BUPT SE ICP Quiz -- Unit 02 How to do the repeat

If you are using LLM to help you in this quiz, you MUST write down 1) the name of the LLM and 2) the PROMPTS. Otherwise, you will get zero score if the instructor finds that you are using LLM.
You may use the following table as a template, putting the table down to the title of each question.

Name of LLM	Prompts
NO LLM USED	

1. The three ways to do repeat in C programming language are ___while{}___, ___for(;;){}___ and ____do{}while();_____.
2. [F] T/F: In a for loop, when the test condition is true, the loop will stop.
3. [F] T/F: The construction `17;` is a legal statement in C, and it is ~~useful.~~ [RQ1, pp101]
4. Why do we need information coding when inputing or displaying all kinds of characters?

My Answer: To make sure the characters is well stored and avoiding the characters is too big that causes overflow.

5. What is the character at the 13rd position in ASCII table? What is the meaning of this character?

My Answer: \n(*CR (Carriage Return),referring to CSDN*); Means to change the cursor to the next line.

6. The distance of character ‘A’ and ‘a’ is _____32_____.

//97-65

7. The chinese character of your family name is ___侯____. The UTF-8 code of your family name is _______ e4 be af_________ (use hex number to represent the code)

// Haven’t download Chinese IME yet… using zi.tools to solve this…

8. Describe the effect of the following statement, assuming that i, j, and k are declared as integer variables: `i = (j + 4) * (k = 16);` [RQ2, pp101]

In this statement, the k is firstly assigned to 16, and i equals to 16*j + 64(j needs to be assigned before this to make sure this code could be compiled, (but it seems that C99 could assign j after this statement))

9. What is the purpose of each of the three expressions that appear in the control line of a for statement? [RQ20, pp102]

The first part is the initialization, a variable (“i” for example) will be declared when the initialization is running. The second part is the test-condition, to limit the code in the {} to be repeated as much as programmer want. The last expression is the increment, means the variable to increase so the loop could run well.

10. What for loop control line would you use in each of the following situations: [RQ21, pp102]
[A] a) Counting form 1 to 100.
[B] b) Counting by sevens starting at 0 until the number has more than two digits.
[A] c) Counting backward by twos form 100 to 0.
A. for loop; B. while loop; C. do-while loop;

11. Why is it best to avoid using a floating-point variable as the index variable in a for loop? [RQ22, pp102]

The floating-point variable takes more bytes than int, and all these numbers after "." is obviously inconvinient and may make the code uneasy to read. (Also, if just write " float i = 1",the 1 would be firstly storaged as an int, making "float" more useless)

Please implement the following 3 quizzes with for, while and do-while:

12. Print Integers from 1 to N: Write a program that uses a loop to print integers from 1 to N, where N is a positive integer entered by the user.

For Version

```
#include<stdio.h>
int main(void){
    int N;
    printf("Please enter a number: ");
    scanf("%d",&N);
    for(int i = 1; i <= N; i++){
        printf("%d\n", i);
    }
    return 0;
}
```

While Version

```
#include<stdio.h>
int main(void){
    int i = 1, N;
    printf("Please enter a number: ");
    scanf("%d",&N);
    while(i <= N){
        printf("%d\n", i);
        i++;
    }
    return 0;
}
```

Do-While Version

```
#include<stdio.h>
int main(void){
    int i = 1, N;
    printf("Please enter a number: ");
    scanf("%d",&N);
    do {printf("%d\n", i);
    i++;
    }while(i <= N);
    return 0;
}
```

13. Calculate the Sum of Integers from 1 to N: Modify the previous program to calculate the sum of all integers from 1 to N and print the result.

FOR VERSION

```
#include<stdio.h>
int main(void){
    int N, sum;
    printf("Please enter a number: ");
    scanf("%d",&N);
    for(int i = 1; i <= N; i++){
        sum += i;
    }
    printf("The sum is %d\n",sum);
return 0;
}
```

WHILE VERSION

```
#include<stdio.h>
int main(void){
    int i = 1, N, sum;
    printf("Please enter a number: ");
    scanf("%d",&N);
    while(i <= N){
        sum += i;
        i++;
    }
    printf("The sum is %d\n",sum);
    return 0;
}
```

DO-WHILE version

```
#include<stdio.h>
int main(void){
    int i = 1, N, sum;
    printf("Please enter a number: ");
    scanf("%d",&N);
    do{
        sum += i;
        i++;
    }while(i <= N);
    printf("The sum is %d\n",sum);
    return 0;
}
```

14. Calculate the Factorial of N: Write a program that uses a loop to calculate the factorial of a positive integer N entered by the user and print the result. Factorial is defined as N! = 1 * 2 * 3 * ... * N.

For VERSION

```
#include<stdio.h>
int main(void){
    int N, factorial = 1;
    printf("Please enter a number: ");
    scanf("%d",&N);
    for(int i = 1; i <= N; i++){
        factorial *= i;
    }
    printf("The factorial is %d\n",factorial);
    return 0;
}
```

WHILE VERSION

```
#include<stdio.h>
int main(void){
    int i = 1, N, factorial = 1;
    printf("Please enter a number: ");
    scanf("%d",&N);
    while(i <= N){
        factorial *= i;
        i++;
    }
    printf("The factorial is %d\n",factorial);
    return 0;
}
```

DO-WHILE VERSION

```
#include<stdio.h>
int main(void){
    int i = 1, N, factorial = 1;
    printf("Please enter a number: ");
    scanf("%d",&N);
    do{
        factorial *= i;
        i++;
    }while(i <= N);
    printf("The factorial is %d\n",factorial);
    return 0;
}
```

15. 打印乘法表：

编写一个程序，使用嵌套循环打印出一个乘法表，从 1 乘到 N，其中 N 是用户输入的正整数。例如，如果用户输入 N 为 5，程序应该打印出以下乘法表（可能需要嵌套的循环）：

```
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15
4 8 12 16 20
5 10 15 20 25
```

FOR VERSION

```
#include<stdio.h>
int main(void){
    int N;
    printf("Please enter a number: ");
    scanf("%d",&N);
    for(int i=1;i<=N;i++){
        for(int j=1;j<=N;j++){
        printf("%d ",i*j);
        }
    printf("\n");
    }
    return 0;
}
```

WHILE VERSION

```
#include<stdio.h>
int main(void){
    int i = 1, j = 1,N;
    printf("Please enter a number: ");
    scanf("%d",&N);
    while(i<=N){
        while(j<=N){
            printf("%d ",i*j);
            j++;
        }
        i++;
        j = 1;
        printf("\n");
    }
    return 0;
}
```

DO-WHILE VERSION

```
#include<stdio.h>
int main(void){
    int i = 1, j = 1,N;
    printf("Please enter a number: ");
    scanf("%d",&N);
    do {
        do {
            printf("%d ", i*j);
            j++;
        } while (j <= N);
        i++;
        j = 1;
        printf("\n");
    } while (i <= N);
    return 0;
}
```

16. 计算阶乘和：

编写一个程序，使用嵌套循环计算并打印出 1 到 N 的所有正整数的阶乘和，其中 N 是用户输入的正整数。阶乘和是指 1! + 2! + 3! + ... + N! 的总和。

FOR VERSION

```
#include<stdio.h>
int main(void){
    int N, f = 1; /*f for Factorial*/
     long long sum = 0; // Cauz when I tried to run it ,I found that the sum exceeded the range of int
    printf("Please enter a number: ");
    scanf("%d",&N);
    for(int k = 1; k <= N; k++){ // Loop from 1 to N
        for(int i=1;i<=k;i++){ // Calculate sum of factorials from 1 to k
            for(int j=1;j<=i;j++){ // Calculate factorial of i
                f *= j;
            }
            sum += f;
            f = 1; // Reset f for next factorial calculation
        }
        printf("%lld\n", sum) ;
    }
    return 0;
}
```

WHILE VERSION

```
#include<stdio.h>
int main(void){
    int N, f = 1; /*f for Factorial*/
    long long sum = 0; // Cauz when I tried to run it ,I found that the sum exceeded the range of int
    printf("Please enter a number: ");
    scanf("%d",&N);
    int k = 1;
    while (k <= N) { // Loop from 1 to N
        int i = 1;
        while (i <= k) { // Calculate sum of factorials from 1 to k
            int j =1;
            while (j <= i) { // Calculate factorial of i
                f *= j;
                j++;
            }
            sum += f;
            f = 1; // Reset f for next factorial calculation
            i++;
        }
        printf("%lld\n",sum);
        k++;
    }
    return 0;
}
```

DO-WHILE VERSION

```
#include<stdio.h>
int main(void){
    int N, f = 1; /*f for Factorial*/
     long long sum = 0; // Cauz when I tried to run it ,I found that the sum exceeded the range of int
    printf("Please enter a number: ");
    scanf("%d",&N);
    int k = 1;
    do { // Loop from 1 to N
        int i = 1;
        do { // Calculate sum of factorials from 1 to k
            int j = 1;
            do { // Calculate factorial of i
                f *= j;
                j++;
            } while (j <= i);
            sum += f;
            f = 1; // Reset f for next factorial calculation
            i++;
        } while (i <= k);
        printf("%lld\n", sum);
        k++;
    } while (k <= N);
    return 0;
}
```

[**OPTIONAL/选做**] The following 2 quizzes need if-else expressions. If you want, just try them out.

17. Check for Prime Number: Write a program that uses a loop to check if a user-entered integer is a prime number (a number that is divisible only by 1 and itself). Print the result indicating whether the number is prime.

```
#include<stdio.h>
int main(void){
    int n, N , boole = 1; //Use boole as a medium
    scanf("%d",&N);
    for(n=2; n*n<=N; n++){ //No 1
        if(N%n != 0){
            boole = 1;
        }
        else if(N%n == 0){
            boole = 0;
            break;
        }
    }
    if(boole == 1){
        printf("%d is a prime number\n",N);
    }
    else{
        printf("%d is not a prime number\n",N);
    }
    return 0;
}
```

18. Generate the Fibonacci Sequence: Write a program that uses a loop to generate the first N Fibonacci numbers (where each number in the Fibonacci sequence is the sum of the two preceding ones). Print the generated sequence.

```
#include <stdio.h>
int main(void) {
    unsigned int N;
    printf("Enter a number ");
    scanf("%u", &N);
    int a = 1, b = 1;
    long long nextnum; //nextnum could be large, so use long long
    printf("%d %d ", a, b);
    for (int i = 1; i <= N; i++) {
        nextnum = a + b;
        printf("%lld ", nextnum);
        a = b;
        b = nextnum;
    }
    printf("\n");
    return 0;
}
```