

# Homework 2

---

## Homework 2

### Problem 1

Program 1

Test Case 1

Test Case 2

### Problem 2

Program 2

Test case

### Problem 3

Program 3

Test Case

### Problem 4

Program 4

Test Case

### Problem 5

Program 5

Test Case 1

Test Case 2

Test Case 3

### Problem 6

Test Case

### Problem 7

Test Case 1

Test Case 2

Test Case 3

### Problem 8

Test Case 1

Test Case 2

### Statement

Homework 2 is mainly concerned with *the **standard** input and output* of C programs. There are *eight* major problems to deal with, which will be discussed in detail as follows:

## Problem 1

---

- Problem 1 has a lot to do with input and output of ASCII.

## Program 1

```
/*-----  
Module name: program_1.c  
Description: The below is a simple C program that prompts for an ASCII  
code value such as 65 (see  
asciitable.com) then prints the character having that code.
```

Author: Bo Yue

Rev. 0 16 Jul 2019

-----\*/

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

```
# define ASC_VALUE 128                                //use a symbolic value for
asc_value
```

```
int main()
```

```
{
```

```
    int asc_value = ASC_VALUE;                          //asc_value stores an ASCII
code value
```

```
    while(asc_value<0 || asc_value>127)                //the input value should
within proper range
```

```
    {
```

```
        printf("Please input your ASCII value:");
```

```
        scanf("%d", &asc_value);
```

```
    }
```

```
    printf("%c\n", (char)asc_value);                    //type conversion from int to
char
```

```
    return 0;
```

```
}
```

## Test Case 1



- This matches the ASCII table on [www.asciitable.com](http://www.asciitable.com).

## Test Case 2

```
Please input your ASCII value:290
Please input your ASCII value:-1
Please input your ASCII value:107
k
Process returned 0 (0x0)   execution time : 11.551 s
Press any key to continue.
```

- In Test Case 2, I input two values "290" and "-1", which are out of the proper ASCII range(0-127, included both).
- However, thanks to my program's robustness, no error occurs.
- And the program loops until a proper value is entered.

## Problem 2

- Problem 2 is concerned with formatted *float-number* output for decimal and exponential notation.

## Program 2

```
/*-----
```

```
Module name: program_2.c
```

Description: This is a program that prompts for a floating-point number then prints the number first in decimal point notation then in exponential notation. Present the output in the following

format: The input is 98.600 or 9.8600 e+001

Author: Bo Yue

Rev. 0 16 Jul 2019

-----\*/

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

```
int main()
```

```
{
```

```
    float k;
```

```
    scanf("%f", &k);
```

```
    printf("The input is %.3f or %.4e", k, k); // the decimal notation and  
the exponential notation
```

```
    return 0;
```

```
}
```

## Test case

```
98.6  
The input is 98.600 or 9.8600e+001  
Process returned 0 (0x0)   execution time : 1.781 s  
Press any key to continue.
```

- The Test Case goes smoothly.

## Problem 3

- Problem 3 is concerned with the scientific notation.

## Program 3

```
/*-----*/
```

Module name: program\_3.c

Description: There are approximately  $3.156 \times 10^7$  seconds in a year. This is a program that requests

the user's age in years and then displays the equivalent value in seconds.

The program prints the

answer in an informative way.

Author: Bo Yue

Rev .0 17 Jul 2019

-----\*/

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

```
# define SEC_PER_YEAR 3.156e+007
```

```
//define the constant
```

```
int main()
```

```
{
```

```

int age          = 0;
double total_sec = 0;
printf("Please input your age in years:");
scanf("%d", &age);
total_sec        = age * SEC_PER_YEAR;
printf("Your age equivalent value in second is: %.3es", total_sec);
return 0;
}

```

## Test Case

```

Please input your age in years:6
Your age equivalent value in second is: 1.894e+008s
Process returned 0 (0x0)   execution time : 1.237 s
Press any key to continue.

```

- Here, I assume the variable "age" a integer. It can also be a float number, according to different circumstances.

## Problem 4

- Problem 4 is also concerned with the scientific notation.

## Program 4

```

/*-----
Module name: program_4.c
Description: The mass of a single molecule of water is about 3.0 x 10-23
grams. A quart of water is
about 950 grams. Write a program that requests an amount of water, in
quarts, and
displays the number of water molecules in that amount. Think about an
informative way
to print the answer.
Author: Bo Yue
Rev .0 18 Jul 2019
-----*/

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

#define GRAM_PER_MOLECULE 3 * pow(10, -23)
#define GRAM_PER_QUART 950
int main()
{
    int amount          = 0;
    float num_of_molecules = 0.0;

    //input session
    printf("Please input the amount of water in quarts: ");
    scanf("%d", &amount);

```

```

    //calculation session
    //there is a bracket outside GRAM_PER_MOLECULE, because 3 and pow(10,
-23) are together
    num_of_molecules      = (GRAM_PER_QUART * amount) /
(GRAM_PER_MOLECULE);

    //format output
    printf("%.2e", num_of_molecules);
    return 0;
}

```

## Test Case

```

Please input the amount of water in quarts: 6
1.90e+026
Process returned 0 (0x0)   execution time : 1.759 s
Press any key to continue.

```

- As long as the input value for "amount" is within proper range, there will not occur any problem.

## Problem 5

- Problem 5 is concerned with if & else statement and standard input & output.

## Program 5

```

/*-----
Module name: program_5.c
Description: This is a program that prompts for a temperature value, the
scale, Celsius, Fahrenheit, or
Kelvin and the scale to which that temperature is to be converted, and
display the result in
the requested scale. The answer is printed in an informative manner.
Author: Bo Yue
Rev .0 18 Jul 2019
-----*/

#include <stdio.h>

int main()
{
    float temperature = 0.0;
    char scale_last   = '0';
    char scale_next   = '0';

    printf("Please input a temperature value: ");
    scanf("%f", &temperature);
    getchar();
    printf("C for Celsius, F for Fahrenheit, and K for kelvin.\n");
}

```

```

printf("Please input the present scale for temperature: ");
scanf("%c", &scale_last);
getchar();
printf("Please input the next scale for temperature: ");
scanf("%c", &scale_next);

if('C' == scale_last)
{
    if('F' == scale_next)
    {
        temperature = 1.8 * temperature + 32;           //Celsius to
Fahrenheit
    }
    else if('K' == scale_next)
    {
        temperature = temperature + 273.15;           //Celsius to
Kelvin
    }
}
else if('F' == scale_last)
{
    if('C' == scale_next)
    {
        temperature = (temperature - 32) * 5 / 9;       //Fahrenheit
to Celsius
    }
    else if('K' == scale_next)
    {
        temperature = (temperature - 32) * 5 / 9 + 273.15; //Fahrenheit
to Kelvin
    }
}
else if('K' == scale_last)
{
    if('C' == scale_next)
    {
        temperature = temperature - 273.15;           //kelvin to
Celsius
    }
    else if('F' == scale_next)
    {
        temperature = (temperature - 273.15) * 9 / 5 + 32; //kelvin to
Fahrenheit
    }
}

if('C' == scale_next) {printf("The converted temperature is: %.2f°C",
temperature);}
else if('F' == scale_next) {printf("The converted temperature is:
%.2f°F", temperature);}
else if('K' == scale_next) {printf("The converted temperature is:
%.2fK", temperature);}

```

```
    return 0;
}
```

## Test Case 1

```
Please input a temperature value: 12
C for Celsius, F for Fahrenheit, and K for Kelvin.
Please input the present scale for temperature: C
Please input the next scale for temperature: F
The converted temperature is: 53.60°F
```

## Test Case 2

```
Please input a temperature value: 53.6
C for Celsius, F for Fahrenheit, and K for Kelvin.
Please input the present scale for temperature: F
Please input the next scale for temperature: K
The converted temperature is: 285.15K
```

## Test Case 3

```
Please input a temperature value: 300.15
C for Celsius, F for Fahrenheit, and K for Kelvin.
Please input the present scale for temperature: K
Please input the next scale for temperature: C
The converted temperature is: 27.00°C
```

- Here, we assume that all input is within reasonable range. For instance, for Kelvin, it is always a non-negative value input.

## Problem 6

- Problem 6 is concerned about standard input, calculation, and standard output.

```
/*-----
Module name: program_6.c
Description: This is a C program that indicates the delay from the point
where the signal enters the path until it leaves. Assumptions:
The delay of a signal propagating along a printed circuit board trace is
180 picoseconds (10-12 seconds) per inch. The delay through a
logic device is approximately 5 nanoseconds (10-9 seconds). The logic
devices are placed on the circuit board with a printed circuit
board trace of 0.1 inches connecting the output of one device to the input
of the next.
The program prompts the user for the number of logic devices in a signal
path and displays the total delay along the path.
The answer is printed in an informative way.
Author: Bo Yue
Rev .0 19 Jul 2019
-----*/

#include <stdio.h>

//all the unit for time is picoseconds
#define DELAY_PER_INCH 180
#define DELAY_PER_DEVICE 5000
//all units for length is inch
```

```
#define LENGTH_PER_CONNECTION 0.1

int main()
{
    int num_of_device;
    int total_delay;
    printf("Please input the number of logic devices:");
    scanf("%d", &num_of_device);

    //calculate total delay based on assumption
    total_delay = num_of_device * DELAY_PER_DEVICE + (num_of_device - 1) *
    LENGTH_PER_CONNECTION * DELAY_PER_INCH;

    printf("The total delay along the path is: %d picoseconds",
    total_delay);
    return 0;
}
```

## Test Case

```
Please input the number of logic devices:13
The total delay along the path is: 65216 picoseconds
```

- The Test Case result is correct and proper.

## Problem 7

- Problem 7 is concerned about fraction output and reduction of a fraction.

```
/*-----
Module name: program_7.c
Description: You have the following recipe for barbeque sauce:
1 tablespoon of fat
1/2 cup of water
2 tablespoons vinegar
1 tablespoon worcestershire sauce
1/4 cup lemon juice
2 tablespoons brown sugar
1 cup chili sauce
1/2 teaspoon salt
1/4 teaspoon paprika
1 teaspoon pepper
1 teaspoon dry mustard
Sometimes, the user want to make a lot and sometimes a little. This is a
program that prompts for the amount
you want (1, 2, 3; times the single recipe) and displays the amount of
each ingredient for the size of the
batch that you want. The answer is printed in an informative way.
Author: Bo Yue
Rev .0 19 Jul 2019
-----*/
```



```

#include <stdio.h>

int main()
{
    //for input
    int amount = 0;
    //for output
    int amount_of_each = 0;
    printf("Please input your amount:");
    scanf("%d", &amount);

    printf("You have the following recipe for barbeque sauce:\n");

    //tablespoon of fat
    amount_of_each = amount;
    printf("%d tablespoon of fat\n", amount_of_each);

    //cup of water
    if(0 == amount % 2)
    {
        amount_of_each = amount / 2;
        printf("%d cup of water\n", amount_of_each);
    }
    else
    {
        printf("%d/2 cup of water\n", amount_of_each);
    }

    //tablespoons vinegar
    amount_of_each = 2 * amount;
    printf("%d tablespoons vinegar\n", amount_of_each);

    //tablespoon worcestershire sauce
    amount_of_each = amount;
    printf("%d tablespoon worcestershire sauce\n", amount_of_each);

    //cup lemon juice
    amount_of_each = amount;
    if(1 == amount % 2)
    {
        printf("%d/4 cup lemon juice\n", amount_of_each);
    }
    else if(2 == amount % 4)
    {
        amount_of_each = amount / 2;
        printf("%d/2 cup lemon juice\n", amount_of_each);
    }
    else
    {
        amount_of_each = amount / 4;
        printf("%d cup lemon juice\n", amount_of_each);
    }
}

```

```

}

//tablespoons brown sugar
amount_of_each      = 2 * amount;
printf("%d tablespoons brown sugar\n", amount_of_each);

//cup chili sauce
amount_of_each      = amount;
printf("%d cup chili sauce\n", amount_of_each);

//teaspoon salt
if(0 == amount % 2)
{
    amount_of_each = amount / 2;
    printf("%d teaspoon salt\n", amount_of_each);
}
else
{
    printf("%d/2 teaspoon salt\n", amount_of_each);
}

//teaspoon paprika
if(1 == amount % 2)
{
    printf("%d/4 teaspoon paprika\n", amount_of_each);
}
else if(2 == amount % 4)
{
    amount_of_each = amount / 2;
    printf("%d/2 teaspoon paprika\n", amount_of_each);
}
else
{
    amount_of_each = amount / 4;
    printf("%d teaspoon paprika\n", amount_of_each);
}

//teaspoon pepper
amount_of_each      = amount;
printf("%d teaspoon pepper\n", amount_of_each);

//teaspoon dry mustard
amount_of_each      = amount;
printf("%d teaspoon dry mustard\n", amount_of_each);

return 0;
}

```

## Test Case 1

```
Please input your amount:3
You have the following recipe for barbeque sauce:
3 tablespoon of fat
3/2 cup of water
6 tablespoons vinegar
3 tablespoon Worcestershire sauce
3/4 cup lemon juice
6 tablespoons brown sugar
3 cup chili sauce
3/2 teaspoon salt
3/4 teaspoon paprika
3 teaspoon pepper
3 teaspoon dry mustard
```

- This is a simple input to verify that the program is correct.

## Test Case 2

```
Please input your amount:10
You have the following recipe for barbeque sauce:
10 tablespoon of fat
5 cup of water
20 tablespoons vinegar
10 tablespoon Worcestershire sauce
5/2 cup lemon juice
20 tablespoons brown sugar
10 cup chili sauce
5 teaspoon salt
5/2 teaspoon paprika
10 teaspoon pepper
10 teaspoon dry mustard
```

- This Test Case 2 is concerned with reduction of a fraction for a factor of **2**.

## Test Case 3

```
Please input your amount:20
You have the following recipe for barbeque sauce:
20 tablespoon of fat
10 cup of water
40 tablespoons vinegar
20 tablespoon Worcestershire sauce
5 cup lemon juice
40 tablespoons brown sugar
20 cup chili sauce
10 teaspoon salt
5 teaspoon paprika
20 teaspoon pepper
20 teaspoon dry mustard
```

- This Test Case 3 is concerned with reduction of a fraction for a factor of **4**.

## Problem 8

- Problem 8 is concerned with exact division.

```
/*-----
Module name: program_8.c
Description: You want to tile a floor with 12 inch tiles (12" x 12").
Tiles come 24 to the box. Write a
program that asks for the length and width of the room and returns the
number of boxes
of tiles needed to complete the job. Think about an informative way to
print the answer.
Author: Bo Yue
Rev .0 19 Jul 2019
-----*/

#include <stdio.h>
```

```

#define tile_side_length 12

int main()
{
    int length          = 0;
    int width           = 0;
    int num_for_length  = 0;
    int num_for_width   = 0;
    int num_of_box      = 0;

    printf("Please input the length and width of th room, respectively:");
    scanf("%d%d", &length, &width);

    if(0 == length % 12)
    {
        num_for_length    = length / 12;
    }
    else
    {
        num_for_length    = length / 12 + 1;//add one if more piles are
needed
    }

    if(0 == width % 12)
    {
        num_for_width     = width / 12;
    }
    else
    {
        num_for_width     = width / 12 + 1;
    }

    num_of_box            = num_for_length * num_for_width / 24;
    if(0 != num_of_box % 24)
    {
        num_of_box        = num_for_length * num_for_width / 24 + 1;//add
one if more boxes are needed
    }

    printf("The number of box is: %d", num_of_box);
    return 0;
}

```

## Test Case 1

```

Please input the length and width of th room, respectively:120 120
The number of box is: 5

```

## Test Case 2

```

Please input the length and width of th room, respectively:100 130
The number of box is: 5

```

- It can be clearly seen that exact division is executed.

## Statement

---

- I acknowledge that I code all the program.
- Signature:

*Bo Yue*