

Programs in C and documentation

by Xuyuan Liu

FILE	12055_XUYUAN_LIU_PROGRAMS_IN_C_AND_DOCUMENTATION_157877_1607012546.DOCX (218.89K)		
TIME SUBMITTED	24-FEB-2019 10:34 PM (UTC+0000)	WORD COUNT	1554
SUBMISSION ID	101123672	CHARACTER COUNT	9393

Programs in C and documentation
– Part A and Part B

Software Development

Table of Contents

Part A..... 4

Part B..... 6

 Exercise 1.....3

 Exercise 2.....4

 Exercise 3.....7

 Exercise 4.....9

List of Figure

1. flowchart of exercise 1	3
2. Screenshot of exercise 1	4
3. flowchart of exercise 2	5
4. Screenshot of exercise 2	7
5. flowchart of exercise 3	7
6. Screenshot of exercise 3	8
7. flowchart of exercise 4	10
8. Screenshot of exercise 4 cumulative score version	14
9. Screenshot of exercise 4 non-cumulative score version.....	15

Programs in C and documentation

By Xuyuan Liu

Part A

A1. What is a variable and why are they important in writing programs?

Variable is a important concept in computer programming. A variable is a letter or name that can store a value. When you program, you can use variables to store numbers, such as the radius of a circle, or store words, such as a person's name. Simply put, variables can be used to represent any information the program needs.

I think the reason the variable is important is because the variable can change its value as the program runs. Constant memory data can only store a fixed amount of data, which will never change, so this is certainly not a memorable data. The data stored in the variable can be changed, you can make it remember 5, and the next step it can become 10. You can use a variable to represent different values without having to re-modify the program each time the value changes. This is indispensable in programming.

A2. Explain why the for-loop can also be referred to as a counting-loop and give an example of when you would use it.

The for loop is an open loop statement in the programming language, and the loop statement consists of the loop body and the termination condition of the loop.

The loop in C language is as follows

For (; ;)

The general form is:

For (single expression; conditional expression; end loop body)

{

Intermediate circulation body;

}

The first ";" in the for loop parentheses is a single expression that does not participate in the loop. It can be used as an initial assignment statement for a variable to give initial values to the loop control variable. It can also be used to calculate Other expressions that are independent of the loop but are processed before the loop part.

The conditional expression between the two ";" signs is a relational expression, which is the official beginning of the loop, and executes the intermediate loop body when the conditional expression is true.

The intermediate loop body that is executed can be a single statement or multiple statements. After executing the intermediate loop body, the loop body is executed.

After the end of the loop body is executed, the condition is judged again. If the condition is still true, the above loop is continued, and when the condition is not satisfied, the current loop is skipped. So the for loop can be used as a counting loop.

For example:

To calculate the sum of 1 to 100. From $i = 1$, add 1 to each iteration till $i = 101$, jump out of the loop. This completes the count from 1 to 100.

```
for(i=1,sum=0; i<=100; i++) {  
    sum = sum + i;  
}
```

A3. Explain what an array is, what their features are, and give an example of when you would use one.

About arrays we can think of it as a collection of all the data of a type, and use an array subscript to distinguish or specify each number.

For example, a football team usually has dozens of people, but when we come to know them, we first regard them as members of a certain pair. Then use their number to distinguish each player. At this time, the team is an array, and the number is the number in the array. When we indicated that it was a few players, we found this player.

Also in programming, if we have a set of data of the same data type. For example, there are 10 numbers. If we want to use variables to store them, we need to use 10 variables separately, and remember the names of the 10 variables. This will be very troublesome. At this point we can use an array variable to store them.

For example:

The user wants to sum of 10 numbers. There need to use the array. According to the 10 numbers entered by the user and saved to an array.

```
int num[10];  
  
int i;  
  
for(i=0;i<10;i++)  
{  
    printf("Please enter the number: ");  
    scanf("%f", &num[i]);  
}
```

Part B

Exercise 1:

This code is very simple. First declare variables with char, int, float. And then assign values to variables. Finally use printf code display these three values on the screen.

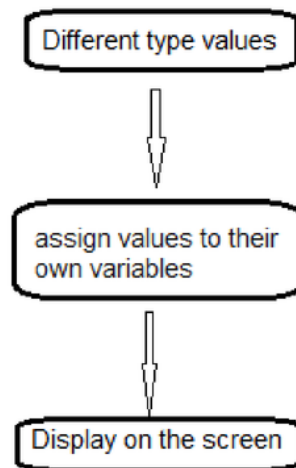
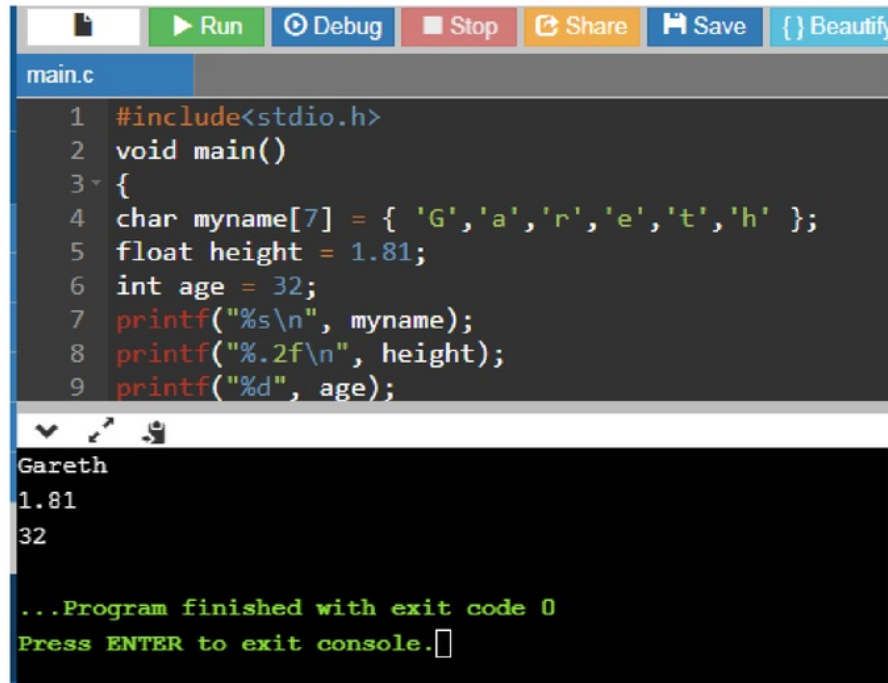


Figure 1. flowchart of exercise 1

```
#include<stdio.h>
void main()
{
    char myname[7] = { 'G','a','r','e','t','h' };// Declare strings
    float height = 1.81;// declare float
    int age = 32;// declare integers
    printf("%s\n", myname);// %d for strings
    printf("%.2f\n", height);// %d for float
    printf("%d", age);// %d for integers
}
```

A screenshot of a code editor window showing a C program named 'main.c'. The code defines a main function that declares a character array 'myname' with the value 'Gareth', a float 'height' with the value 1.81, and an integer 'age' with the value 32. It then uses printf to output these values on separate lines. The console output shows 'Gareth', '1.81', and '32'. The program finishes with exit code 0, and a message prompts the user to press ENTER to exit the console.

```
1 #include<stdio.h>
2 void main()
3 {
4     char myname[7] = { 'G','a','r','e','t','h' };
5     float height = 1.81;
6     int age = 32;
7     printf("%s\n", myname);
8     printf("%.2f\n", height);
9     printf("%d", age);
10 }

Gareth
1.81
32

...Program finished with exit code 0
Press ENTER to exit console.
```

Figure 2. Screenshot of exercise 1

Exercise 2:

In order to find the area of the shadow part, first ask for the area of the circle and the area of the triangle.

For convenience and clarity, use two separate functions, float circlearea(float radius) and float triangulararea(float b, float h), to find the area of the circle and triangle, and use the enterfloat function to enter the value.

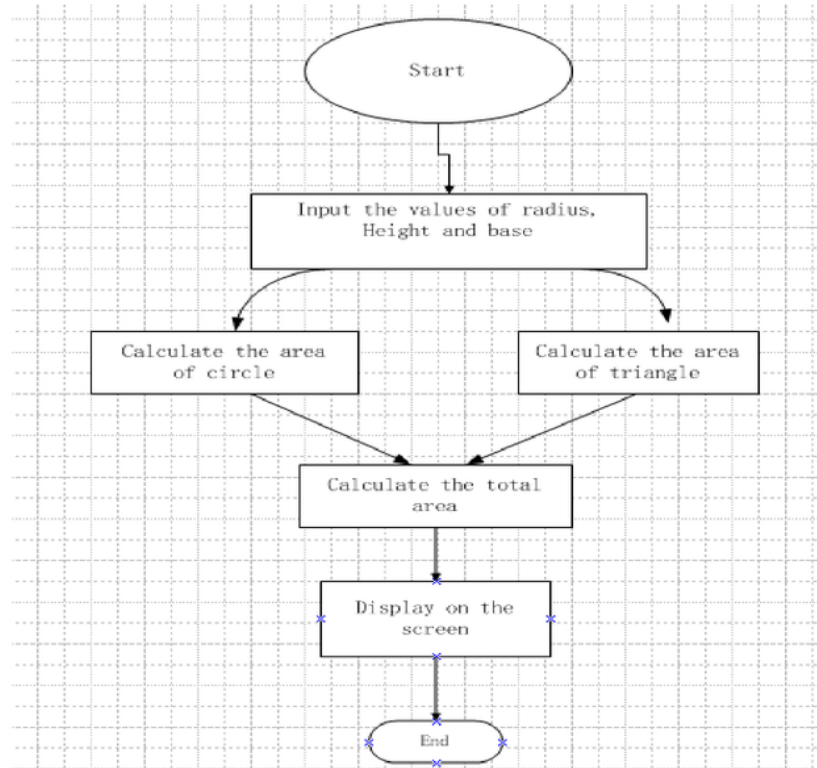


Figure 3. flowchart of exercise 2

```
#include <stdio.h>
#define PI 3.142
float enterfloat();
float circlearea(float radius);
float triangulararea(float b, float h);
// Declare functions
void main()
{
    float value1,value2,result1,result2,total;
    printf("Please enter your radius value:");
    value1 = enterfloat();//call function
    result1 = circlearea(value1); //call function
    printf("Please enter your base value:");
```

```
value1 = enterfloat();//call function
printf("Please enter your height value:");
value2 = enterfloat();//call function
result2 = triangulararea(value1, value2); //call function
total = result1-result2;
printf("The total area of the shape is: %.2f", total);//output the result
}
```

```
float enterfloat()
{
float decvalue;
scanf("%f", &decvalue);
return decvalue;
}
//input decimal values
```

```
float circlearea(float radius)
{
float area;
area = radius*radius*PI;
return area;
}
//calculate the area of circle
```

```
float triangulararea(float b,float h)
{
float area;
area = b*h/2;
return area;
}
//calculate the area of triangle
```

```
Please enter your radius value:2
Please enter your base value:2
Please enter your height value:1
The total area of the shape is: 11.57

...Program finished with exit code 37
Press ENTER to exit console.□
```

Figure 4. Screenshot of exercise 2

Exercise 3:

This program requires two for loops to make the program the most concise. A for loop is used to enter 5 values. Another for loop is used for summation.

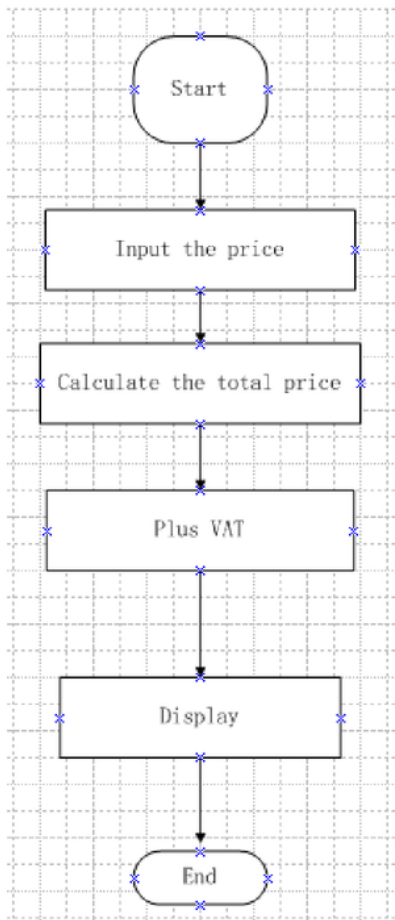


Figure 5. flowchart of exercise 3

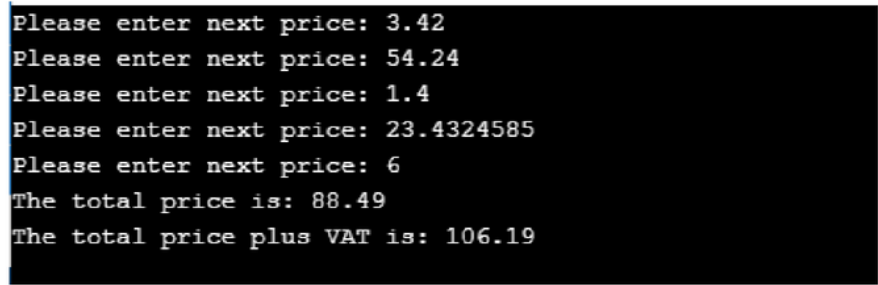
```
#include <stdio.h>

void main()
{
    float prices[5]; //set up array
    int i;
    float total = 0;
    float VAT;

    for (i = 0; i < 5; i++) //Allow the user to enter prices
    {
        printf("Please enter next price: ");
        scanf("%f", &prices[i]);
    }

    for (i = 0; i < 5; i++) //Calculate the total price
    {
        total = total + prices[i];
    }

    VAT = total + 0.2*total;
    printf("The total price is: %.2f\n", total);
    printf("The total price plus VAT is: %.2f", VAT);
}
```

A screenshot of a terminal window showing the output of the C program. The text is as follows:

```
Please enter next price: 3.42
Please enter next price: 54.24
Please enter next price: 1.4
Please enter next price: 23.4324585
Please enter next price: 6
The total price is: 88.49
The total price plus VAT is: 106.19
```

Figure 6. Screenshot of exercise 3

Exercise 4:

This program is more complicated. First use an if function to determine if you need to accumulate the scores.

Then use the while loop to keep the game can come back. Use the srand function to generate a random number. The random number % 5 and plus one to get a random number from 1 to 5. Judging the score.

And whether spin again.

End the game.

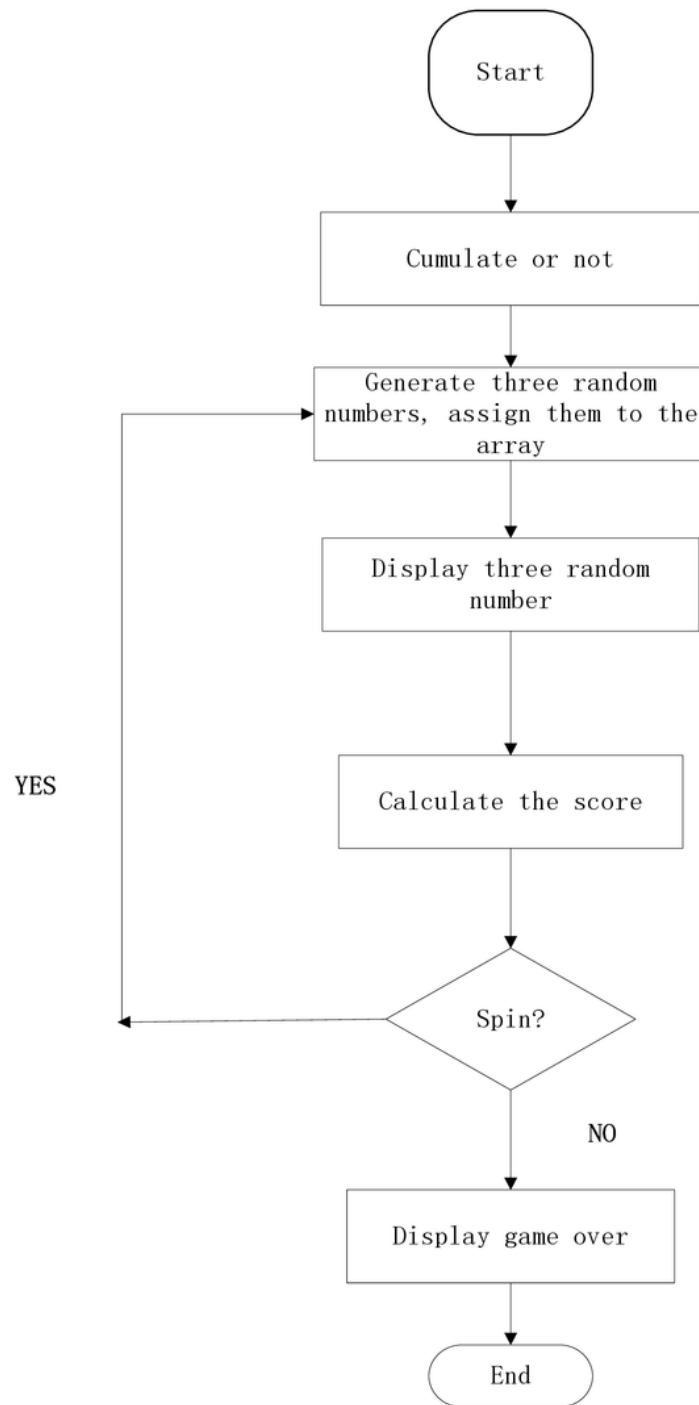


Figure 7. flowchart of exercise 4

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

char isCumulative = 'Y';// Whether to accumulate points
int points = 0;
int calculationPoint(int point) {
    if(isCumulative == 'Y' || isCumulative == 'y')
    {
        points = points + point;
    }
    else
    {
        points = point;
    }
    return points;
}

int main()
{
    char isSpin;
    char c_tmp;

    printf("please choose whether use cumulative score Y/N?");
    scanf("%c",&isCumulative);

    while(1)
    {
        // Generate three random numbers [1, 5] into the array fruits[3]
        int fruits[3];
        int i ;
        srand(time(NULL));// Generate random number
        for (i = 0; i < 3; i++)
```

```
{
    fruits[i] = rand()%5 + 1; // Remainder plus 1
}
printf("%d %d %d \n", fruits[0],fruits[1],fruits[2]); //Display these three numbers
//Scoring
//When three number is same
if(fruits[0]==fruits[1] && fruits[1]==fruits[2])
{
    //When three numbers are all 5
    if(fruits[0] == 5) {
        points = calculationPoint(2000);
        printf("You win, and your points are %d\n",points);
    }
    //Three same number but do not have 5
    else
    {
        points = calculationPoint(20);
        printf("You win, and your points are %d\n",points);
    }
}

//Two of 5
else if((fruits[0]==fruits[1] && fruits[0] == 5) ||(fruits[0]==fruits[2] && fruits[0] ==
5)||fruits[1]==fruits[2] && fruits[1] == 5))
{
    points = calculationPoint(100);
    printf("You win, and your points are %d\n",points);
}

//Other cases
else {
    points = calculationPoint(0);
    printf("You lose, and your points are %d\n",points);
}
```

```
printf("Spin?"); //Continue or not
while((c_tmp = getchar() != '\n') && c_tmp != EOF); //empty loop, empty the
garbage in the cache and other garbage characters
scanf("%c",&isSpin); // Enter characters from the keyboard again
if(isSpin == 'y') // Convert the input lowercase y to Y
{
    isSpin = 'Y';
}
// If it is not Y, jump out of the while loop and end the game
if(isSpin != 'Y')
{
    printf("Game Over! Think you!");
    break;
}
}
}
```

```
please choose whether use cumulative score Y/N?y
2 4 3
You lose, and your points are 0
Spin?y
1 4 4
You lose, and your points are 0
Spin?y
2 3 4
You lose, and your points are 0
Spin?y
5 2 4
You lose, and your points are 0
Spin?y
5 4 5
You win, and your points are 100
Spin?y
5 4 1
You lose, and your points are 100
Spin?y
3 4 2
You lose, and your points are 100
Spin?y
1 4 5
You lose, and your points are 100
Spin?y
5 3 5
You win, and your points are 200
Spin?y
1 5 4
You lose, and your points are 200
```

Figure 8. Screenshot of exercise 4 cumulative score version

```
please choose whether use cumulative score Y/N?n
5 5 3
You win, and your points are 100
Spin?y
2 5 3
You lose, and your points are 0
Spin?y
4 4 4
You win, and your points are 20
Spin?y
4 3 5
You lose, and your points are 0
Spin?y
4 1 3
You lose, and your points are 0
Spin?y
2 5 5
You win, and your points are 100
Spin?y
1 4 4
You lose, and your points are 0
Spin?y
4 2 2
You lose, and your points are 0
Spin?y
3 3 1
You lose, and your points are 0
Spin?y
3 3 4
```

Figure 9. Screenshot of exercise 4 non-cumulative score version

Programs in C and documentation

GRADEMARK REPORT

FINAL GRADE

/100

GENERAL COMMENTS

Instructor

Part A

Variables - You have covered the main point, but you need to talk about the different types ie int, float char. You have alluded to them but you could have given an example 3/5

Counting Loops - The main point here is that the number of iterations for a for loop is known before you start the loop. Example is suitable 4/5

Arrays You have covered the main points, you could also state that arrays start from 0 in most languages. Example is suitable 4.5/5

Part B

Exercise 1

Code-Annotation - You need to get the user to input the values, not hard code them. Then you could also validate age to between 0-120 etc 2/5

Documentation Flowchart is not detailed enough and doesn't use the right symbols. You need to show each in turn for name age and height. Testing is limited because the program is limited 1.5/5

Exercise 2

Code-Annotation A good effort with use of functions, but you have not validated any of the input. The values should not be negative, and the dimensions of the triangle and the circle relative to each other should be checked. 3/5

Documentation You don't need a branch in your flowchart, but otherwise OK. Note that input and output should be shown in a parallelogram symbol. Your testing only shows the "happy path" - you need to test with a wider range of values 2.5/5

Exercise 3

Code-Annotation A working solution, but you need to validate for negative values. Prices should not be negative. 3.5/5

Documentation - Again, testing is limited to the "happy path". You need to extend to other values eg all 0s or negative numbers to see how the program performs. 2.5 /5

Exercise 4

Code-Annotation Global variables are not good practice. srand should not appear in the while loop, it only need to be used once per program. Other than that good! You could have combined your last if statement by using a for loop to check for 5s, but this is a minor point. 8.5 /10

Documentation More detail is needed in the flowchart about the scoring scenarios. Your testing is comprehensive however 8.5/10

Total: 43.5/65 (67%)

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18
