." << endl;

            do
            {
                if (!cin)
                {
                    cin.clear();
                    cin.ignore(INT_MAX, '\n');
                }

                cout << "Choice: ";
                cin >> choice;
                cin.ignore(INT_MAX, '\n');
            }
            while (!cin);

            if (choice > 3 || choice < 1)
            {
                cout << "Invalid selection!" << endl << endl;
            }
        }
        while (choice > 3 || choice < 1);

        switch (choice)
        {
            case 1:

```

```

do
{
    cout << "Please enter a binary number to convert: ";
    cin >> userInput;
    if (!validateBinaryNumber(userInput))
    {
        cout << "Invalid binary number! Please try again." << endl << endl;
    }
}
while (!validateBinaryNumber(userInput));

cout << convertBinaryToDecimal(userInput) << endl << endl;
break;
case 2:
do
{
    cout << "Please enter a decimal number to convert (int): ";
    cin >> userInput;
    if (!validatePositiveDecimalNumber(userInput))
    {
        cout << "Invalid positive decimal number! Please try again." << endl << endl;
    }
}
while (!validatePositiveDecimalNumber(userInput));

convertDecimalToBinary(atoi(userInput.c_str()), appendString);
cout << appendString << endl << endl;
break;
case 3:
    exit(0);
    // no break statement needed
default:
    cout << "This is never reached.";
    //this is never reached.
}

}

while(true); //the user will always exit using the menu system.

return 0; // not really needed, but put it here for stylistic reasons
}

```

Testing

Hailstone.cpp

Input	Expected Output	Output
3	7	7
20	7	7
2	1	1

randfun.cpp

Input	Expected Output	Output
1, 10	number between 1-10	same
1, 20	number between 1-20	same
1, 40	number between 1-40	same

recFun.cpp

Input	Expected Output	Output
20	6765	6765
40	102334155	102334155
46	1836311903	1836311903

refFun.cpp

Input	Expected Output	Output
helo, hello	string different	strings different
hello, hello	strings same	strings same
hello, abcde	strings different	strings different

sortFun.cpp

Input	Expected Output	Output
1, 2,3	1,2,3	1,2,3
1, 30, 20	1, 20, 30	1, 20 , 30
12, 1 , 4	1, 4, 12	1, 4, 12

recConvert.cpp

Input	Expected Output	Output
d -> b	85	1010101
b -> d	1010101	85
3 (exit)	exit	exit

Reflect

1. What did you learn?

1. Nothing was really new this time around for me. I learned some nice tricks for conversion in c++ (int to char) from stack overflow, which was neat. I was able to reuse a lot of code as well, which was nice.

2. Was your understanding complete at the start of the project.

1. I believe my understanding was complete at the start of the project.

3. Was your initial design adequate

1. Besides minor tweaks, yes it was.

4. Did your tests work as expected

1. Yes

5. Did implementation go without any problems

1. I design in psuedo-code that is quite close to c++. This make implementation easy. My logic all seemed to translate really well to the actual projects. I did need help in converting an int to a char, which I got from SO.

6. What techniques have we covered this week and in past weeks that helped you approach the problem, did you need to use outside sources to help solve the problem (list sites, books, or other materials that were helpful)? Does this project seem related to previous projects and do you see any names for future projects that it might be related to?

1. I was able to reuse code I wrote from last week. Otherwise, I didn't use any other used one stack overflow source.