Homework 4

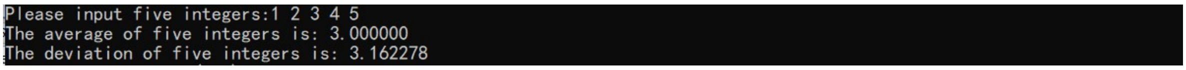
Problem 1

Program 1

```
/*---------------------------------------
Module name: program_1.c
Description: For a number of reasons, when testing a product as part of the manufacturing cycle, we
are interested how the values of certain signals may vary based upon such things as
different part vendors, part date codes, or with the manufacturing process itself. Two
```

```
important measures are the average values of such signals and standard
deviation of those
values across a production run or between runs. Write a function ave5 that
returns the
average of five integer values passed to it as parameters. Write a similar
function stdev5
that returns the standard deviation of its five integer parameters.
Author: Bo Yue
Rev. 0 03 Aug 2019
----------------------------------------*/

#include <stdio.h>
#include <math.h>//for maths calculation

//declaration of function ave5() & function stdev5()
float ave5(int a, int b, int c, int d, int e);
float stdev5(int a, int b, int c, int d, int e, float* f);

int main()
{
    int ave_1       = 0;
    int ave_2       = 0;
    int ave_3       = 0;
    int ave_4       = 0;
    int ave_5       = 0;
    float average   = 0.0;
    float deviation = 0.0;

    printf("Please input five integers:");
    scanf("%d %d %d %d %d", &ave_1, &ave_2, &ave_3, &ave_4, &ave_5);

    average = ave5(ave_1, ave_2, ave_3, ave_4, ave_5);
    printf("The average of five integers is: %f\n", average);

    deviation = stdev5(ave_1, ave_2, ave_3, ave_4, ave_5, &average);
    printf("The deviation of five integers is: %f", deviation);

    return 0;
}

//definition of function ave5()
float ave5(int a, int b, int c, int d, int e)
{
    float ave = 0.0;
    ave = (a + b + c + d + e) / 5;
    return ave;
}

//definition of function stdev5()
float stdev5(int a, int b, int c, int d, int e, float* f)
{
    float dev = 0.0;
    dev = sqrt((a-*f)*(a-*f)+(b-*f)*(b-*f)+(c-*f)*(c-*f)+(d-*f)*(d-*f)+(e-
*f)*(e-*f));
    return dev;
}
```

Test Case



```
Please input five integers:1 2 3 4 5
The average of five integers is: 3.000000
The deviation of five integers is: 3.162278
```

- The function stdev5() utilizes the Least Square Method to calculate the deviation of five integers

Characteristics and Side Effects

- Maybe the average of five integers does not need that much number of decimals, perhaps we can limit the number of the result if the number behind present bit are all zeros.

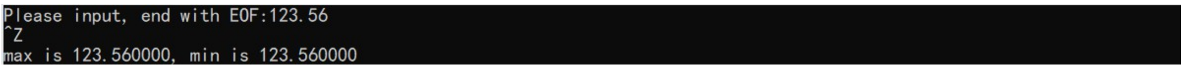
Problem 2

Program 2

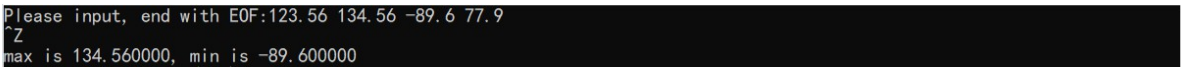
```
/*----------------------------------------
Module name: program_2.c
Description: This is a program to print the largest and smallest values of
its input. That is, your user is
prompted to enter a value until they get bored. At that point, print out the
largest and smallest
values entered.
Author: Bo Yue
```

```
Rev. 0 03 Aug 2019
----------------------------------------*/

#include <stdio.h>

int main()
{
    double receive = 0.0;// get the input value
    double max     = 0.0;
    double min     = 0.0;

    // max and min have to be initialized with the first input
    printf("Please input, end with EOF:");
    if(1 == scanf("%lf", &receive))
    {
          max = receive;
          min = receive;
    }
    else
    {
           printf("Please input again:");
    }

    // the second and the following input
    while(1 == scanf("%lf", &receive))
    {
        if(receive > max) {max = receive;}
        if(receive < min) {min = receive;}
    }
    printf("max is %lf, min is %lf", max, min);
    return 0;
}
```

Test Cases



```
Please input, end with EOF:123.56
^Z
max is 123.560000, min is 123.560000
```



```
Please input, end with EOF:123.56 134.56 -89.6 77.9
^Z
max is 134.560000, min is -89.600000
```

- The user can input as their will.

Characteristics and Side Effects

- Precision is limited due to the type double, and although the output is mathematically the same as the input, they are not exactly the same. Perhaps a char list which gets the number of decimal bits for every input can be applied to solve this issue, but the code may look upset.

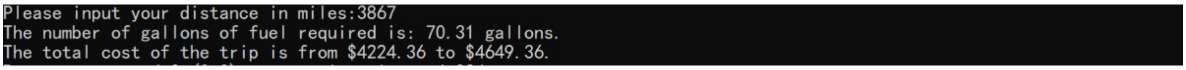
Problem 3

Program 3

```
/*----------------------------------------
Module name: program_3.c
Description: Bill is planning a major motorcycle trip all over the west coast at the end of the quarter.
He wants to start saving money for his trip now and wants an estimate as to what the trip
might cost. Write a program to help him with that. For now, he is focusing on the fuel
and operating expenses for his bike and the potential cost of food.
Write a function that calculates the number of gallons of fuel required for traveling a
specific distance in miles. His motorcycle gets 55.0 miles to the gallon.
On the road, he averages 300-500 miles a day and food costs for each day he is on the
road average $85.00. If the average price of fuel where he will be traveling is $4.32 a
gallon and its costs $0.86 to operate the bike each mile, write a second function to
calculate the range in dollars for the total cost of the trip. He will ultimately decide on
the number of days for the trip based upon total cost.
Author: Bo Yue
Rev. 0 03 Aug 2019
----------------------------------------*/

#include <stdio.h>

//symbolic constants
#define Miles_Per_Gallon 55.0
#define Min_Miles_Daily 300
#define Max_Miles_Daily 500
```

```c
#define Price_Per_Gallon 4.32
#define Bike_Per_Mile 0.86
#define Food_cost 85.00

//declaration of function gallon() & moneyRange()
double gallonTotal(int* a);
void moneyRange(int* a, double *b, double* c, int* d);

int main()
{
    int distance = 0;
    double gallon = 0.0;
    double min_total_cost = 0.0;
    double max_total_cost = 0.0;

    printf("Please input your distance in miles:");
    scanf("%d", &distance);

    gallon = gallonTotal(&distance);
    moneyRange(&distance, &min_total_cost, &max_total_cost, &distance);

    printf("The number of gallons of fuel required is: %lf gallons.\n",
gallon);
    printf("The total cost of the trip is from $%.2lf to $%.2lf.",
min_total_cost, max_total_cost);

    return 0;
}

//definition of gallon()
double gallonTotal(int *a)
{
    return (*a) / Miles_Per_Gallon;
}

//definition of moneyRange()
void moneyRange(int* a, double* b, double* c, int *d)
{
    *b = (*d / Miles_Per_Gallon) * Price_Per_Gallon + *d * Bike_Per_Mile +
(*d / Max_Miles_Daily) * Food_cost;
    *c = (*d / Miles_Per_Gallon) * Price_Per_Gallon + *d * Bike_Per_Mile +
(*d / Min_Miles_Daily) * Food_cost;
}
```

Test Case



```
Please input your distance in miles:3867
The number of gallons of fuel required is: 70.31 gallons.
The total cost of the trip is from $4224.36 to $4649.36.
```

- Here, I reserve two decimals to keep the outcomes clear to users.

Characteristics and Side Effects

- In the program, I use the pointer to return two values to the main() function, and to keep the robustness of the program
- Besides, I utilize the symbolic constants instead of magic numbers to make the program more readable.
- As for the side effect, distance has to be an integer when it is input from the user, and all outcomes are based on assumptions made in the description, which indicates that in real world there maybe some deviations.

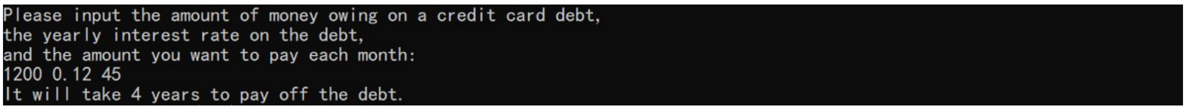
Problem 4

Program 4

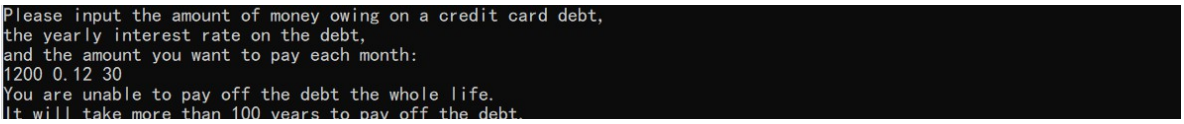
```
/*----------------------------------------
Module name: program_4.c
Description: Write a program that prompts you for the amount of money owing on a credit card debt,
the yearly interest rate on the debt, and the amount you want to pay each month. Write a
function that then computes the number of years it will take to pay off the debt given the
information provided.
Author: Bo Yue
Rev. 0 03 Aug 2019
----------------------------------------*/

#include <stdio.h>
#include <math.h>

//declaration of function years_pay_debt()
```

```
int years_pay_debt(double* a, double* b, double* c);

int main()
{
    double money_owe            = 0.0;
    double yearly_interest_rate = 0.0;
    double monthly_amount       = 0.0;
    double monthly_inc_debt     = 1.0;

    //make sure the debtor can pay off the debt per month
    while(monthly_amount <= monthly_inc_debt)
    {
        printf("Please input the amount of money owing on a credit card
debt,\nthe yearly interest rate on the debt,\nand the amount you want to pay
each month:\n");
        scanf("%lf %lf %lf", &money_owe, &yearly_interest_rate,
&monthly_amount);
        monthly_inc_debt     = money_owe * yearly_interest_rate / 12;
    }

    years_pay_debt(&money_owe, &yearly_interest_rate, &monthly_amount);

    return 0;
}

//definition of function years_pay_debt()
int years_pay_debt(double* money_owe, double* yearly_interest_rate, double*
monthly_amount)
{
    int i = 1;
    while(pow((1+(*yearly_interest_rate)),i) * (*money_owe) >
((*monthly_amount) * 12 * i ))

        i++;
        if(i >= 100)
        {
            printf("You are unable to pay off the debt the whole life.\n");
            printf("It will take more than %d years to pay off the debt.",
i);
            break;
        }
    }
    if(i < 100)
    {
        printf("It will take %d years to pay off the debt.", i);
    }

    return i;
}
```

Test Case



```
Please input the amount of money owing on a credit card debt,
the yearly interest rate on the debt,
and the amount you want to pay each month:
1200 0.12 45
It will take 4 years to pay off the debt.
```



```
Please input the amount of money owing on a credit card debt,
the yearly interest rate on the debt,
and the amount you want to pay each month:
1200 0.12 30
You are unable to pay off the debt the whole life.
It will take more than 100 years to pay off the debt.
```

- In the input(scanf()) session, I only take into account that the amount the user pay each month is bigger than the monthly interest. However, this does **not** necessarily means the debtor can pay off his or her debt in limited years, because it is calculated in compound interest. Thus, the base will be larger as time passes by. Therefore, probably, there will be no finite answer for the question.
- The second case solves the problem in some way, as a natural person normally lives a life about a hundred years. If he or she cannot pay off his or her debt in a hundred years' time, then it is reasonable to assume that he or she cannot pay off his or her debt in his or her lifetime.

Characteristics and Side Effects

- I use the pointer in this function to realize pass by reference.
- Side effect is that the outcome may not be an finite answer.

Problem 5

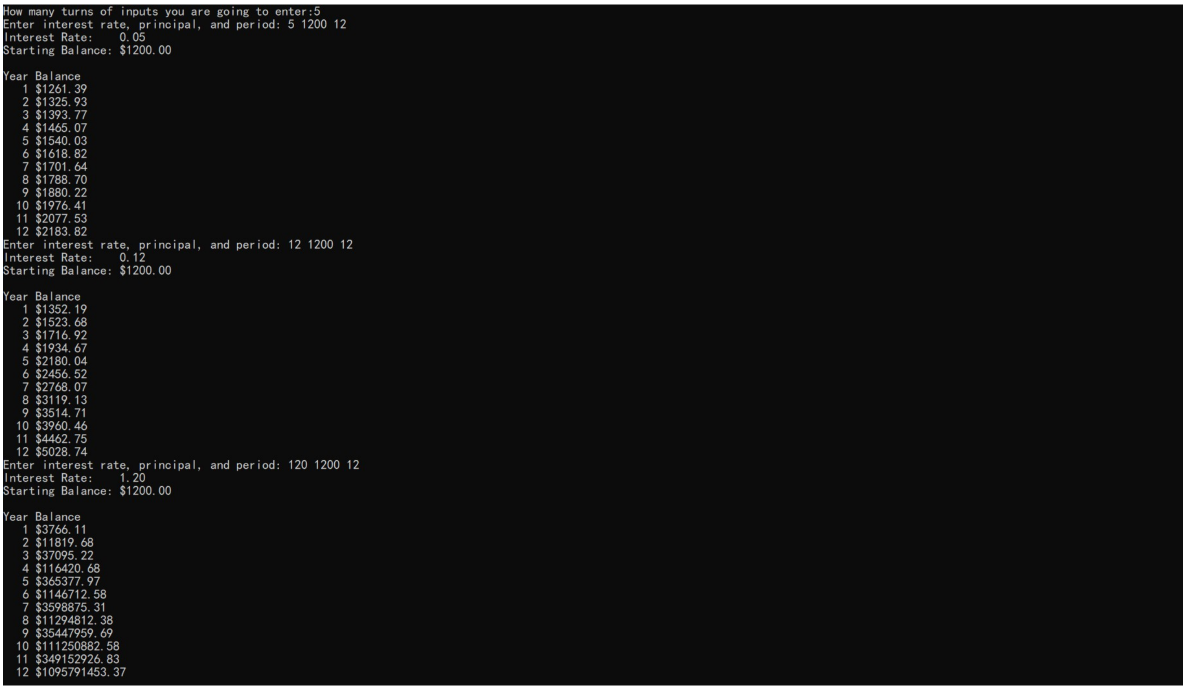
Program 5

```
/*----------------------------------------
Module name: program_5.c
Description: Modify the following program to allow more than one interest
rate, period, and balance to be entered.
```

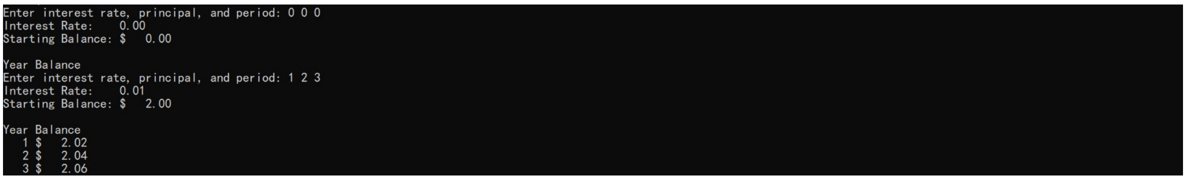
```
#include <stdio.h>
// function prototypes
void displayValues(double yrlyPct, double startBal);
double yearEndBalance(double intrate, double monthlyBal);
int main(void)
{
// working variables
int period = 0; // length of period in years
int year = 0; // year of period
double balance = 0; // balance at end of year
double intrate = 0; // interest rate
// prompt the user and get inputs
printf("Enter interest rate, principal, and period: ");
scanf("%lf %lf %i", &intrate, &balance, &period);
// convert percent to fraction
intrate = intrate/100;
// print out values entered
displayValues(intrate, balance);
// print heading
printf("Year Balance\n");
// compute and print yearly balance
for (year = 1; year <= period; year = year + 1)
{
balance = yearEndBalance (intrate, balance);
printf("%4i $%7.2f\n", year, balance);
}
return 0; // assume program succeeded
}
// display user information
void displayValues (double yrlyPct, double startBal)
{
printf("Interest Rate: %7.2f%\n", yrlyPct);
printf("Starting Balance: $%7.2f\n\n", startBal);
}
// compute yearly balance
double yearEndBalance (double intrate, double monthlyBal)
{
// working variables
int month = 0;
double monthlyIntrate = 0;
// convert annual to monthly interest rate
monthlyIntrate = intrate / 12;
// compute year end balance
```

```
for (month = 0; month < 12; month = month + 1)
{
monthlyBal = monthlyBal * monthlyIntrate + monthlyBal;
}
return monthlyBal;
}

Author: Bo Yue
Rev. 0 03 Aug 2019
----------------------------------------*/

#include <stdio.h>
// function prototypes
void displayValues(double yrlyPct, double startBal);
double yearEndBalance(double intrate, double monthlyBal);
void loop(int cnt, int turn);

int main(void)
{
    int cnt  = 0;
    int turn = 0;
    printf("How many turns of inputs you are going to enter:");
    scanf("%d", &turn);

    //loop for cnt times
    loop(cnt, turn);

    return 0; // assume program succeeded
}

// display user information
void displayValues (double yrlyPct, double startBal)
{
    printf("Interest Rate: %7.2f\n", yrlyPct);
    fflush(stdin);
    printf("Starting Balance: $%7.2f\n\n", startBal);
}

// compute yearly balance
double yearEndBalance (double intrate, double monthlyBal)
{
// working variables
    int month = 0;
    double monthlyIntrate = 0;
    // convert annual to monthly interest rate
    monthlyIntrate = intrate / 12;
    // compute year end balance
    for (month = 0; month < 12; month = month + 1)
    {
        monthlyBal = monthlyBal * monthlyIntrate + monthlyBal;
    }
    return monthlyBal;
}

void loop(int cnt, int turn)
{
     for (; cnt < turn; cnt++)
    {
        // working variables
        int period = 0; // length of period in years
        int year = 0; // year of period
        double balance = 0; // balance at end of year
        double intrate = 0; // interest rate
        // prompt the user and get inputs
        //fflush();
        printf("Enter interest rate, principal, and period: ");
        scanf("%lf %lf %i", &intrate, &balance, &period);
        // convert percent to fraction
        intrate = intrate/100;
        // print out values entered
        displayValues(intrate, balance);
        // print heading
        printf("Year Balance\n");
        // compute and print yearly balance
        for (year = 1; year <= period; year = year + 1)
        {
            balance = yearEndBalance (intrate, balance);
            printf("%4i $%7.2f\n", year, balance);
        }
    }
    return;
}
```

Test Case



```
How many turns of inputs you are going to enter:5
Enter interest rate, principal, and period: 5 1200 12
Interest Rate:    0.05
Starting Balance: $1200.00

Year Balance
   1 $1261.39
   2 $1325.93
   3 $1393.77
   4 $1465.07
   5 $1540.03
   6 $1618.82
   7 $1701.64
   8 $1788.70
   9 $1880.22
  10 $1976.41
  11 $2077.53
  12 $2183.82
Enter interest rate, principal, and period: 12 1200 12
Interest Rate:    0.12
Starting Balance: $1200.00

Year Balance
   1 $1352.19
   2 $1523.68
   3 $1716.92
   4 $1934.67
   5 $2180.04
   6 $2456.52
   7 $2768.07
   8 $3119.13
   9 $3514.71
  10 $3960.46
  11 $4462.75
  12 $5028.74
Enter interest rate, principal, and period: 120 1200 12
Interest Rate:    1.20
Starting Balance: $1200.00

Year Balance
   1 $3766.11
   2 $11819.68
   3 $37095.22
   4 $116420.68
   5 $365377.97
   6 $1146712.58
   7 $3598875.31
   8 $11294812.38
   9 $35447959.69
  10 $111250882.58
  11 $349152926.83
  12 $1095791453.37
```



```
Enter interest rate, principal, and period: 0 0 0
Interest Rate:    0.00
Starting Balance: $   0.00

Year Balance
Enter interest rate, principal, and period: 1 2 3
Interest Rate:    0.01
Starting Balance: $   2.00

Year Balance
   1 $   2.02
   2 $   2.04
   3 $   2.06
```

- First, the user enters how many turns the user will enter. Then the user can enter the interest rate, balance and period for each turn.

Characteristics and Side Effects

- I use a function loop() to eliminate prolix main() function body.
- I use pass by value instead of pass by reference.
- The outcomes may not be in ordered align, as the format of output is determined in the function beforehand.