



Jiangxi University of Science and Technology

Chapter 5 Repetition

Lecture0503 Using a while Loop

THE
C
PROGRAMMING
LANGUAGE



5.3 Computing Sums and Averages Using a while Loop

Program 5.5

Counting

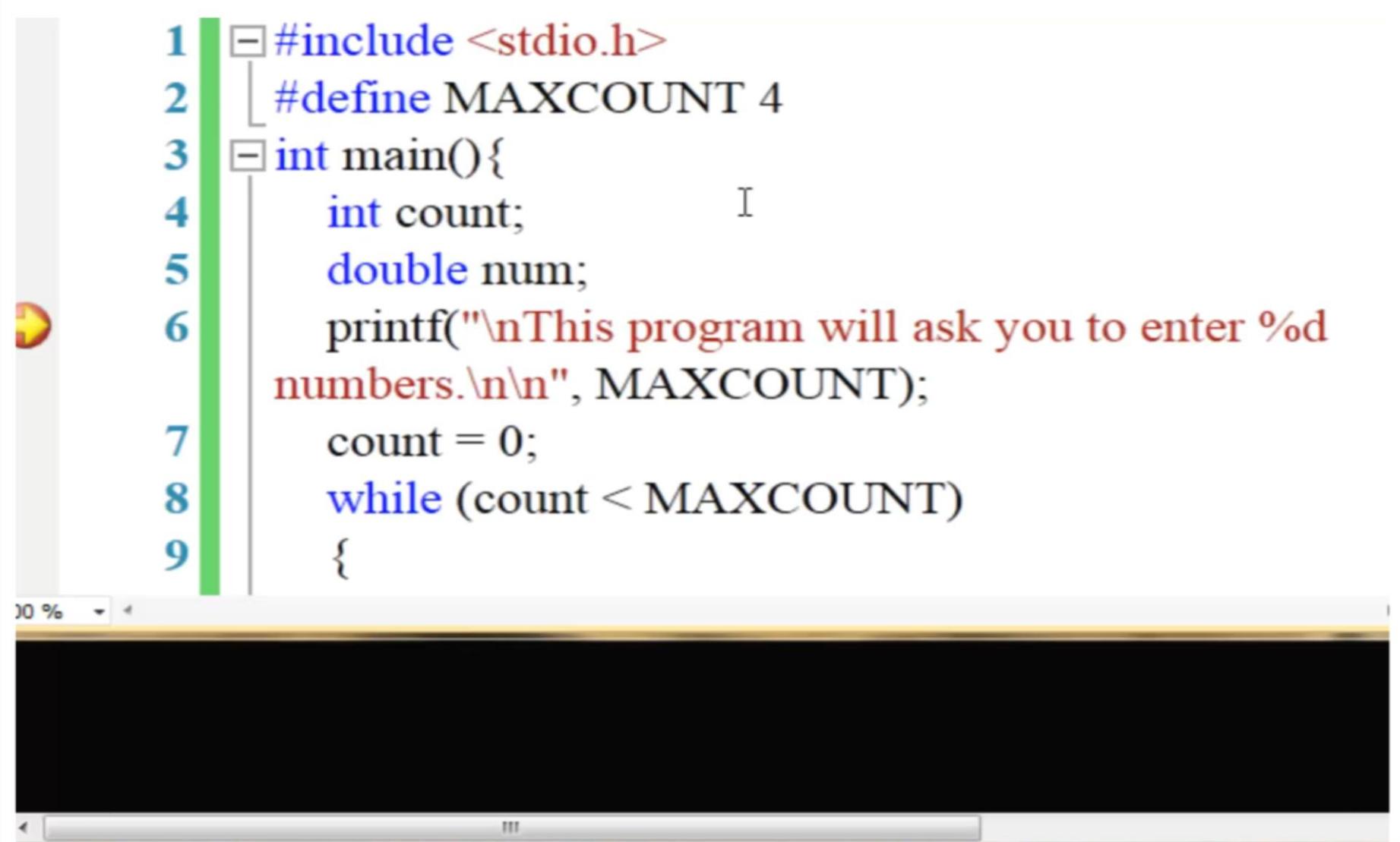
计数

TEST
ME!

```
1. #include <stdio.h>
2. #define MAXCOUNT 4
3. int main()
4. {
5.     int count;
6.     double num;
7.     printf("\nThis program will ask you to enter %d numbers.\n\n", MAXCOUNT);
8.     count = 0;
9.     while (count <MAXCOUNT)
10.    {
11.        printf("Enter a number: ");
12.        scanf("%lf", &num);
13.        printf("The number entered is %f\n", num);
14.        count++;
15.    }
16.    return 0;
17. }
```

5.3 Computing Sums and Averages Using a while Loop

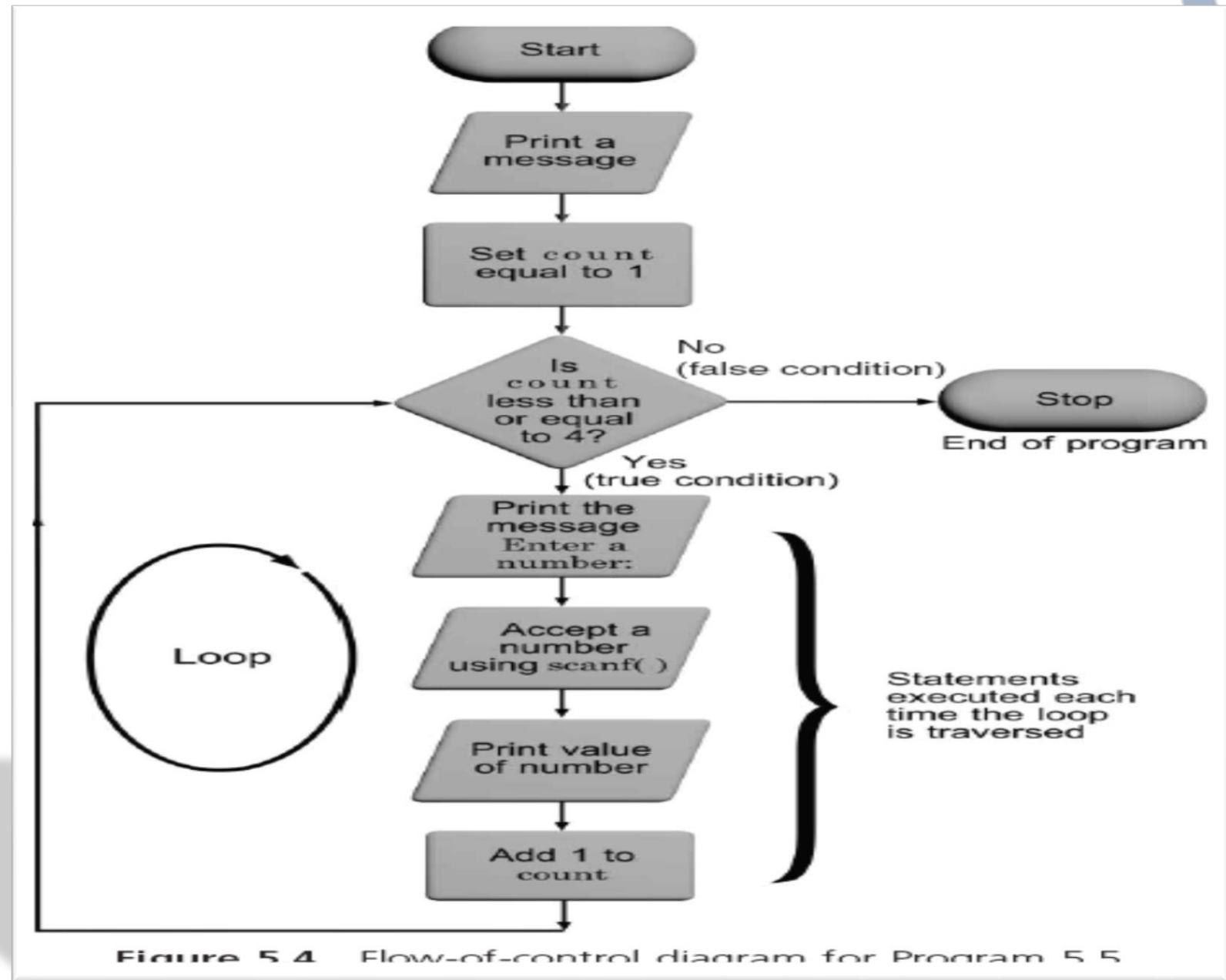
Program 5.5 Counting



```
1 #include <stdio.h>
2 #define MAXCOUNT 4
3 int main(){
4     int count;
5     double num;
6     printf("\nThis program will ask you to enter %d
numbers.\n\n", MAXCOUNT);
7     count = 0;
8     while (count < MAXCOUNT)
9     {
```

5.3 Computing Sums and Averages Using a while Loop

➤ Program 5.5 Counting



5.3 Computing Sums and Averages Using a while Loop

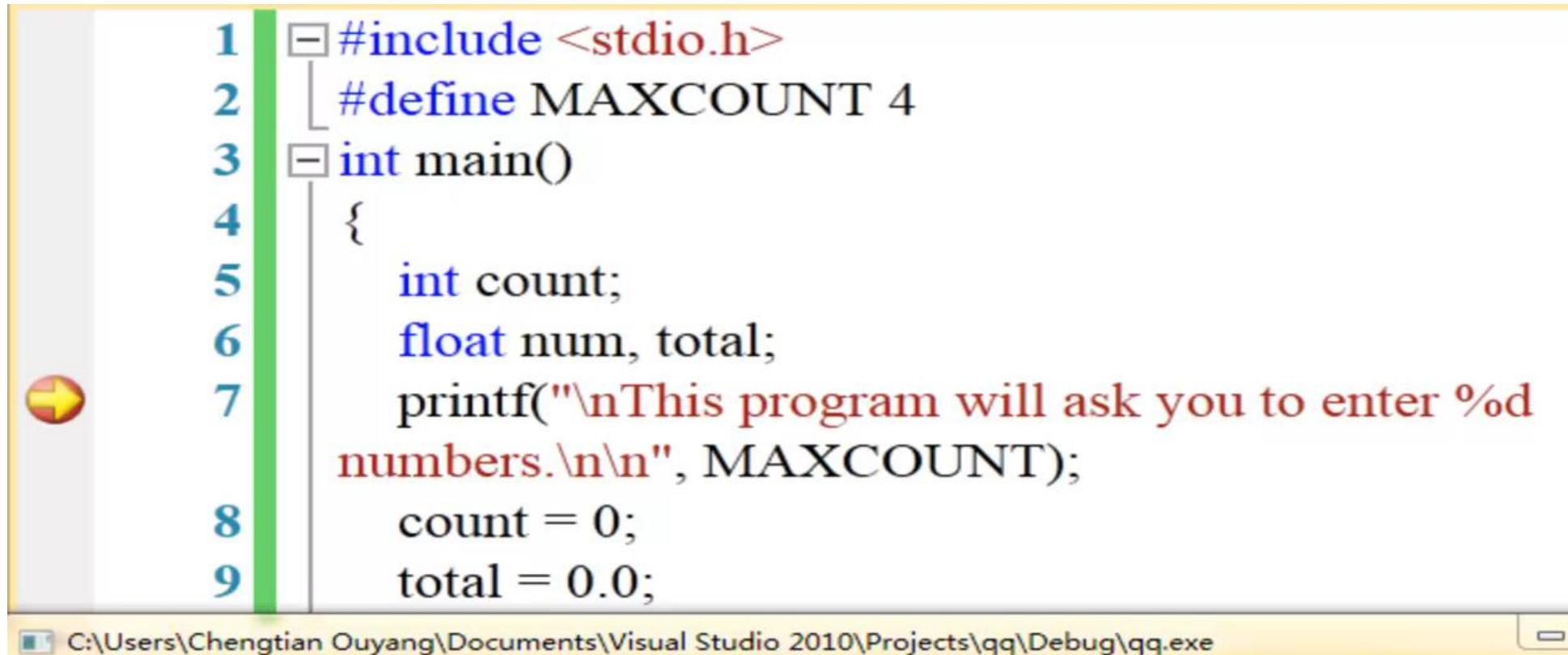
Program 5.6 Accumulation 累加

TEST
ME!

```
1. #include <stdio.h>
2. #define MAXCOUNT 4
3. int main(){
4.     int count;
5.     float num, total;
6.     printf("\nThis program will ask you to enter %d numbers.\n\n", MAXCOUNT);
7.     count = 0;
8.     total = 0.0;
10.    while (count < MAXCOUNT){
11.        printf("Enter a number: ");
12.        scanf("%f", &num);
13.        total += num;
14.        printf("The total is now %f\n", total);
15.        count++;
16.    }
17.    printf("\n\nThe final total of the %d numbers is %f\n", count, total);
18.    return 0;
19. }
```

5.3 Computing Sums and Averages Using a while Loop

➤ Program 5.6 Accumulation



The screenshot shows a Microsoft Visual Studio 2010 IDE window. On the left is a code editor with a vertical line of line numbers from 1 to 9. The code itself is a C program:

```
1 #include <stdio.h>
2 #define MAXCOUNT 4
3 int main()
4 {
5     int count;
6     float num, total;
7     printf("\nThis program will ask you to enter %d
numbers.\n\n", MAXCOUNT);
8     count = 0;
9     total = 0.0;
```

The status bar at the bottom displays the path: C:\Users\Chengtian Ouyang\Documents\Visual Studio 2010\Projects\qq\Debug\qq.exe.

5.3 Computing Sums and Averages Using a while Loop

➤ Program 5.6 Accumulation

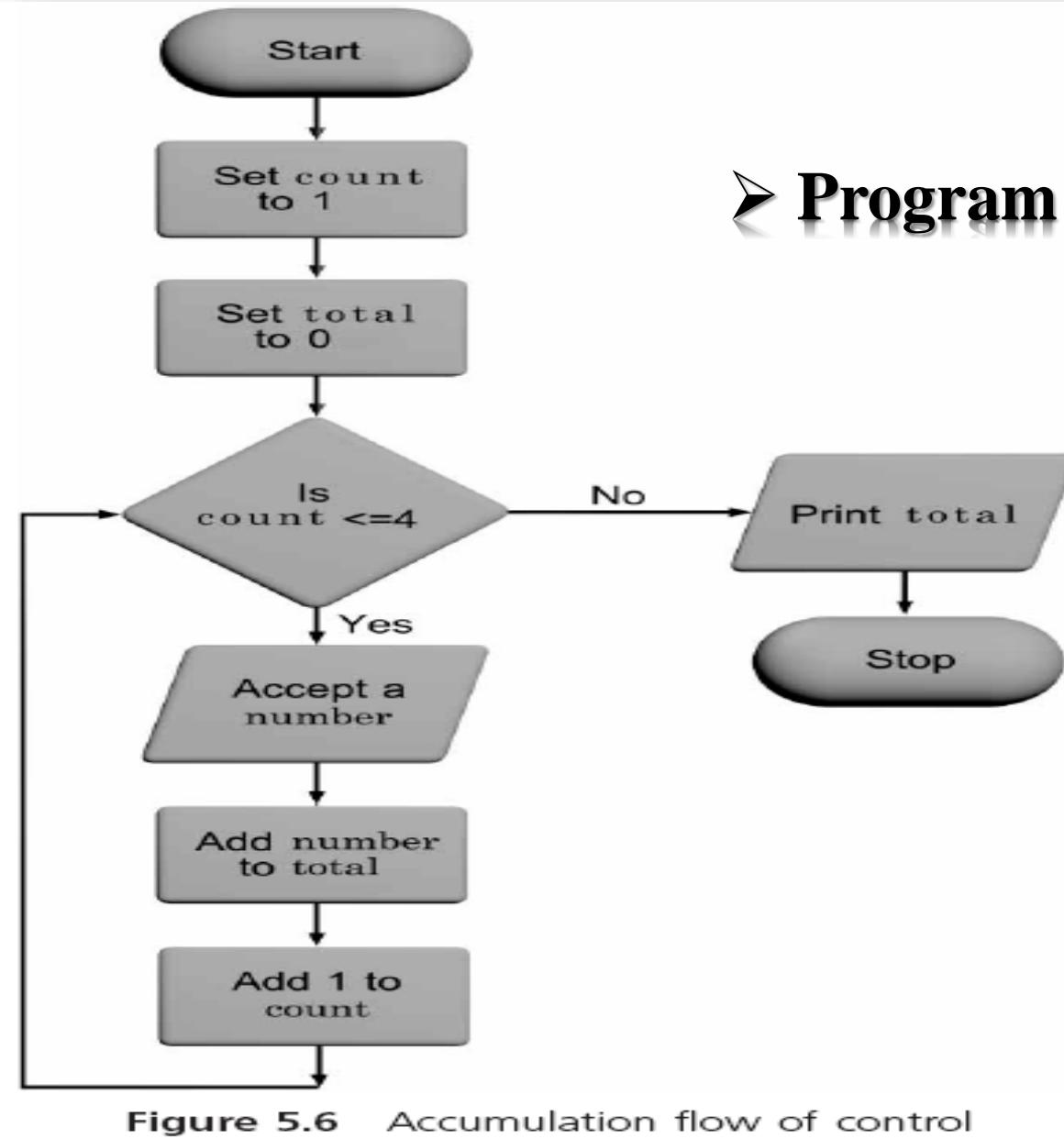


Figure 5.6 Accumulation flow of control

5.3 Computing Sums and Averages Using a while Loop

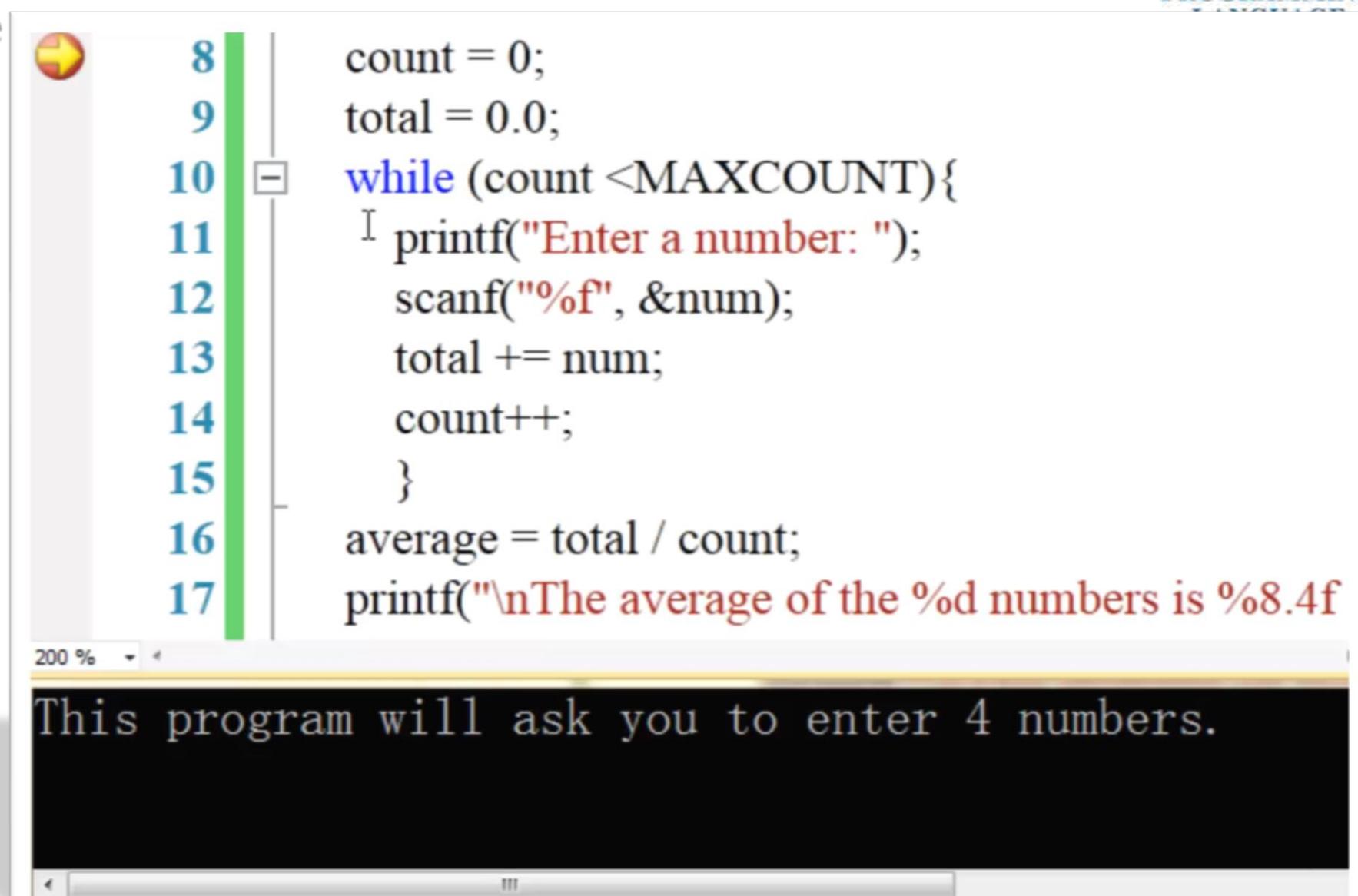
Program 5.7 Average

TEST
ME!

```
1. #include <stdio.h>
2. #define MAXCOUNT 4
3. int main()
4. {
5.     int count;
6.     float num, total, average;
7.     printf("\nThis program will ask you to enter %d numbers.\n\n", MAXCOUNT);
8.     count = 0;
9.     total = 0.0;
10.    while (count <MAXCOUNT){
11.        printf("Enter a number: ");
12.        scanf("%f", &num);
13.        total += num;
14.        count++;
15.    }
16.    average = total / count;
17.    printf("\nThe average of the %d numbers is %8.4f\n", count, average);
18.    return 0;
19. }
```

5.3 Computing Sums and Averages Using a while Loop

Program 5.7 Average



```
8 count = 0;
9 total = 0.0;
10 while (count < MAXCOUNT){
11     printf("Enter a number: ");
12     scanf("%f", &num);
13     total += num;
14     count++;
15 }
16 average = total / count;
17 printf("\nThe average of the %d numbers is %8.4f
```

This program will ask you to enter 4 numbers.

5.3 Computing Sums and Averages Using a while Loop

➤ Sentinels 哨兵

- A program, such as Program 5.7, can be made much more general by **removing the restriction** that exactly **four numbers** are to be entered
- The user enters a value for how many numbers will be averaged
- You can use a **sentinel** (a data value used to signal either the start or end of a data series)
- The sentinel values must be selected ***so as not to conflict with*** legitimate data values

5.3 Computing Sums and Averages Using a while Loop

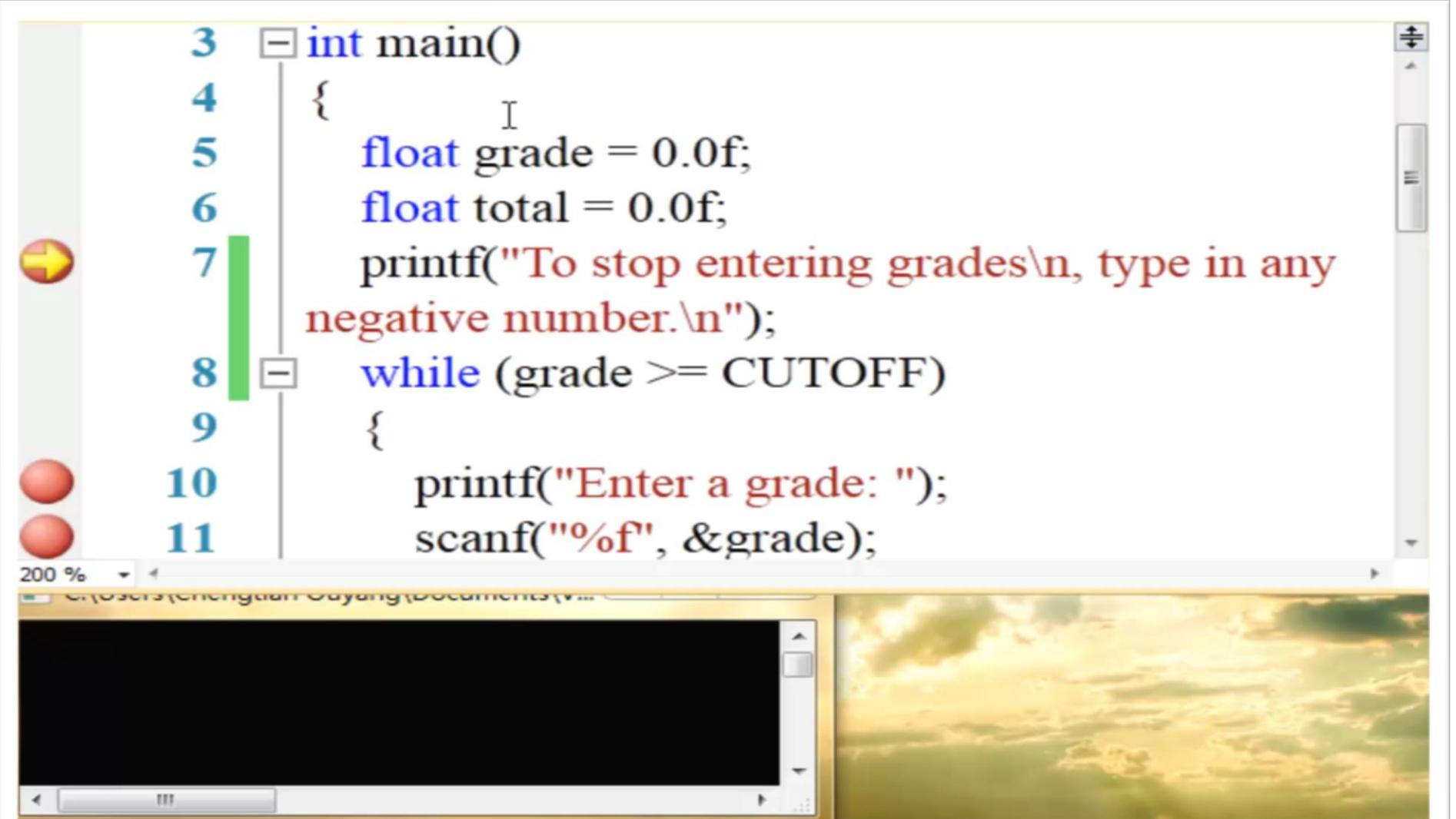
Program 5.8 Sentinels

TEST
ME!

```
1. #include <stdio.h>
2. #define CUTOFF 0
3. int main(){
4.     float grade = 0.0f;
5.     float total = 0.0f;
6.     printf("\nTo stop entering grades, type in any negative number.\n\n");
7.     while(grade >= CUTOFF){
8.         printf("Enter a grade: ");
9.         scanf("%f", &grade);
10.        total = total + grade;
11.    }
12.    total=total-grade;
13.    printf("\nThe total of the grades is %f\n", total);
14.    return 0;
15. }
```

5.3 Computing Sums and Averages Using a while Loop

Program 5.8 Sentinels

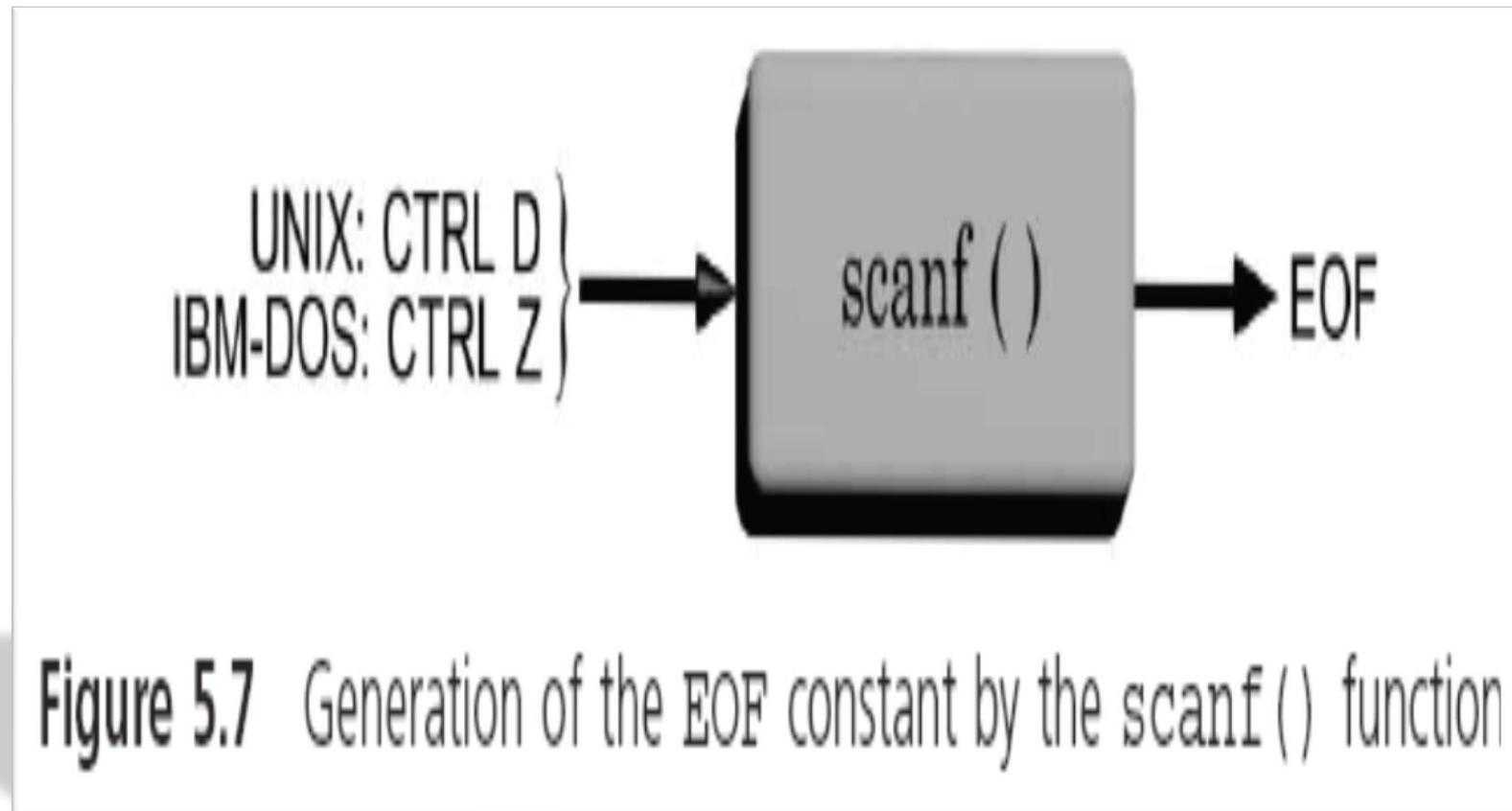


```
3 int main()
4 {
5     float grade = 0.0f;
6     float total = 0.0f;
7     printf("To stop entering grades\n, type in any
8         negative number.\n");
9     while (grade >= CUTOFF)
10    {
11        printf("Enter a grade: ");
12        scanf("%f", &grade);
```

5.3 Computing Sums and Averages Using a while Loop

➤ Sentinels

- *One useful sentinel* in C is the named constant **EOF** (End Of File)
- The actual value of EOF is **compiler-dependent**.
- **EOF is defined in stdio.h**



5.3 Computing Sums and Averages Using a while Loop

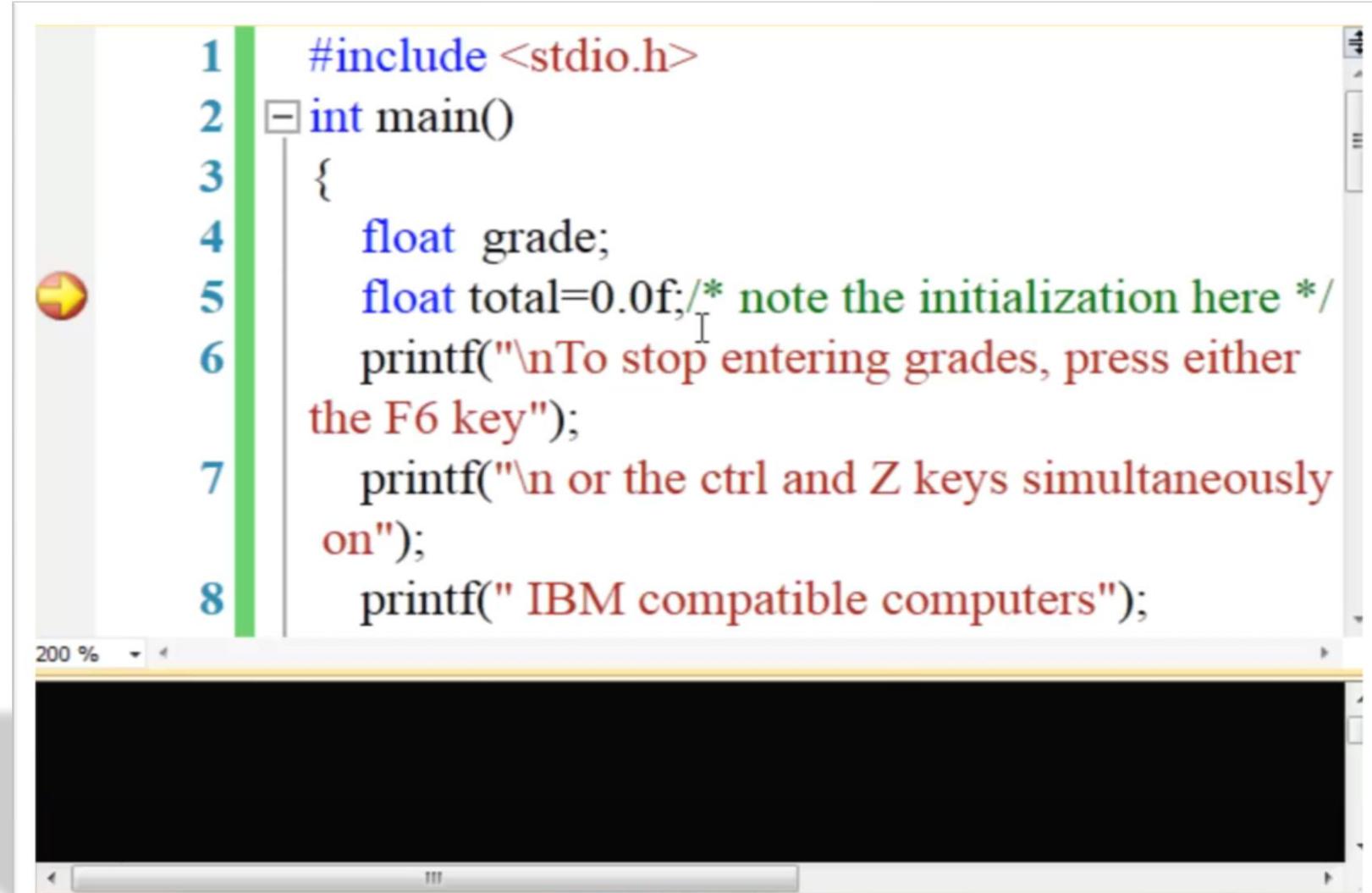
TEST
ME!

➤ Program 5.9 End Of File

```
1. #include <stdio.h>
2. int main(){
3.     float grade;
4.     float total = 0.0f; //note the initialization here
5.     printf("\nTo stop entering grades, press either the F6 key");
6.     printf("\n or the ctrl and Z keys simultaneously on");
7.     printf(" IBM compatible computers");
8.     printf("\n or the ctrl and D keys for UNIX operating systems.\n\n");
10.    printf("Enter a grade: ");
11.    while (scanf("%f", &grade) != EOF)
12.    {
13.        total += grade;
14.        printf("Enter a grade: ");
15.    }
16.    printf("\nThe total of the grades is %f\n",total);
17.    return 0;
18. }
```

5.3 Computing Sums and Averages Using a while Loop

Program 5.9 End Of File



```
1 #include <stdio.h>
2 int main()
3 {
4     float grade;
5     float total=0.0f; /* note the initialization here */
6     printf("\nTo stop entering grades, press either
7 the F6 key");
8     printf("\n or the ctrl and Z keys simultaneously
on");
9     printf(" IBM compatible computers");
```

5.3 Computing Sums and Averages Using a while Loop

➤ The break Statement

- A **break** forces an immediate exit from **while**, **switch**, **for**, and **do-while** statements only

➤ The continue Statement

- The **continue** applies to **loops** only; when a continue statement is encountered in a loop, the next iteration of the loop begins immediately

5.3 Computing Sums and Averages Using a while Loop

➤ Program 1282 prime number

- *A prime number* is a natural number greater than 1 that has no positive divisors other than 1 and itself. A natural number greater than 1 that is not a prime number is called *a composite number*.
- Enter a natural number, determine whether it is a prime, if it is a prime, print “*? is a prime number*”. Otherwise, print “*? is not a prime number*”.
- If the input is not a natural number, Please continue to enter another number. To stop entering numbers, press the EOF key.

5.3 Computing Sums and Averages Using a while Loop

Program 1282 prime number

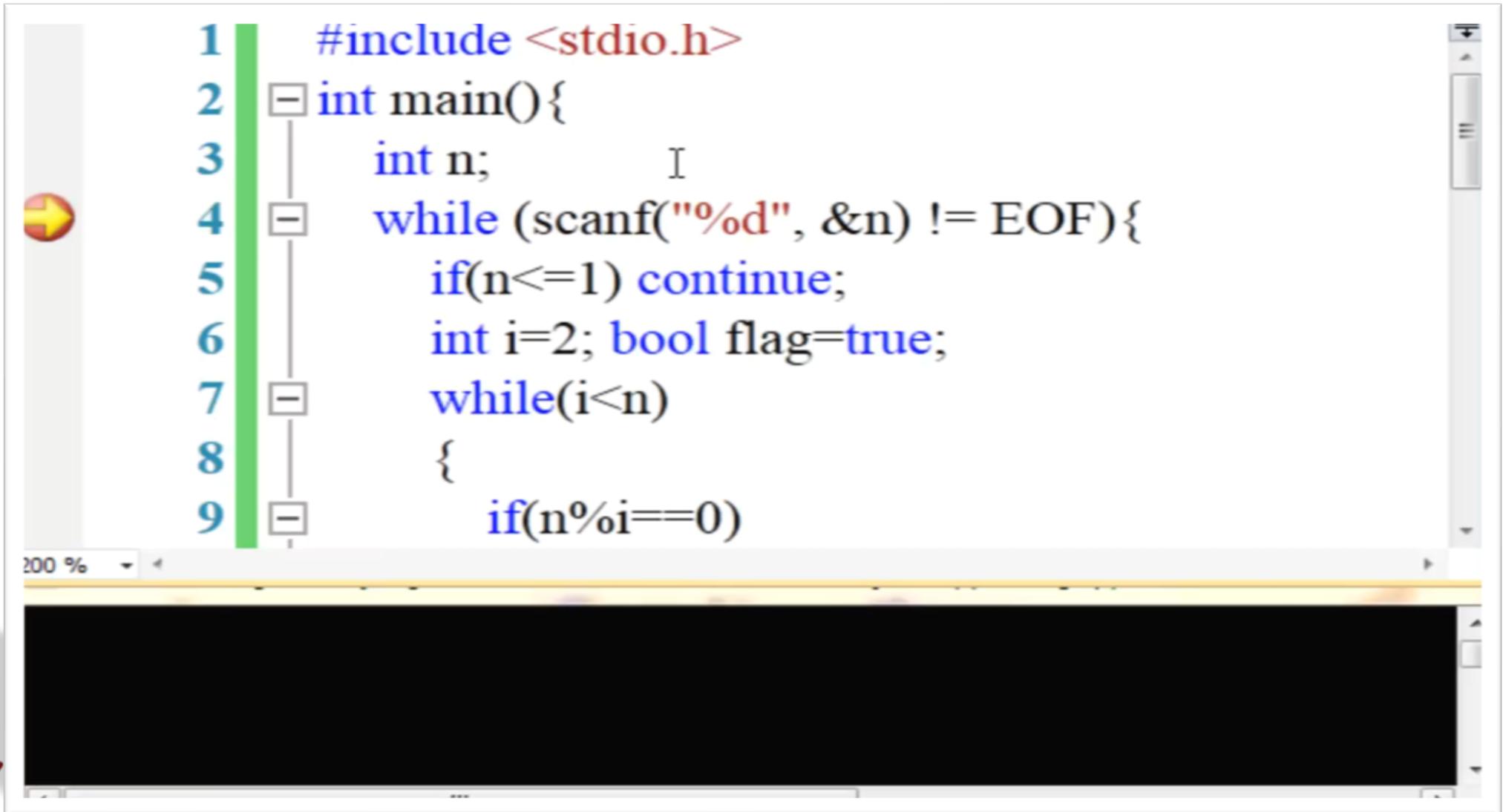
TEST
ME!

```
1. #include <stdio.h>
2. int main(){
3.     int n;
4.     while (scanf("%d", &n) != EOF){
5.         if(n<=1) continue;
6.         int i=2; bool flag=true;
7.         while(i<n){
8.             if(n%i==0) {flag=false; break;}
9.             i++;
10.        }
11.        if ( flag )
12.            printf("%d is a prime number\n", n);
13.        else
14.            printf("%d is a composite number\n", n);
15.    }
16.    return 0;
17. }
```



5.3 Computing Sums and Averages Using a while Loop

- Program 1282 prime number



The screenshot shows a code editor window with a green vertical line highlighting the first nine lines of C code. The code is a program to check if a number is prime. It includes an include directive, a main function, variable declarations, a while loop for input, an if statement to skip 1, and a nested while loop for checking divisibility from 2 to n-1. The code editor has a toolbar on the left with a red play button icon, a status bar at the bottom showing '200 %' zoom, and scroll bars on the right.

```
1 #include <stdio.h>
2 int main(){
3     int n;
4     while (scanf("%d", &n) != EOF){
5         if(n<=1) continue;
6         int i=2; bool flag=true;
7         while(i<n)
8         {
9             if(n%i==0)
```

5.3 Computing Sums and Averages Using a while Loop

➤ The Null Statement

- A semicolon with nothing preceding it is also a valid statement, called the null statement
 - ;
- Use the null statement where a statement is syntactically required, but no action is needed
- Null statements typically are used either with while or for statements

Reference

- BOOK
- Some part of this PPT given by Prof 欧阳城添
(Prof: Chengtian Quyang)
- with special thank
- <https://www.codingunit.com/c-tutorial-first-c-program-hello-world>

