**Question 1**

#include <stdio.h>

#include <stdbool.h>

/\*Vivian Lam, CS2211 Assignment3: Program 1\*/

/\* Name: Program1

Purpose: Asks user to input the time of day (24hour time) and time

duration and add the two together

\*/

int main (void)

{

/\*Declares and initializes variables.

First integer represents time of day

second integer represents time duration.

Third variable is the sum of the two integers.

digitTens is used to extract the digit in the tens column of

timeDur and timeDay

checkCorrect is a boolean used to check if the digit in the tens

column of timeDay and timeDur are valid\*/

int timeDay = 0;

int timeDur;

int result;

int digitTens;

bool checkCorrect = false;

/\*NOTE:

first integer must be less than 2400 and the minutes must be less than 59

(tens digit is <6)

second integer minutes must be less than 59 (tens digit < 6)

\*/

/\*if the above conditions are not met then loop and keep prompting

user to enter a correct value\*/

//loop for first value. also makes sure that time of day is positive

while((timeDay > 2400) || (timeDay < 0) || (checkCorrect == false))

{

/\*resets the boolean checking variable to be false\*/

checkCorrect = false;

/\*Asks the user to enter the first integer\*/

printf("Please enter the first integer (time of day on a 24 hour clock): ");

/\*reads the integer and stores it into timeDay\*/

scanf("%d", &timeDay);

/\*extracts the digit from the tens column from input and checks if it's under 6\*/

digitTens = (timeDay / 10) %10;

if (digitTens <=5)

{

checkCorrect = true;

}

}

checkCorrect = false; //resets the value of the boolean variable to be false

//loop for second variable

while(checkCorrect == false)

{

/\*resets the boolean checking variable to be false\*/

checkCorrect = false;

/\*prompts user to enter the second integer and stores it into timeDur\*/

printf("Please enter the second integer (time duration): ");

scanf("%d", &timeDur);

/\*extracts the digit from the tens column from input and

checks if it's between -5 and 5 (inclusive\*/

digitTens = (timeDur / 10) %10;

if ((digitTens <=5) && (digitTens >= -5))

{

checkCorrect = true;

}

}

/\*adds the time duration to the time of day\*/

result = timeDay + timeDur;

/\*mods result so that it will be in proper 24 hour time\*/

result = result % 2400;

/\*if the result is a negative number, converts it to proper 24

hour time\*/

if(result < 0)

{

result = 2400 + result;

}

/\*print the results to the screen in 4 digit time\*/

printf("Time of day + time duration is : %.4d", result);

return 0;

}

**obelix.gaul.csd.uwo.ca[43]% prog1**

**Please enter the first integer (time of day on a 24 hour clock): 456**

**Please enter the second integer (time duration): -500**

**Time of day + time duration is : 2356**

**obelix.gaul.csd.uwo.ca[44]% prog1**

**Please enter the first integer (time of day on a 24 hour clock): 1234**

**Please enter the second integer (time duration): +3750**

**Time of day + time duration is : 0184**

**obelix.gaul.csd.uwo.ca[45]% prog1**

**Please enter the first integer (time of day on a 24 hour clock): 1234**

**Please enter the second integer (time duration): -3750**

**Time of day + time duration is : 2284**

**obelix.gaul.csd.uwo.ca[46]% prog1**

**Please enter the first integer (time of day on a 24 hour clock): 123**

**Please enter the second integer (time duration): 456**

**Time of day + time duration is : 0579**

**obelix.gaul.csd.uwo.ca[47]% prog1**

**Please enter the first integer (time of day on a 24 hour clock): 3**

**Please enter the second integer (time duration): +4**

**Time of day + time duration is : 0007**

**obelix.gaul.csd.uwo.ca[48]% prog1**

**Please enter the first integer (time of day on a 24 hour clock): 1234**

**Please enter the second integer (time duration): -1250**

**Time of day + time duration is : 2384**

**obelix.gaul.csd.uwo.ca[49]% prog1**

**Please enter the first integer (time of day on a 24 hour clock): 6420**

**Please enter the first integer (time of day on a 24 hour clock): 2064**

**Please enter the first integer (time of day on a 24 hour clock): -6420**

**Please enter the first integer (time of day on a 24 hour clock): -2064**

**Please enter the first integer (time of day on a 24 hour clock): 0000**

**Please enter the second integer (time duration): +2064**

**Please enter the second integer (time duration): -2064**

**Please enter the second integer (time duration): 0000**

**Time of day + time duration is : 0000**

Start

Initialize variables:

timeDay = 0; timeDur; result; digitTens; checkCorrect = false;

While loop

Reset checkCorrect = false

Is timeDay > 2400 OR timeDay < 0 OR checkCorrect = false?

Prompts user for input

YES

Extract digit in 10’s column (timeDay/10)mod10

Store in timeDay

Done

NO

YES

Is the ten’s digit <=5?

checkCorrect = true

Reset checkCorrect = false

While loop

NO

Reset checkCorrect = false

Prompts user for input

Store in timeDur

Is checkCorrect = false?

YES

Extract digit in 10’s column (timeDur/10)mod10

NO

Is the ten’s digit in between -5 and 5 (inclusive)

Done

NO

YES

checkCorrect = true

Add the duration to the time of day and make sure it is written in 2400 time

result = (timeDay + timeDur) mod 2400

NO

YES

Print the result

Return 0 (exit status)

Terminate

Makes sure that the result is a positive number between 0000 and 2400:

(result = 2400 – result)

Is result <0?

**Question 2:**

#include <stdio.h>

/\*Vivian Lam, CS2211 Assignment3: Program 2\*/

/\* Name: Program2

Purpose: calculates the remaining balance on a load after each of the first n monthly payments\*/

int main (void)

{

/\*declares and intializes variables\*/

float loan = -1;

float yearlyInterest = -1;

float monthlyInterest = -1;

float monthlyPayment =-1 ;

int n = -1; //number of monthly payments

float balance; //calculates the balance

int counter = 1; //counter to increment and keep track of the loop

//iterations

float lastPay;//variable to display the last payment if loan is

//paid off

/\*promots the user to enter the amounf of loan, yearly interest rate,

monthly payment, and n (number of monthly payments) \*/

/\*loops to make sure the entered values are positive\*/

while(loan < 0)

{

printf("Please enter the loan value: ");

scanf("%f", &loan);

}

while(yearlyInterest < 0)

{

printf("Please enter the yearly interest value: ");

scanf("%f", &yearlyInterest);

}

while(monthlyPayment < 0)

{

printf("Please enter the monthly payment value: ");

scanf("%f", &monthlyPayment);

}

while(n < 0) //note: if the value entered is a float, rounds

//down to nearest integer

{

printf("Please enter the value of the number of monthly payments: ");

scanf("%d", &n);

}

/\*makes the yearlyInterest a percent and sets the values fo monthlyInterest and balance\*/

yearlyInterest = yearlyInterest / 100;

monthlyInterest = yearlyInterest / 12;

balance = loan;

/\*loop to display for all the monthly payments, or until loan is paid off\*/

while((counter <=n) && (balance >=0)){

/\*sets the value of the last payment\*/

lastPay = balance + (balance \*monthlyInterest);

/\*calculates the value for balance\*/

balance = balance + (balance \* monthlyInterest) - monthlyPayment;

/\*displays each balance with two digits after the decial point\*/

printf ("The balance after %d payment(s): \n", counter);

/\*these conditional statements ensure that only positive values and 0 are printed\*/

if(balance<0){ //if the balance is negative, it will

//display 0.00

printf("0.00 \n");

}

else{ //otherwise print the amount for balance

printf("%.2f \n",balance);

}

/\*if the loan is paid off then reports the amount of the

last payment with two digits after the decimal\*/

if(balance <=0)

{

printf("The amount of the last payment is: %.2f \n", lastPay);

}

counter++; //increments counter

}

return 0;

}

**obelix.gaul.csd.uwo.ca[48]% prog2**

**Please enter the loan value: 12345**

**Please enter the yearly interest value: 12**

**Please enter the monthly payment value: 1234**

**Please enter the value of the number of monthly payments: 15**

**The balance after 1 payment(s):**

**11234.45**

**The balance after 2 payment(s):**

**10112.79**

**The balance after 3 payment(s):**

**8979.92**

**The balance after 4 payment(s):**

**7835.72**

**The balance after 5 payment(s):**

**6680.08**

**The balance after 6 payment(s):**

**5512.88**

**The balance after 7 payment(s):**

**4334.01**

**The balance after 8 payment(s):**

**3143.35**

**The balance after 9 payment(s):**

**1940.78**

**The balance after 10 payment(s):**

**726.19**

**The balance after 11 payment(s):**

**0.00**

**The amount of the last payment is: 733.45**

**obelix.gaul.csd.uwo.ca[49]% prog2**

**Please enter the loan value: 12345**

**Please enter the yearly interest value: 12**

**Please enter the monthly payment value: 543.21**

**Please enter the value of the number of monthly payments: 15**

**The balance after 1 payment(s):**

**11925.24**

**The balance after 2 payment(s):**

**11501.28**

**The balance after 3 payment(s):**

**11073.08**

**The balance after 4 payment(s):**

**10640.61**

**The balance after 5 payment(s):**

**10203.80**

**The balance after 6 payment(s):**

**9762.63**

**The balance after 7 payment(s):**

**9317.05**

**The balance after 8 payment(s):**

**8867.01**

**The balance after 9 payment(s):**

**8412.47**

**The balance after 10 payment(s):**

**7953.38**

**The balance after 11 payment(s):**

**7489.71**

**The balance after 12 payment(s):**

**7021.39**

**The balance after 13 payment(s):**

**6548.40**

**The balance after 14 payment(s):**

**6070.67**

**The balance after 15 payment(s):**

**5588.17**

**obelix.gaul.csd.uwo.ca[50]% prog2**

**Please enter the loan value: 54321**

**Please enter the yearly interest value: 12**

**Please enter the monthly payment value: 543.21**

**Please enter the value of the number of monthly payments: 15**

**The balance after 1 payment(s):**

**54321.00**

**The balance after 2 payment(s):**

**54321.00**

**The balance after 3 payment(s):**

**54321.00**

**The balance after 4 payment(s):**

**54321.00**

**The balance after 5 payment(s):**

**54321.00**

**The balance after 6 payment(s):**

**54321.00**

**The balance after 7 payment(s):**

**54321.00**

**The balance after 8 payment(s):**

**54321.00**

**The balance after 9 payment(s):**

**54321.00**

**The balance after 10 payment(s):**

**54321.00**

**The balance after 11 payment(s):**

**54321.00**

**The balance after 12 payment(s):**

**54321.00**

**The balance after 13 payment(s):**

**54321.00**

**The balance after 14 payment(s):**

**54321.00**

**The balance after 15 payment(s):**

**54321.00**

**obelix.gaul.csd.uwo.ca[51]% prog2**

**Please enter the loan value: 54321**

**Please enter the yearly interest value: 12**

**Please enter the monthly payment value: 321**

**Please enter the value of the number of monthly payments: 15**

**The balance after 1 payment(s):**

**54543.21**

**The balance after 2 payment(s):**

**54767.64**

**The balance after 3 payment(s):**

**54994.32**

**The balance after 4 payment(s):**

**55223.26**

**The balance after 5 payment(s):**

**55454.50**

**The balance after 6 payment(s):**

**55688.04**

**The balance after 7 payment(s):**

**55923.92**

**The balance after 8 payment(s):**

**56162.16**

**The balance after 9 payment(s):**

**56402.78**

**The balance after 10 payment(s):**

**56645.80**

**The balance after 11 payment(s):**

**56891.26**

**The balance after 12 payment(s):**

**57139.18**

**The balance after 13 payment(s):**

**57389.57**

**The balance after 14 payment(s):**

**57642.46**

**The balance after 15 payment(s):**

**57897.89**

**obelix.gaul.csd.uwo.ca[52]% prog2**

**Please enter the loan value: 1000**

**Please enter the yearly interest value: 12**

**Please enter the monthly payment value: 800**

**Please enter the value of the number of monthly payments: 15**

**The balance after 1 payment(s):**

**210.00**

**The balance after 2 payment(s):**

**0.00**

**The amount of the last payment is: 212.10**

Start

Initialize variables:

loan = -1; yearlyInterest = -1; monthlyInterest = -1; monthlyPayment =-1 ; n = -1; balance; counter = 1; lastPay;

While loop

Store value into loan

Prompts user for input

Is loan <0?

YES

NO

NO

NO

NO

YES

YES

YES

Is n<0?

Is monthlyPayment<0?

Done

Done

Is yearlyInterest <0?

Store value into yearlyInterest

Prompts user for input

While loop

Store value into n

Prompts user for input

Done

While loop

Prompts user for input

While loop

Store value into monthlyPayment

Done

NO

YES

YES

NO

Print 0.00 and lastPay

Print balance

Is balance <=0?

Return 0 (exit status)

Terminate

Done

Print counter

Calculates the value for balance: balance = balance + (balance \* monthlyInterest) - monthlyPayment;

Sets the value of lastPay: lastPay = balance + (balance \*monthlyInterest);

Is counter <=n & balance >=0?

While loop

yearlyInterest = yearlyInterest / 100;

monthlyInterest = yearlyInterest / 12;

balance = loan;

**Question 3**

#include <stdio.h>

#include <stdbool.h>

/\*Vivian Lam, CS2211 Assignment3: Program 3\*/

/\* Name: Program3

Purpose: approximmates the value of the constant e\*/

int main (void)

{

/\*declares and initializes variables\*/

double accuracy = -1; //used to store the precision number the user wants

double eVal = 1; //the value of e that will be approximated

double fact =1; //variable to store the factorial

long nextFact =0; //stores the value which the next

//factorial should be multiplied by

int numbTerms =1; //variable to keep track of number of terms

bool loopChecker = true; //boolean to check if loop should

//continue looping

/\*loops to make sure the user enters a valid positive number\*/

while(accuracy<0){

printf("Please enter a positive decimal number (to represent how precise you want e to be approximated): \n");

scanf("%lf", &accuracy); //stores user input into this

//variable

}

/\*loops until the term to be added becomes less than a small

positive float point numbered entered by the user\*/

while ((1/fact >= accuracy) && (loopChecker == true)){

nextFact = nextFact + 1;//increments the value of nextFact

fact = fact \* nextFact; //sets the new value of fact

eVal = eVal + ( 1 / fact ); //sets the new value of e by

//adding the new factorial decimal

numbTerms++; //increments the value of numbTerms by 1

/\*checks to see if the next term after this will violate the condition of the loop. if so then set the value of

loopChecker to be false so that the loop terminates\*/

if (1 / (fact\*(nextFact+1)) <accuracy){

loopChecker = false;

}

}

/\*prints the value of e with 15 digits after the decimal\*/

printf("this is the value of e: %.15lf \n", eVal);

//prints the number of terms required to reach ths value

printf("Number of terms required to read this value: %d \n", numbTerms);

return 0; //returns 0

}

**obelix.gaul.csd.uwo.ca[46]% prog3**

**Please enter a positive decimal number (to represent how precise you want e to be approximated):**

**-1**

**Please enter a positive decimal number (to represent how precise you want e to be approximated):**

**-0.1**

**Please enter a positive decimal number (to represent how precise you want e to be approximated):**

**0.1**

**this is the value of e: 2.666666666666667**

**Number of terms required to read this value: 4**

**obelix.gaul.csd.uwo.ca[47]% prog3**

**Please enter a positive decimal number (to represent how precise you want e to be approximated):**

**0.01**

**this is the value of e: 2.708333333333333**

**Number of terms required to read this value: 5**

**obelix.gaul.csd.uwo.ca[48]% prog3**

**Please enter a positive decimal number (to represent how precise you want e to be approximated):**

**0.001**

**this is the value of e: 2.718055555555555**

**Number of terms required to read this value: 7**

**obelix.gaul.csd.uwo.ca[49]% prog3**

**Please enter a positive decimal number (to represent how precise you want e to be approximated):**

**0.0001**

**this is the value of e: 2.718253968253968**

**Number of terms required to read this value: 8**

**obelix.gaul.csd.uwo.ca[50]% prog3**

**Please enter a positive decimal number (to represent how precise you want e to be approximated):**

**0.00001**

**this is the value of e: 2.718278769841270**

**Number of terms required to read this value: 9**

**obelix.gaul.csd.uwo.ca[51]% prog3**

**Please enter a positive decimal number (to represent how precise you want e to be approximated):**

**0.000001**

**this is the value of e: 2.718281525573192**

**Number of terms required to read this value: 10**

**obelix.gaul.csd.uwo.ca[52]% prog3**

**Please enter a positive decimal number (to represent how precise you want e to be approximated):**

**0.0000001**

**this is the value of e: 2.718281801146385**

**Number of terms required to read this value: 11**

**obelix.gaul.csd.uwo.ca[53]% prog3**

**Please enter a positive decimal number (to represent how precise you want e to be approximated):**

**0.00000001**

**this is the value of e: 2.718281826198493**

**Number of terms required to read this value: 12**

**obelix.gaul.csd.uwo.ca[54]% prog3**

**Please enter a positive decimal number (to represent how precise you want e to be approximated):**

**0.000000001**

**this is the value of e: 2.718281828286169**

**Number of terms required to read this value: 13**

**obelix.gaul.csd.uwo.ca[55]% prog3**

**Please enter a positive decimal number (to represent how precise you want e to be approximated):**

**0.0000000001**

**this is the value of e: 2.718281828446759**

**Number of terms required to read this value: 14**

**obelix.gaul.csd.uwo.ca[56]% prog3**

**Please enter a positive decimal number (to represent how precise you want e to be approximated):**

**0.00000000001**

**this is the value of e: 2.718281828458230**

**Number of terms required to read this value: 15**

NO

Is the next factorial reciprocal term < accuracy?

Increase the value of numbTerm by 1

YES

NO

YES

NO

YES

Done

While loop

Is the current factorial reciprocal >= the value of accuracy & is loopChecker = true?

Add the next reciprocal factorial term to the current value of e:

nextFact = nextFact+1

fact = fact \*nextFact

eVal = eVal + 1/fact

loopChecker = false

Terminate

Return 0 (exit status)

Done

Store into accuracy

Prompts user for input

Is accuracy<0?

While loop

Initialize variables:

accuracy = -1; eVal = 1; fact =1; nextFact =0; numbTerms =1; loopChecker = true;

Start