Gavin McRoy
Lab: C1011-100-005 Lecture: CPSC 1010-100
Lab 09 – Part 2
10/28/2019
Valary Olon-CPSC 1010-100 , Elana Welch-CPSC 1010-200

```
const MAX_SIZE = 20;

typedef struct romanNumeral_struct {
        Declare romanNumerals of characters[20]
        Declare integer romanInDecimal
} romanNumeral;

Function bool Prompt(char romanNumeral[], int maxSize);

Function bool ProperRoman(const char romanNumeral[]);

Function romanNumeral ConvertRoman(romanNumeral romanNumeralToConvert);

Function void PrintRoman(romanNumeral romanNumeralToConvert);

 Function int main(void)
        romanNumeral myRoman;

        While (Prompt(myRoman.numeral, MAX_SIZE))
                myRoman = ConvertRoman(myRoman);
                PrintRoman(myRoman);
        Write("Thank you for using the Roman Numeral to Decimal calculator.\nGoodbye!\n");


/*
  This function prompts the user for a Roman Numeral
  and calls ProperRoman to validate the input until
  a valid Roman Numeral is entered or the user types EXIT
  to end the program.

  When the Roman Numeral is valid, the function returns true.

  When EXIT is entered, the function returns false.
*/

bool Prompt(char romanNumeral[], int maxSize)
        Pass in: character array of roman numerals , an integer specifying the size of
```

```
        character array
        Do
                Write("Enter a roman numeral digit")
                Read(romanNumeral)
                Bool isProperRoman = ProperRoman(romaNumeral)

        while(!isProperRoman || romanNumeral equals "EXIT")

        Pass out: return true if legal roman numerals digits entered , false if exit is entered


/*
  This function takes a Roman Numeral and validates that only
  legal digits have been entered.
  MDCLXVI

  The function returns true when a valid Roman Numeral is entered, otherwise
  it returns false.
*/
bool ProperRoman(const char romanNumeral[])
        Pass in: character array of roman numerals
        Set array eachRomanNumerals to {'M','D','C','L','X','V','I'}
        Int sumFound = 0
        for(i = 0; i< length of romanNumeral; i++)
                for(j = 0; j<length of eachRomanNumerals; j++)
                        if(romanNumerals[i]==eachRomanNumerals[j])
                                Sum++
        Pass out: true if sum equals length of romanNumerals else return false

/*
  This Function takes a valid Roman Numeral struct, calculates the integer
  equivalent, and returns the updated struct.
*/
romanNumeral ConvertRoman(romanNumeral romanNumeralToConvert);
  MDCLXVI
        Set previousVal to 0
        Set currentVal to 0
        Set sum = 0;
        for(i = 0; i< length of romanNumeralToConvert.romanNumerals; i++)
                if(romanNumeralToConvert.romanNumerals[i] equals 'M')
                        sum+=1000;
                        currentVal = 1000;
                Else if(romanNumeralToConvert.romanNumerals[i] equals 'D')
```

```
                sum+=500;
                currentVal = 500;
        Else if(romanNumeralToConvert.romanNumerals[i] equals ‘C’)
                sum+=100;
                currentVal = 100;
        Else if(romanNumeralToConvert.romanNumerals[i] equals ‘L’)
                sum+=50;
                currentVal = 50;


        Else if(romanNumeralToConvert.romanNumerals[i] equals ‘X’)
                sum+=10;
                currentVal = 10;
        Else if(romanNumeralToConvert.romanNumerals[i] equals ‘V’)
                sum+=5;
                currentVal = 5;
        Else
                sum+=1;
                currentVal = 1;
        if(previousVal>currentVal)
                sum-=previousVal;

        previousVal = currentVal

/*
  This function takes a valid Roman Numeral struct and prints out the roman number
  and integer value nicely formatted as:
  XX = 20
*/
void PrintRoman(romanNumeral romanNumeralToConvert);
        Pass in: structure romanNumeral
        Write(romanNumeralToConvert.romanNumerals“=” +
        romanNumeralToConvert.romanInDecimal)
        Pass out: Void
```