CSC2402  
Assignment 1  
Author: Cameron Duncan-Kemp  
u1032841

# Question 1:

**Source Code:**

#include <iostream>

#include <cmath>

#include <fstream>

#include <string>

#include <stdlib.h>

#include <sstream>

using namespace std;

bool numberPalindrome(string, int& sum); // Function prototype

int sum = 0; //Initialisation of sum

int main()

{

string number;

int x;

int temp;

ifstream inFile; //Open filestream

inFile.open("numbers.txt");

if (!inFile) { //Test if file opened correctly

cout << "Unable to open the requested file";

exit(1);

}

while(inFile.good()){ //Run while file still has data left

getline (inFile, number);

if(numberPalindrome(number, sum)){

cout << "Number: " << number << " is a palindromic number." << endl;

}

}

cout << "A total of " << sum << " palindromic numbers were found.";

return 0;

}

bool numberPalindrome(string number, int& sum){ //Function to test if palindrome

int length = number.length();

for (int i = 0; i < length / 2; i++)

if (number[i] != number[length - 1 - i])

return false;

sum++;

return true;

}

**Output:**

Number: 1011111101 is a palindromic number.

Number: 112121211 is a palindromic number.

Number: 4436666344 is a palindromic number.

Number: 2331991332 is a palindromic number.

Number: 1744114471 is a palindromic number.

Number: 1838558381 is a palindromic number.

Number: 293191392 is a palindromic number.

Number: 2089009802 is a palindromic number.

A total of 8 palindromic numbers were found.

# Question 2:

**Source Code:**

#include <iostream>

#include <cmath>

#include <fstream>

#include <string>

#include <stdlib.h>

#include <sstream>

#include <iomanip>

using namespace std;

// Declaration of functions and variables

bool isPrime(int);

bool isArmstrong(int);

int x, i,primeTotal, digitSum, temp, remain, digit;

bool flag;

string lineBreak = "+----------+----------+----------+\n"; //Set line divider as string for easy insertion

int main(){

ofstream myFile;

myFile.open("output.txt"); //Start outstream to file

int primeTotal = 0; //Initialisation of totals

int armTotal= 0;

int doubleTotal= 0;

// Printing of headers

myFile << left

<< lineBreak

<< setw(11) << "|Number"

<< setw(11) << "|Prime"

<< setw(11) << "|Armstrong |"

<< endl;

for(int i = 100; i <= 500; i++) //Initialise for-loop to run from 100-500 inclusive

{

if(isPrime(i)){ //Print if number is prime and add to prime total count

myFile

<< lineBreak << "|"

<< setw(10) << i << "|"

<< setw(10) << "X" << "|"

<< setw(10) << " " << "|"

<< endl;

primeTotal++;

}

if (isArmstrong(i)) { //Print if number is Armstrong and add to Armstrong total count

myFile << lineBreak << "|"

<< setw(10) << i << "|"

<< setw(10) << " " << "|"

<< setw(10) << "X" << "|"

<< endl;

armTotal++;

}

if(isArmstrong(i) && isPrime(i)){ //Add to a dual total if a number is both armstrong and prime.

doubleTotal++;

}

}

// Final total outputs

myFile << lineBreak << endl;

myFile << "Between 100 and 500 range a total of " << primeTotal << " prime numbers were found." << endl;

myFile << "Between 100 and 500 range a total of " << armTotal << " Armstrong numbers were found." << endl;

myFile << "Between 100 and 500 range a total of " << doubleTotal << " numbers were found that meet both the prime number and Armstrong number conditions." << endl;

return 0;

}

bool isPrime(int x){ //Function to test if the number is prime

flag= true;

for(i = 2; i <= x / 2; ++i){

if(x % i == 0)

{

flag= false;

break;

}

}

return flag;

}

bool isArmstrong(int i){ //Function to test if the number is Armstrong

temp = i;

digit = 0;

while (temp != 0) {

digit++;

temp = temp/10;

}

temp = i;

digitSum = 0;

while (temp != 0) {

remain = temp%10;

digitSum = digitSum + pow(remain, digit);

temp = temp/10;

}

if(digitSum == i){

return true;

}

else

{

return false;

}

}

**Output:**

See output.txt file

# Question 3:

**Source code:**

#include <iostream>

#include <cmath>

#include <tuple>

using namespace std;

//Initiliase variables

int a, b, t, c;

int sum= 0;

int spare= 0;

struct foo //Create structure for function return data.

{

int sum;

int spare;

};

foo calc(int, int, int) //Calculation function

{

c= a + b;

if(t>=a) //for first run through

{

t= t-a;

sum++;

}

while(t>=c) //for each subsequent run through

{

t -= c;

c += b;

sum++;

}

spare += t;

foo result = {sum, spare}; //return both to foo structure

return result;

}

int main(){

// I/O statements

cout << "Enter the number of minutes to prepare first dish: ";

cin >> a;

cout << "Enter how many more minutes to prepare the next dish: ";

cin >> b;

cout << "Enter the total time (in minutes) to prepare all the dishes: ";

cin >> t;

calc(a, b, t); //Function call

cout << "In " << t << " minute(s) the pastry cook can prepare " << sum << " dishes with " << spare << " minutes to spare.";

return 0;

}

**Output:**

Enter the number of minutes to prepare first dish: 6

Enter how many more minutes to prepare the next dish: 4

Enter the total time (in minutes) to prepare all the dishes: 57

In 9 minute(s) the pastry cook can prepare 4 dishes with 9 minutes to spare.

# Question 4:

**Source code:**

#include <iostream>

#include <iomanip>

#include <cmath>

#include <tuple>

using namespace std;

struct foo2 //Create structure for function return data.

{

float firstYear;

float secondYear;

};

struct foo1 //Create structure for function return data.

{

float a;

float b;

float c;

};

float calculate(float&, float&, float&, float&, float&); //Function declarations.

void display(float&, float&);

float input(float&, float&, float&);

int main(){

float firstYear,secondYear;

float cur, one, two;

input(cur, one, two);

calculate(cur, one, two,firstYear, secondYear);

display(firstYear,secondYear);

return 0;

}

float input(float& cur, float& one, float& two) //Receive user input

{

cout << "Enter the current price of the item: ";

cin >> cur;

cout << "Enter the price of the item one year ago: ";

cin >> one;

cout << "Enter the price of the item two years ago: ";

cin >> two;

void display(float calculate(float& cur,float& one, float& two));

return 0;

};

float calculate(float& cur, float& one, float& two,float& firstYear, float& secondYear) //Calculate inflation rates

{

firstYear= (cur - one)/one;

secondYear = (one - two)/two;

void display(float& firstYear, float& secondYear);

return 0;

}

void display(float& firstYear, float& secondYear) //Function to display the results

{

cout << "Current year inflation: " << setprecision(1) << firstYear << endl;

cout << "Inflation one year ago: " << setprecision(1) << secondYear << endl;

if (firstYear > secondYear) {

cout << "Inflation is increasing.";

};

if (firstYear < secondYear){

cout << "Inflation is decreasing.";

}

}

**Output:**

Enter the current price of the item: 33.95

Enter the price of the item one year ago: 36.99

Enter the price of the item two years ago: 40.99

Current year inflation: -0.08

Inflation one year ago: -0.1

Inflation is increasing.