Assignment #3
By: Josh Jackson

Question #1

Code:

```c
#include <stdio.h>
#include <stdlib.h>

#define FALSE 1
#define TRUE 0
#define Subtract 1
#define Add 0


int main(void){
    //defining variables

    //time of day
    int tod;
    //time duration
    int td;
    //boolean operators for loops and switch statement
    int flag, operation;
    //minutes and hours separated for TOD and TD
    int part1_tod, part2_tod, part1_td, part2_td;
    //minutes and hours for printing end values
    int minutes, hours;
    //temp hours for adjusting time, carry over effect
    int temphours;

    //do-while to take input of Time of Day and separate into two parts
    do {
        printf("Enter a Time of Day based on 24 hour clock: \n");
        fflush(stdout);
        scanf("%d", &tod);


        //operation to separate the minutes and hours of TOD
        part1_tod = tod / 100;
        part2_tod = tod % 100;

        //Error Checking
        if (part1_tod < 00 || part1_tod > 23) {
            printf("Input Invalid.\n");
            fflush(stdout);
            flag = FALSE;
        }
        else if (part2_tod < 00 || part2_tod >= 60){
            printf("Input Invalid. \n");
            fflush(stdout);
```

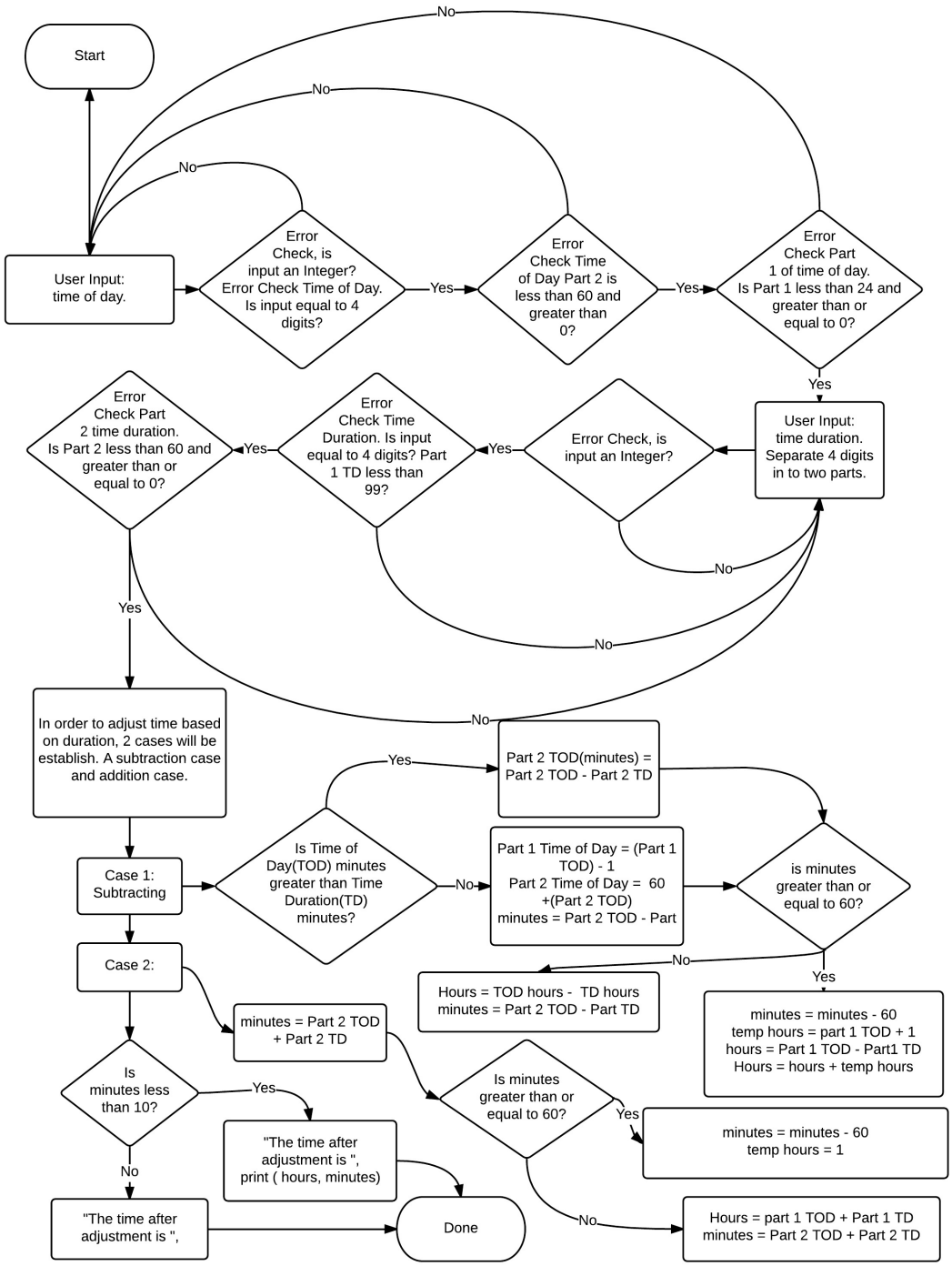
```
            flag = FALSE;
        }
        else{
            flag = TRUE;
        }
    }while (flag == FALSE);



    //do-while loop to get user input for Time Duration and separate minutes and hours
    do {
        printf("Enter a Time Duration to adjust 24 hour clock by: \n");
        fflush(stdout);
        scanf("%d", &td);

        //assigning operation value for switch statement
        if (td < 0) {
            operation = Subtract;
        }
        else{
            operation = Add;
        }

        //absolute value operation to make everything positive after assigning which
method to follow (add, or sub)
        td = abs(td);

        //operation to separate into two parts minutes and hours
        part1_td = td / 100;
        part2_td = td % 100; //CHANGED THIS

        //Error Checking
        if (part1_td > 99) {
            printf("Input Invalid. \n");
            fflush(stdout);
            flag = FALSE;
        }
        else if (part2_td < 0 || part2_td >= 60){
            printf("Invalid Input. \n");
            fflush(stdout);
            flag = FALSE;
        }
        else{
            flag = TRUE;
        }
    } while(flag == FALSE);



    //switch operation to determine how to adjust time
    switch(operation){
```

```
        //if time duration is negative
        case Subtract:
            if (part2_td > part2_tod) {
                part1_tod = part1_tod - 1;
                part2_tod = 60 + part2_tod;
                minutes = part2_tod - part2_td;
                //temphours = part1_tod + 1;
                hours = abs(part1_tod - part1_td);
                //hours = hours + temphours;

            }
            else if (part2_td <= part2_tod) {
                minutes = part2_tod - part2_td;
                hours = part1_tod - abs(part1_td);
            }

            if (minutes >= 60) {
                minutes = minutes - 60;
                temphours = part1_tod + 1;
                hours= abs(part1_tod - part1_td);
                hours = hours + temphours;
            }

            //if minutes is less the 10 it will print with a 0 in front avoidin time
such 10:1
            //if hours are greater then 24 it will wrap around to 00:00
            if(minutes < 10 && hours <=23){
                if (hours >= 10) {
                    printf("The time after adjustment is %d:0%d\n", hours, minutes);
                    fflush(stdout);
                }
                else{
                    printf("The time after adjustment is 0%d:0%d\n", hours, minutes);
                    fflush(stdout);
                }
                break;
            }
            if (minutes < 10 && hours > 23){
                hours = hours - (hours / 24)*24;
                if (hours >= 10) {
                    printf("The time after adjustment is %d:0%d\n", hours, minutes);
                }
                else{
                    printf("The time after adjustment is 0%d:0%d\n", hours, minutes);
                    fflush(stdout);
                }
                break;
            }
            if (minutes >= 10 && hours > 23) {
                hours = hours - (hours / 24)*24;
```

```
            if (hours >= 10) {
                printf("The time after adjustment is %d:%d\n", hours, minutes);
                fflush(stdout);
            }
            else{
                printf("The time after adjustment is 0%d:%d\n", hours, minutes);
                fflush(stdout);
            }
            break;
        }

        if (minutes >= 10 && hours <= 23) {
            //hours = hours - (hours / 24)*24;
            if (hours >= 10) {
                printf("The time after adjustment is %d:%d\n", hours, minutes);
                fflush(stdout);
            }
            else{
                printf("The time after adjustment is 0%d:%d\n", hours, minutes);
                fflush(stdout);
            }
            break;
        }
        break;

        //if time duration is positive
    case Add:
        minutes = part2_tod + part2_td;
        if (minutes >= 60){
            minutes = minutes - 60;
            temphours = 1;
            hours = part1_tod + part1_td + temphours;
        }
        else if (minutes <60){
            minutes = part2_tod + part2_td;
            hours = part1_tod + part1_td;
        }

        if(minutes < 10 && hours <=23){
            if (hours < 10) {
                printf("The time after adjustment is 0%d:0%d\n", hours, minutes);
                fflush(stdout);
            }
            else{
                printf("The time after adjustment is %d:0%d\n", hours, minutes);
                fflush(stdout);
            }
            break;
        }

        if (minutes < 10 && hours > 23) {
```

```
                hours = hours - (hours / 24)*24;
                if (hours < 10) {
                    printf("The time after adjustment is 0%d:0%d\n", hours, minutes);
                    fflush(stdout);
                }
                else{
                    printf("The time after adjustment is %d:0%d\n", hours, minutes);
                    fflush(stdout);
                }
                break;
            }

            if (minutes >= 10 && hours <= 23) {
                hours = hours - (hours / 24)*24;
                if (hours < 10) {
                    printf("The time after adjustment is 0%d:%d\n", hours, minutes);
                    fflush(stdout);
                }
                else{
                    printf("The time after adjustment is %d:%d\n", hours, minutes);
                    fflush(stdout);
                }
                break;
            }

            if(minutes >= 10 && hours > 23){
                hours = hours - (hours / 24)*24;
                if (hours < 10) {
                    printf("The time after adjustment is 0%d:%d\n", hours, minutes);
                    fflush(stdout);
                }
                else{
                    printf("The time after adjustment is %d:%d\n", hours, minutes);
                    fflush(stdout);
                }
                break;
            }
            break;
    }
}
```

Flow Chart:


Start
User Input: time of day.
Error Check, is input an Integer? Error Check Time of Day. Is input equal to 4 digits?
Error Check Time of Day Part 2 is less than 60 and greater than 0?
Error Check Part 1 of time of day. Is Part 1 less than 24 and greater than or equal to 0?
User Input: time duration. Separate 4 digits in to two parts.
Error Check, is input an Integer?
Error Check Time Duration. Is input equal to 4 digits? Part 1 TD less than 99?
Error Check Part 2 time duration. Is Part 2 less than 60 and greater than or equal to 0?
In order to adjust time based on duration, 2 cases will be establish. A subtraction case and addition case.
Case 1: Subtracting
Is Time of Day(TOD) minutes greater than Time Duration(TD) minutes?
Part 2 TOD(minutes) = Part 2 TOD - Part 2 TD
Part 1 Time of Day = (Part 1 TOD) - 1
Part 2 Time of Day = 60 +(Part 2 TOD)
minutes = Part 2 TOD - Part
is minutes greater than or equal to 60?
Hours = TOD hours - TD hours
minutes = Part 2 TOD - Part TD
minutes = minutes - 60
temp hours = part 1 TOD + 1
hours = Part 1 TOD - Part1 TD
Hours = hours + temp hours
Case 2:
minutes = Part 2 TOD + Part 2 TD
Is minutes less than 10?
"The time after adjustment is ", print (hours, minutes)
"The time after adjustment is ",
Done
Is minutes greater than or equal to 60?
minutes = minutes - 60
temp hours = 1
Hours = part 1 TOD + Part 1 TD
minutes = Part 2 TOD + Part 2 TD
Yes
No

Test Cases:

Test	Input	Output
1	6420 – 456	**Enter a Time of Day based on 24 hour clock:** 6420 **Input Invalid.** **Enter a Time of Day based on 24 hour clock:** (Repeats question because TOD was invalid)
2	2064 + 456	**Enter a Time of Day based on 24 hour clock:** 2064 **Input Invalid.** **Enter a Time of Day based on 24 hour clock:** (Repeats question TOD was invalid)
3	456 + 2064	**Enter a Time of Day based on 24 hour clock:** 456 **Enter a Time Duration to adjust 24 hour clock by:** 2064 **Input Invalid.** **Enter a Time Duration to adjust 24 hour clock by:** (Repeats question as Time Duration is Invalid)
4	456 + 500	**Enter a Time of Day based on 24 hour clock:** 456 **Enter a Time Duration to adjust 24 hour clock by:** 500 **The time after adjustment is 9:56** Program ended with exit code: 0
5	1234 + 3750	**Enter a Time of Day based on 24 hour clock:** **1234** **Enter a Time Duration to adjust 24 hour clock by:** **3750** **The time after adjustment is 02:24** Program ended with exit code: 0
6	1234 – 3750	**Enter a Time of Day based on 24 hour clock:** 1234 **Enter a Time Duration to adjust 24 hour clock by:** -3750 **The time after adjustment is 02:44** Program ended with exit code: 0
7	1234 – 1250	**Enter a Time of Day based on 24 hour clock:** 1234 **Enter a Time Duration to adjust 24 hour clock by:** -1250 **The time after adjustment is 00:44** Program ended with exit code: 0
8	123 + 456	**Enter a Time of Day based on 24 hour clock:** **123** **Enter a Time Duration to adjust 24 hour clock by:**

		456 **The time after adjustment is 06:19** Program ended with exit code: 0
9	3 + 4	**Enter a Time of Day based on 24 hour clock:** **3** **Enter a Time Duration to adjust 24 hour clock by:** **4** **The time after adjustment is 00:07** Program ended with exit code: 0

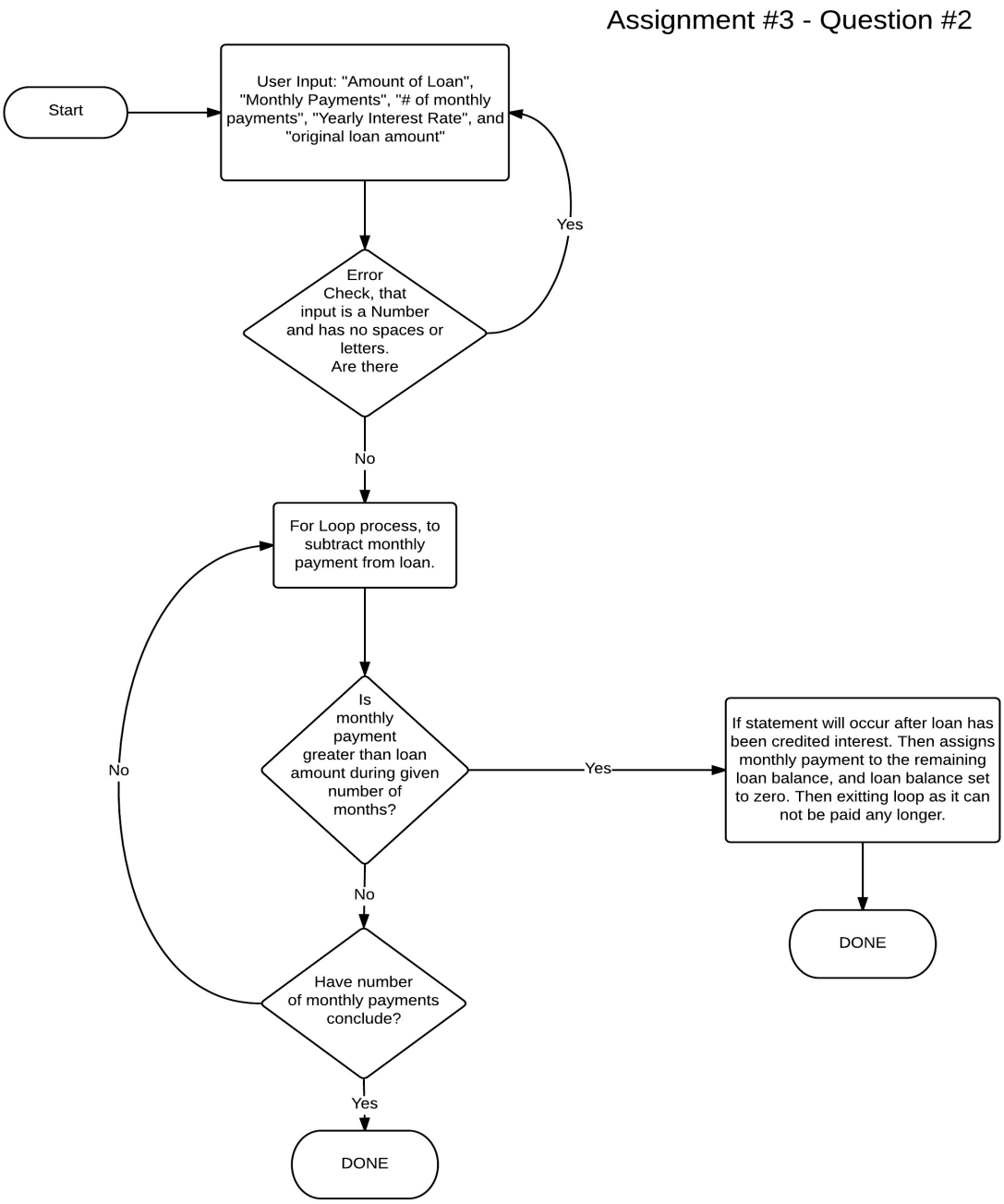
Question 2

Code:

```
#include <stdio.h>
#include <stdlib.h>
#define TRUE 0
#define FALSE 1

int main(void){
    //define variables
    float loan = 0;
    float interest = 0.0, monthly, n;
    int flag;
    char term;

    //boolean value flag=true to reset when error occurs
    flag = FALSE;
    //error handling

    /*start boolean value = false as long as it stay false it
     *can continue through loop flag = true will reset loop to repeat question
     */
    while(flag == FALSE){
        //print question if statement to check it is an integer with no spaces
        printf("Amount of the Loan: \n$");
        if (scanf("%f%c", &loan, &term) != 2 || term != '\n' ){
            fflush(stdout);
            printf("Not Valid Input\n");
            flag = FALSE;
    }
        else{
            flag = TRUE;
        }

        //print question if statement to check it is an integer with no spaces
        printf("Yearly Interest on Loan: \n");
        if (scanf("%f%c", &interest, &term) != 2 || term != '\n' ){
            fflush(stdout);
            printf("Not Valid Input\n");
            flag = FALSE;
        }
        else{
            flag = TRUE;
        }
    }
        //print question if statement to check it is an integer with no spaces
        printf("Amount of Monthly Payment: \n$");
        if (scanf("%f%c", &monthly, &term) != 2 || term != '\n' ){
            fflush(stdout);
            printf("Not an Float\n");
            flag = FALSE;
```

```
        }
        else{
            flag = TRUE;
        }

        //print question if statement to check it is an integer with no spaces
        printf("How many monthly payments: \n");
        if (scanf("%f%c", &n, &term) != 2 || term != '\n' ){
            fflush(stdout);
            printf("Not an Float\n");
            flag = FALSE;
        }
        else{
            flag = TRUE;
        }

            //loop to build balance of loan amount
            for (int i = 1; i <= n; i++) {
                loan += (loan * ((interest/12)/100));
                //if statement to stop loop and print monthly payment as remainder of
balnce and set balance to 0
                if (loan< monthly){
                    monthly=loan;
                    loan=0;
                    printf("The balance after month %d of the payment of $%.2f is
$%.2lf\n", i, monthly, loan);
                    break;
                }
                    loan -= monthly;
                    printf("The balance after month %d of the payment of $%.2f is
$%.2lf\n", i, monthly, loan);
            }
        }
```

Flow Chart:


Assignment #3 - Question #2
Start
User Input: "Amount of Loan", "Monthly Payments", "# of monthly payments", "Yearly Interest Rate", and "original loan amount"
Error Check, that input is a Number and has no spaces or letters. Are there
Yes
No
For Loop process, to subtract monthly payment from loan.
Is monthly payment greater than loan amount during given number of months?
Yes
If statement will occur after loan has been credited interest. Then assigns monthly payment to the remaining loan balance, and loan balance set to zero. Then exitting loop as it can not be paid any longer.
DONE
No
Have number of monthly payments conclude?
No
Yes
DONE

Test Cases

Test	Input	Output
1	Loan Amount: $12345 Yearly Interest: 12% Monthly Payment: $1234 15 Payments	**Amount of the Loan:** **$**12345 **Yearly Interest on Loan:** 12 **Amount of Monthly Payment:** **$**1234 **How many monthly payments:** 15 **The balance after month 1 of the payment of $1234.00 is $11234.45** **The balance after month 2 of the payment of $1234.00 is $10112.79** **The balance after month 3 of the payment of $1234.00 is $8979.92** **The balance after month 4 of the payment of $1234.00 is $7835.72** **The balance after month 5 of the payment of $1234.00 is $6680.08** **The balance after month 6 of the payment of $1234.00 is $5512.88** **The balance after month 7 of the payment of $1234.00 is $4334.01** **The balance after month 8 of the payment of $1234.00 is $3143.35** **The balance after month 9 of the payment of $1234.00 is $1940.78** **The balance after month 10 of the payment of $1234.00 is $726.19** **The balance after month 11 of the payment of $733.45 is $0.00** Program ended with exit code: 0
2	Loan Amount: $12345 Yearly Interest: 12% Monthly Payment: $543.21 15 Payments	**Amount of the Loan:** **$**12345 **Yearly Interest on Loan:** 12 **Amount of Monthly Payment:** **$**543.21 **How many monthly payments:** 15 **The balance after month 1 of the payment of $543.21 is $11925.24** **The balance after month 2 of the payment of $543.21 is $11501.28** **The balance after month 3 of the payment of $543.21 is $11073.08** **The balance after month 4 of the payment of $543.21 is $10640.61** **The balance after month 5 of the payment**

		of $543.21 is $10203.80 **The balance after month 6 of the payment of $543.21 is $9762.63** **The balance after month 7 of the payment of $543.21 is $9317.05** **The balance after month 8 of the payment of $543.21 is $8867.01** **The balance after month 9 of the payment of $543.21 is $8412.47** **The balance after month 10 of the payment of $543.21 is $7953.38** **The balance after month 11 of the payment of $543.21 is $7489.71** **The balance after month 12 of the payment of $543.21 is $7021.39** **The balance after month 13 of the payment of $543.21 is $6548.40** **The balance after month 14 of the payment of $543.21 is $6070.67** **The balance after month 15 of the payment of $543.21 is $5588.17**
3	Loan Amount: $54321 Yearly Interest: 12% Monthly Payment: $543.21 15 Payments	**Amount of the Loan:** **$**54321 **Yearly Interest on Loan:** 12 **Amount of Monthly Payment:** **$**543.21 **How many monthly payments:** 15 **The balance after month 1 of the payment of $543.21 is $54321.00** **The balance after month 2 of the payment of $543.21 is $54321.00** **The balance after month 3 of the payment of $543.21 is $54321.00** **The balance after month 4 of the payment of $543.21 is $54321.00** **The balance after month 5 of the payment of $543.21 is $54321.00** **The balance after month 6 of the payment of $543.21 is $54321.00** **The balance after month 7 of the payment of $543.21 is $54321.00** **The balance after month 8 of the payment of $543.21 is $54321.00** **The balance after month 9 of the payment of $543.21 is $54321.00** **The balance after month 10 of the payment of $543.21 is $54321.00** **The balance after month 11 of the payment of $543.21 is $54321.00**

		The balance after month 12 of the payment of $543.21 is $54321.00 **The balance after month 13 of the payment of $543.21 is $54321.00** **The balance after month 14 of the payment of $543.21 is $54321.00** **The balance after month 15 of the payment of $543.21 is $54321.00** Program ended with exit code: 0 (balance remains the same as monthly interest is equal to monthly payments)
4	Loan Amount: $54321 Yearly Interest: 12% Monthly Payment: $321 15 Payments	**Amount of the Loan:** **$**54321 **Yearly Interest on Loan:** 12 **Amount of Monthly Payment:** **$**321 **How many monthly payments:** 15 **The balance after month 1 of the payment of $321.00 is $54543.21** **The balance after month 2 of the payment of $321.00 is $54767.64** **The balance after month 3 of the payment of $321.00 is $54994.32** **The balance after month 4 of the payment of $321.00 is $55223.26** **The balance after month 5 of the payment of $321.00 is $55454.50** **The balance after month 6 of the payment of $321.00 is $55688.04** **The balance after month 7 of the payment of $321.00 is $55923.92** **The balance after month 8 of the payment of $321.00 is $56162.16** **The balance after month 9 of the payment of $321.00 is $56402.78** **The balance after month 10 of the payment of $321.00 is $56645.80** **The balance after month 11 of the payment of $321.00 is $56891.26** **The balance after month 12 of the payment of $321.00 is $57139.18** **The balance after month 13 of the payment of $321.00 is $57389.57** **The balance after month 14 of the payment of $321.00 is $57642.46** **The balance after month 15 of the payment of $321.00 is $57897.89** Program ended with exit code: 0 (monthly payments less than interest so

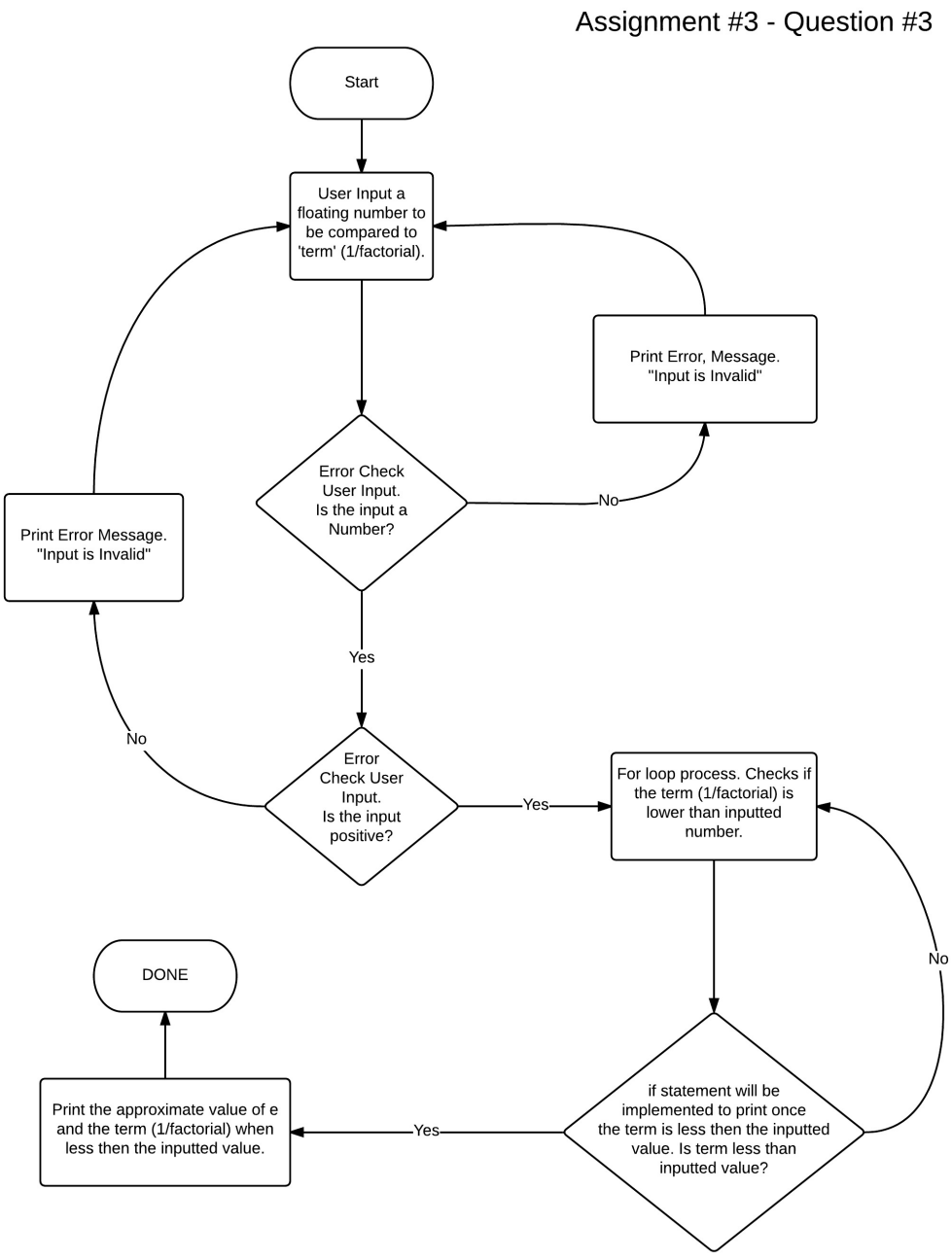
		balance increases)

Question 3

Code:

```
#define FALSE 1
#define TRUE 0

#include <stdio.h>

int main(void){
    //defining variables
    double number;
    double e = 1;
    double temp;
    int i, j;
    int flag;
    //char term;

    flag = FALSE;
    while (flag == FALSE) {
    printf("Enter a floating number (anything with lots of decimal places)\n");
        if(scanf("%lf", &number) != 1 || number <= 0){
            printf("Input is Invalid\n");
            flag = FALSE;
        }
        else{
            flag = TRUE;
        }

    }

    //outer loop calculating how many terms for factorial
    for (i = 1; i < 20; i++) {
            //defining i as temp to be used in next loop for factorial
            temp = i;
        //loop uses i-1 time temp to process factorial continuously
        for (j = i-1; j > 0; j--){
            temp = temp * j;
        }
            //e is equal to e + 1/factorial
            e = e + (1/temp);
        //if statement to end loop when factorial is less then inputted number from user
        if((1/temp) < number){
            printf("e is equal to %.15lf, it took %d terms to reach a number less than
inputted value.\n", e, i);
            break;
        }
    }
}
```

Flow Chart:


Assignment #3 - Question #3
Start
User Input a floating number to be compared to 'term' (1/factorial).
Print Error, Message. "Input is Invalid"
Error Check User Input. Is the input a Number?
No
Print Error Message. "Input is Invalid"
Yes
No
Error Check User Input. Is the input positive?
Yes
For loop process. Checks if the term (1/factorial) is lower than inputted number.
No
DONE
Print the approximate value of e and the term (1/factorial) when less then the inputted value.
Yes
if statement will be implemented to print once the term is less then the inputted value. Is term less than inputted value?

Test Cases:

Test	Input	Output
1	0.01	**Enter a floating number (anything with lots of decimal places)** 0.01 **e is equal to 2.716666666666666, it took 5 terms to reach a number less than inputted value.** Program ended with exit code: 0
2	0.001	**Enter a floating number (anything with lots of decimal places)** 0.001 **e is equal to 2.718253968253968, it took 7 terms to reach a number less than inputted value.** Program ended with exit code: 0
3	0.0001	**Enter a floating number (anything with lots of decimal places)** 0.0001 **e is equal to 2.718278769841270, it took 8 terms to reach a number less than inputted value.** Program ended with exit code: 0
4	0.00001	**Enter a floating number (anything with lots of decimal places)** 0.00001 **e is equal to 2.718281525573192, it took 9 terms to reach a number less than inputted value.** Program ended with exit code: 0
5	0.000001	**Enter a floating number (anything with lots of decimal places)** 0.000001 **e is equal to 2.718281801146385, it took 10 terms to reach a number less than inputted value.** Program ended with exit code: 0
6	0.0000001	**Enter a floating number (anything with lots of decimal places)** 0.0000001 **e is equal to 2.718281826198493, it took 11 terms to reach a number less than inputted value.** Program ended with exit code: 0
7	0.00000001	**Enter a floating number (anything with lots of decimal places)** 0.00000001 **e is equal to 2.718281828286169, it took 12 terms to reach a number less than inputted value.** Program ended with exit code: 0

8	0.000000001	**Enter a floating number (anything with lots of decimal places)** **0.000000001** **e is equal to 2.718281828446759, it took 13 terms to reach a number less than inputted value.** **Program ended with exit code: 0**
9	0.0000000001	**Enter a floating number (anything with lots of decimal places)** 0.0000000001 **e is equal to 2.718281828458230, it took 14 terms to reach a number less than inputted value.** Program ended with exit code: 0
10	0.00000000001	**Enter a floating number (anything with lots of decimal places)** 0.00000000001 **e is equal to 2.718281828458995, it took 15 terms to reach a number less than inputted value.** Program ended with exit code: 0