Under the C

GDSC class

22 December, 2024

https://github.com/Mr-Tony-Lee/GDSC_Cprog.git

Today's Lesson

1. Quick Review
2. Logic Training
3. Some Classic Problem

●●●●●

Topic : Quick Review

1. basic output
2. basic input
3. if/else
4. for/while loop

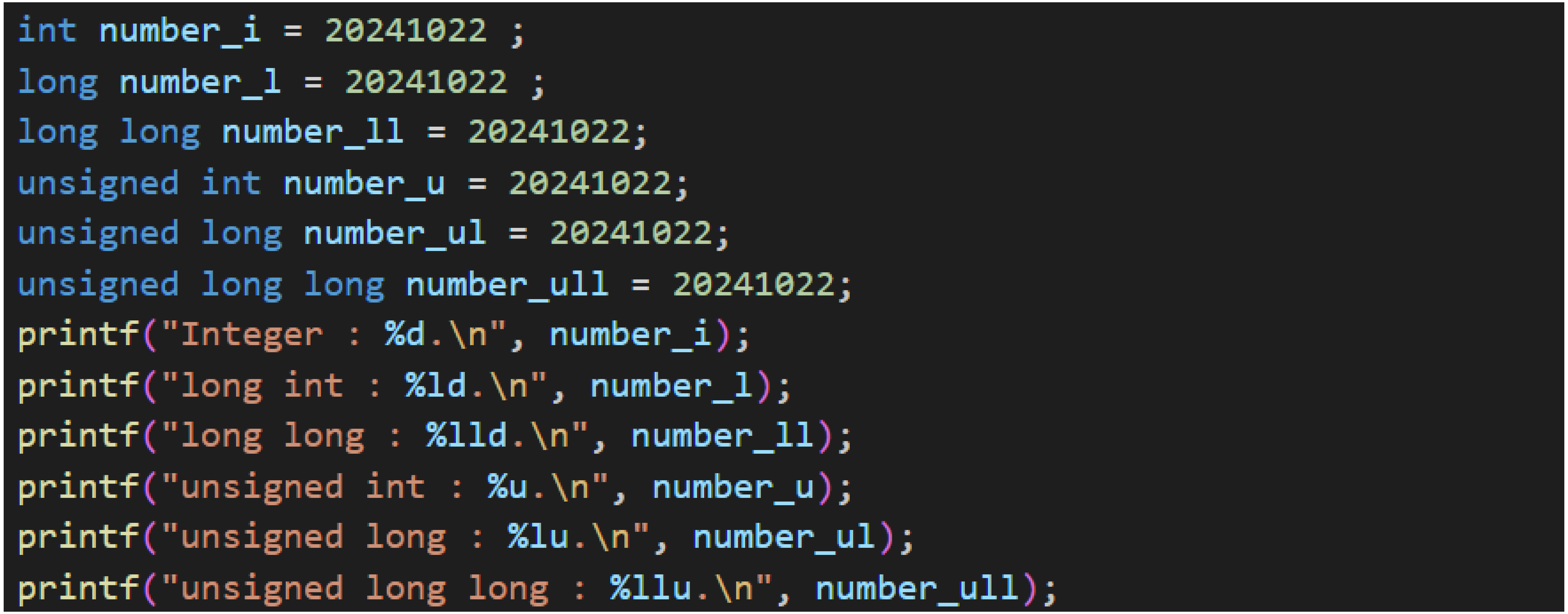
●●●●●

1. *Basic output*

printf : output decimal number

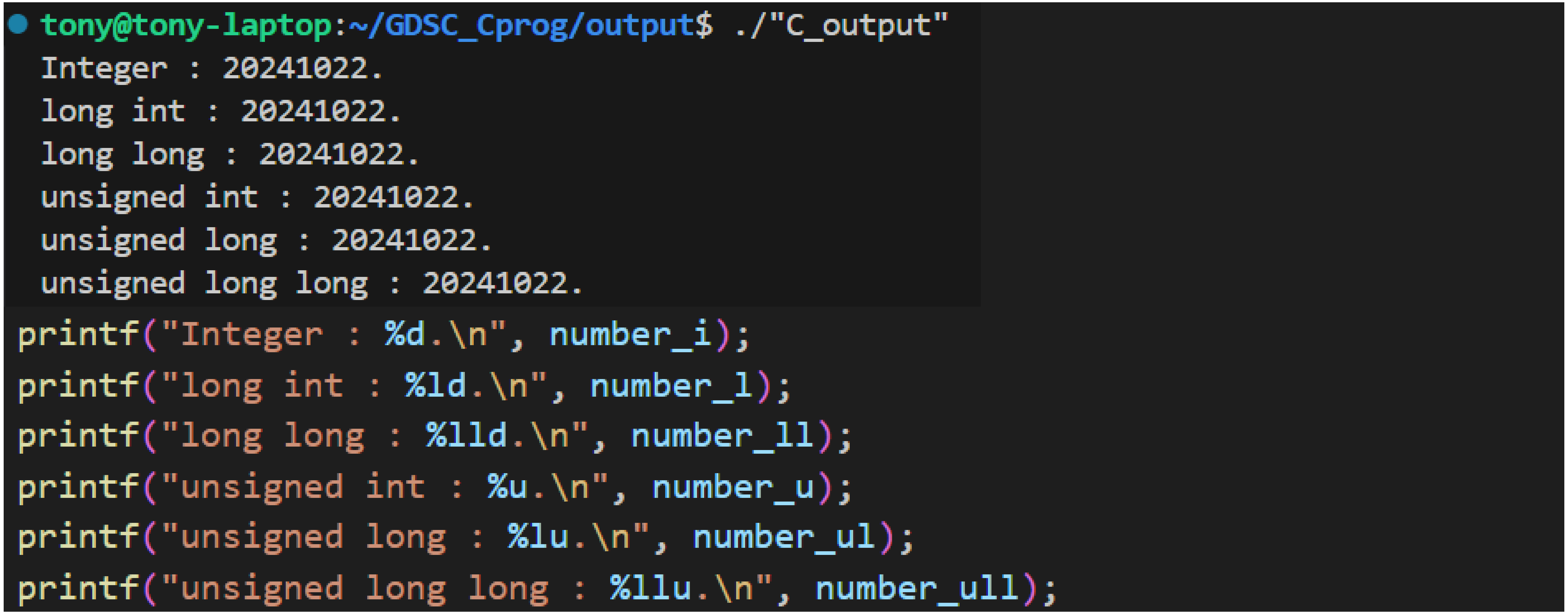
%d	>> Output decimal number(int)
%ld	>> Output long int
%lld	>> Output long long int
%u	>> Output unsigned int
%lu	>> Output unsigned long int
%llu	>> Output unsigned long long int

printf : output decimal number



```c
int number_i = 20241022 ;
long number_l = 20241022 ;
long long number_ll = 20241022;
unsigned int number_u = 20241022;
unsigned long number_ul = 20241022;
unsigned long long number_ull = 20241022;
printf("Integer : %d.\n", number_i);
printf("long int : %ld.\n", number_l);
printf("long long : %lld.\n", number_ll);
printf("unsigned int : %u.\n", number_u);
printf("unsigned long : %lu.\n", number_ul);
printf("unsigned long long : %llu.\n", number_ull);
```

printf : output decimal number



```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_output"
 Integer : 20241022.
 long int : 20241022.
 long long : 20241022.
 unsigned int : 20241022.
 unsigned long : 20241022.
 unsigned long long : 20241022.
```

```c
printf("Integer : %d.\n", number_i);
printf("long int : %ld.\n", number_l);
printf("long long : %lld.\n", number_ll);
printf("unsigned int : %u.\n", number_u);
printf("unsigned long : %lu.\n", number_ul);
printf("unsigned long long : %llu.\n", number_ull);
```

printf : output decimal number

%nd   >> To print the number with a total of at least ‘n’ space covered. (number at right side .)

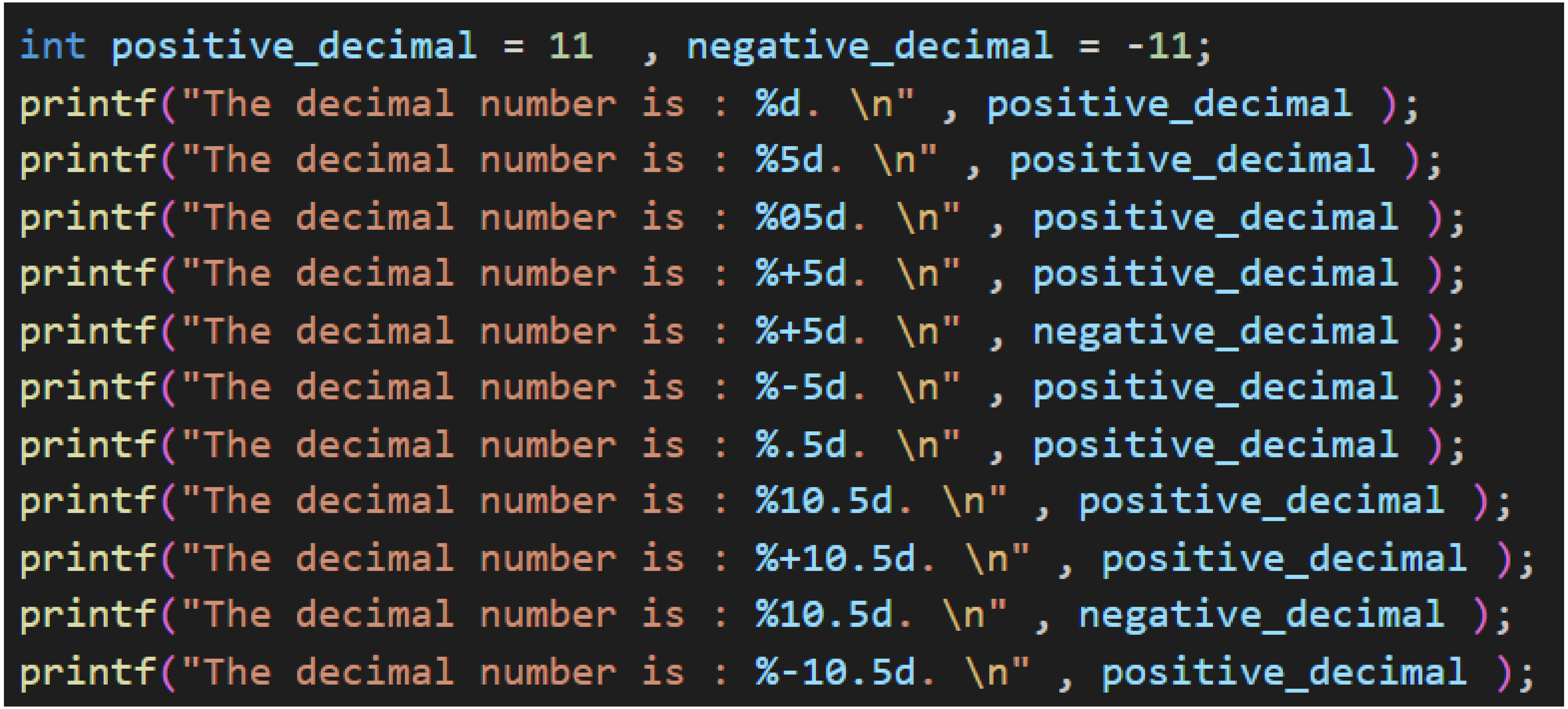
%0nd >> Same as above, except pre-padding with 0′s

%+d   >> If number > 0 , that will be a ‘+’ before the number.

%-nd >> To print the number with a total of at least ‘n’ space covered. (number at left side .)

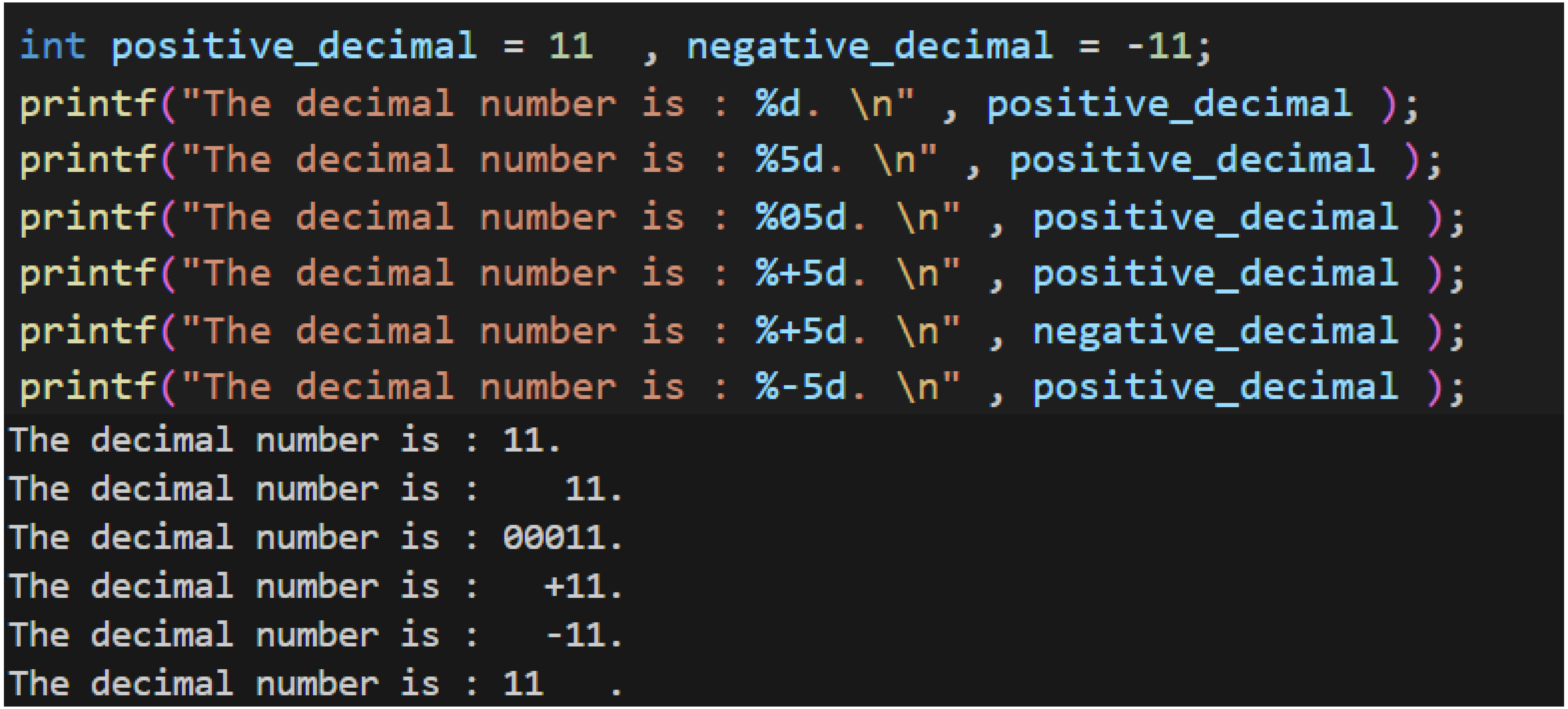
%.nd >> In decimal output , same as %0nd.

printf : output decimal number



```c
int positive_decimal = 11  , negative_decimal = -11;
printf("The decimal number is : %d. \n" , positive_decimal );
printf("The decimal number is : %5d. \n" , positive_decimal );
printf("The decimal number is : %05d. \n" , positive_decimal );
printf("The decimal number is : %+5d. \n" , positive_decimal );
printf("The decimal number is : %+5d. \n" , negative_decimal );
printf("The decimal number is : %-5d. \n" , positive_decimal );
printf("The decimal number is : %.5d. \n" , positive_decimal );
printf("The decimal number is : %10.5d. \n" , positive_decimal );
printf("The decimal number is : %+10.5d. \n" , positive_decimal );
printf("The decimal number is : %10.5d. \n" , negative_decimal );
printf("The decimal number is : %-10.5d. \n" , positive_decimal );
```

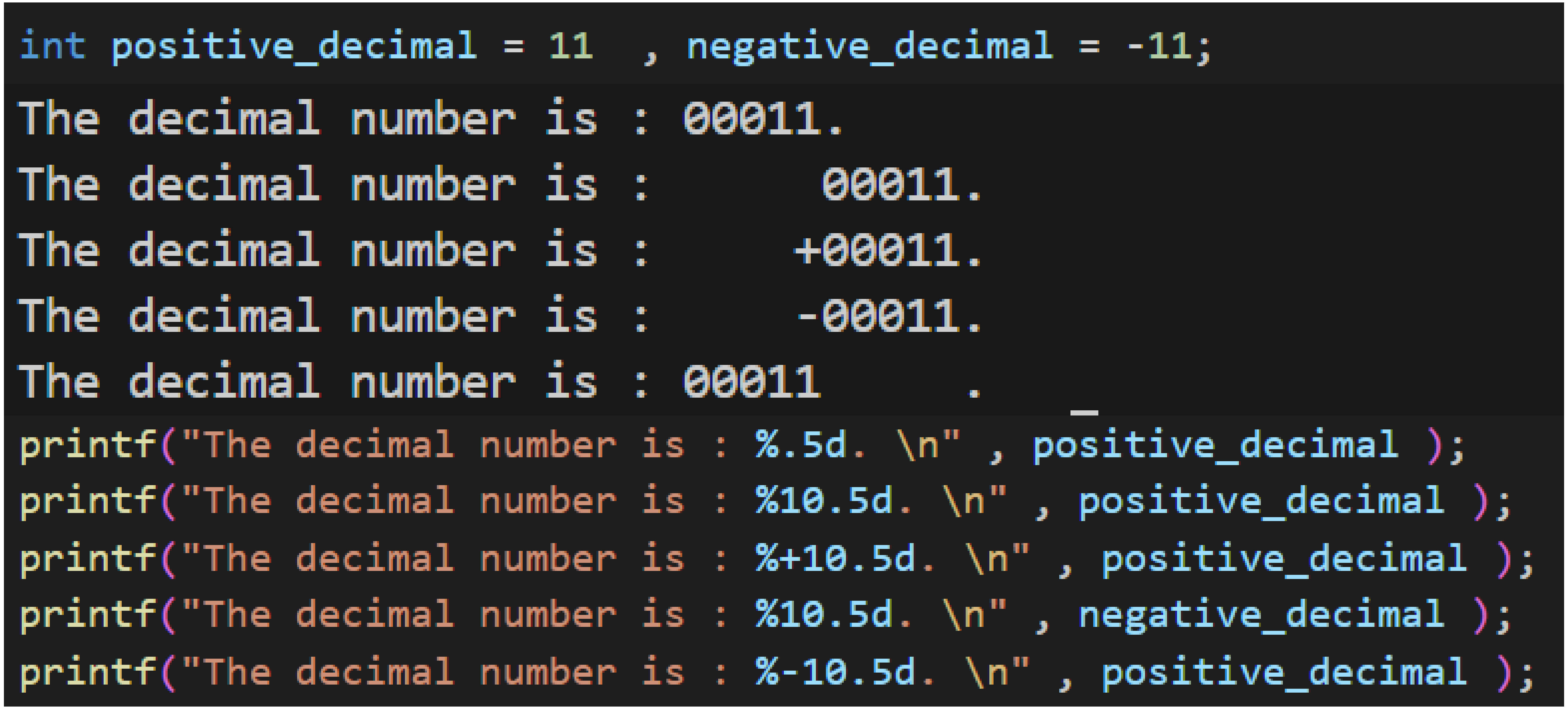
printf : output decimal number



```c
int positive_decimal = 11  , negative_decimal = -11;
printf("The decimal number is : %d. \n" , positive_decimal );
printf("The decimal number is : %5d. \n" , positive_decimal );
printf("The decimal number is : %05d. \n" , positive_decimal );
printf("The decimal number is : %+5d. \n" , positive_decimal );
printf("The decimal number is : %+5d. \n" , negative_decimal );
printf("The decimal number is : %-5d. \n" , positive_decimal );
```

```
The decimal number is : 11.
The decimal number is :    11.
The decimal number is : 00011.
The decimal number is :   +11.
The decimal number is :   -11.
The decimal number is : 11   .
```

printf : output decimal number



```c
int positive_decimal = 11  , negative_decimal = -11;
```

```
The decimal number is : 00011.
The decimal number is :      00011.
The decimal number is :     +00011.
The decimal number is :     -00011.
The decimal number is : 00011     .
```

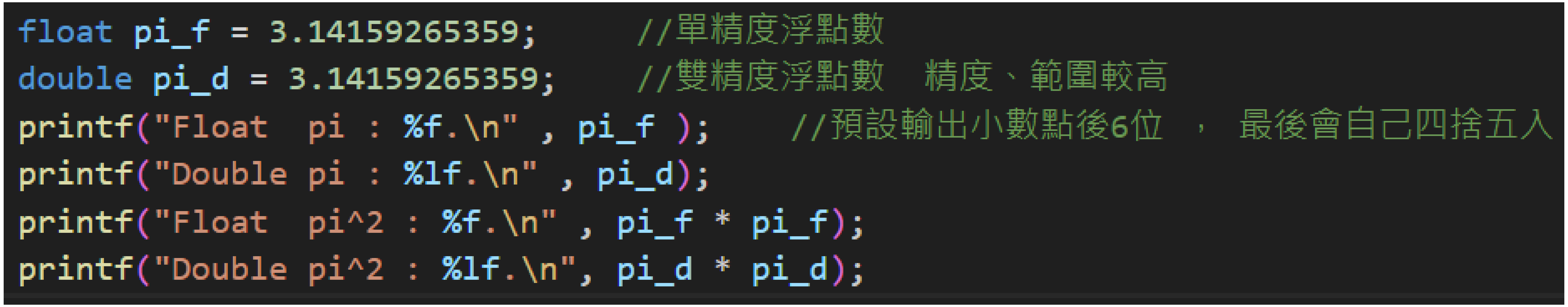
```c
printf("The decimal number is : %.5d. \n" , positive_decimal );
printf("The decimal number is : %10.5d. \n" , positive_decimal );
printf("The decimal number is : %+10.5d. \n" , positive_decimal );
printf("The decimal number is : %10.5d. \n" , negative_decimal );
printf("The decimal number is : %-10.5d. \n" , positive_decimal );
```

printf : output floating number

Float >> Float is used to store single-precision floating point numbers.
Double >> Double is used to store double-precision floating point numbers.

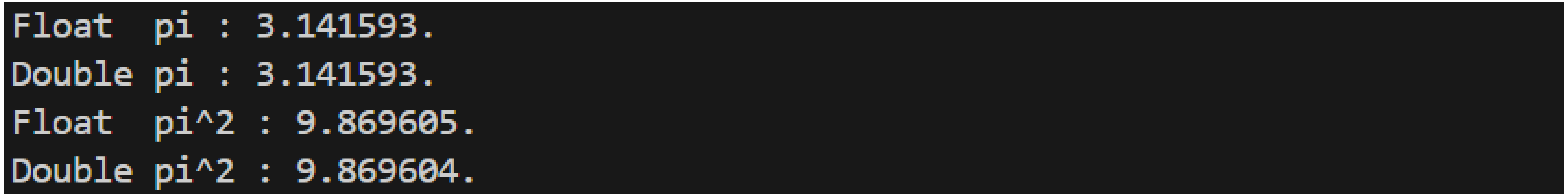
%f >> Used to output ‘float’ number.
%lf >> Used to output ‘double’ number.

printf : output float number



```c
float pi_f = 3.14159265359;     //單精度浮點數
double pi_d = 3.14159265359;    //雙精度浮點數  精度、範圍較高
printf("Float  pi : %f.\n" , pi_f );    //預設輸出小數點後6位 ， 最後會自己四捨五入
printf("Double pi : %lf.\n" , pi_d);
printf("Float  pi^2 : %f.\n" , pi_f * pi_f);
printf("Double pi^2 : %lf.\n", pi_d * pi_d);
```

Careful!!!



```
Float  pi : 3.141593.
Double pi : 3.141593.
Float  pi^2 : 9.869605.
Double pi^2 : 9.869604.
```

printf : output floating number

%.nf >> Used to round the floating-point number to n decimal places. ‘n’ means how many digits to show after the decimal point .

%.f >> Rounds the number to 0 decimal places. (You can think that ‘n’ is 0 .)

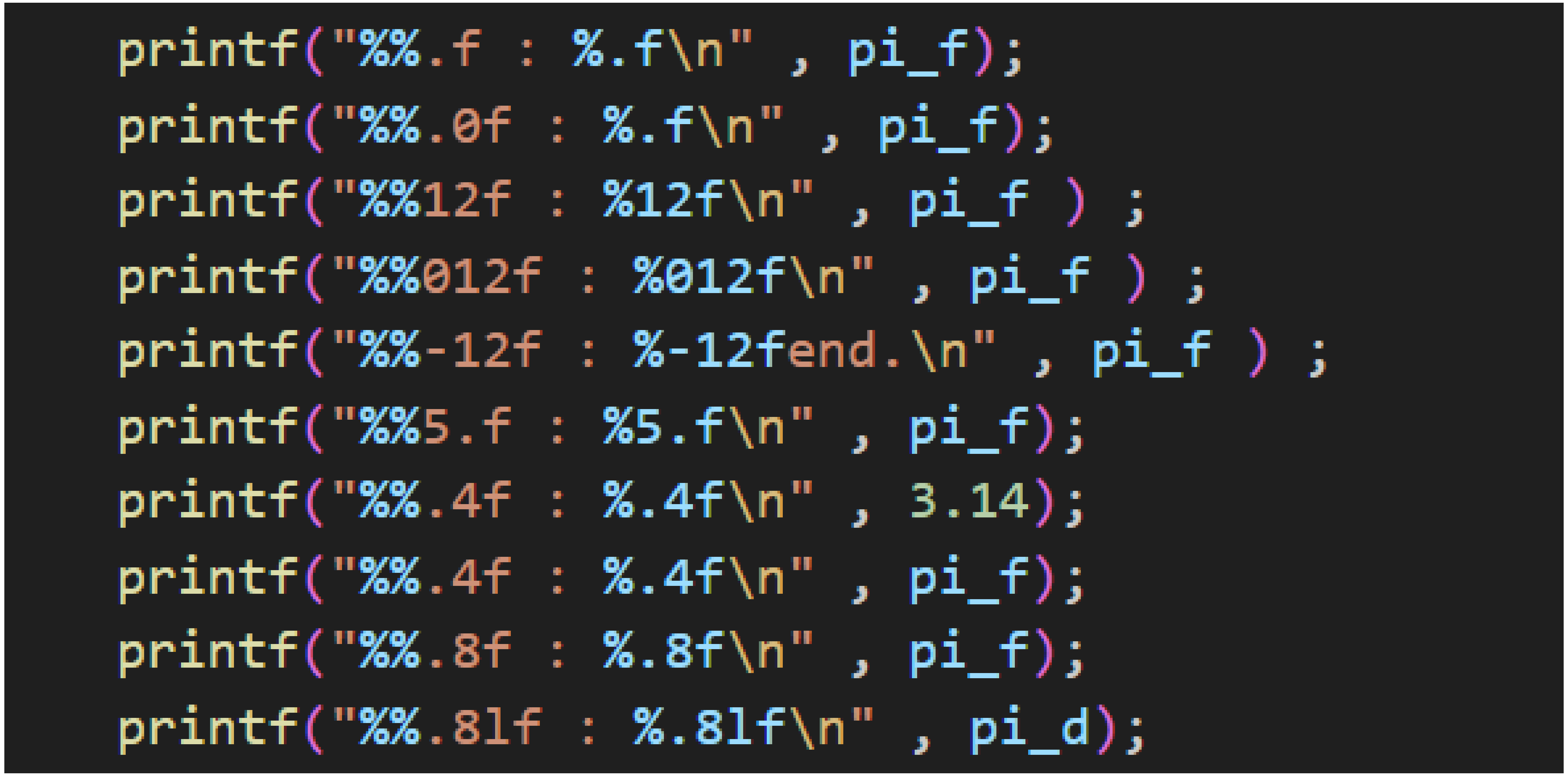
printf : output floating number

%nf >> To print the number with a total of at least ‘n’ space covered. (number at right side .)

%0nf >> Same as above , except pre-padding ‘0’.

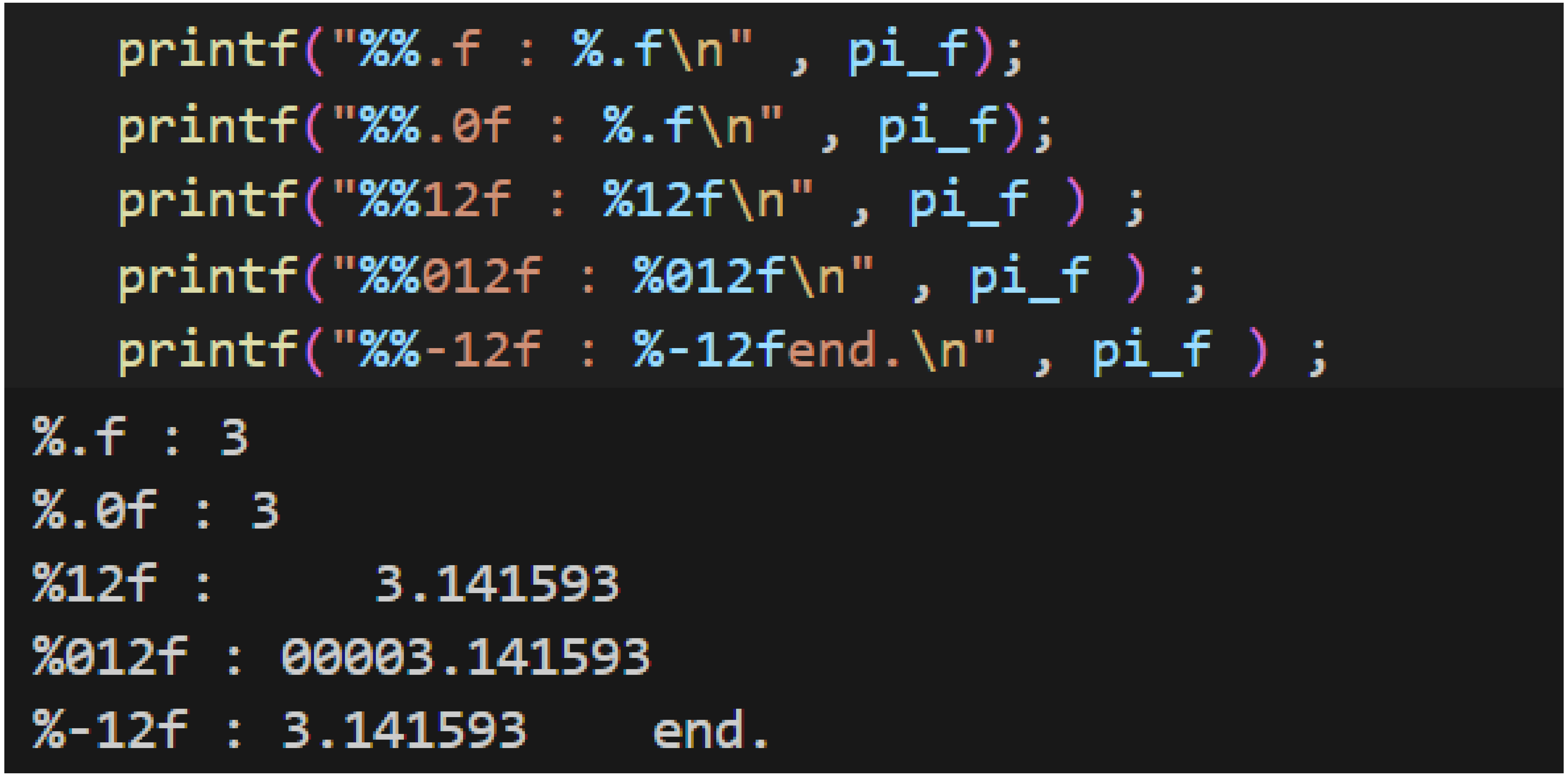
%-nf >> To print the number with a total of at least ‘n’ space covered. (number at left side .)

printf : output float number



```
printf("%%.f : %.f\n" , pi_f);
printf("%%.0f : %.f\n" , pi_f);
printf("%%12f : %12f\n" , pi_f ) ;
printf("%%012f : %012f\n" , pi_f ) ;
printf("%%-12f : %-12fend.\n" , pi_f ) ;
printf("%%5.f : %5.f\n" , pi_f);
printf("%%.4f : %.4f\n" , 3.14);
printf("%%.4f : %.4f\n" , pi_f);
printf("%%.8f : %.8f\n" , pi_f);
printf("%%.8lf : %.8lf\n" , pi_d);
```

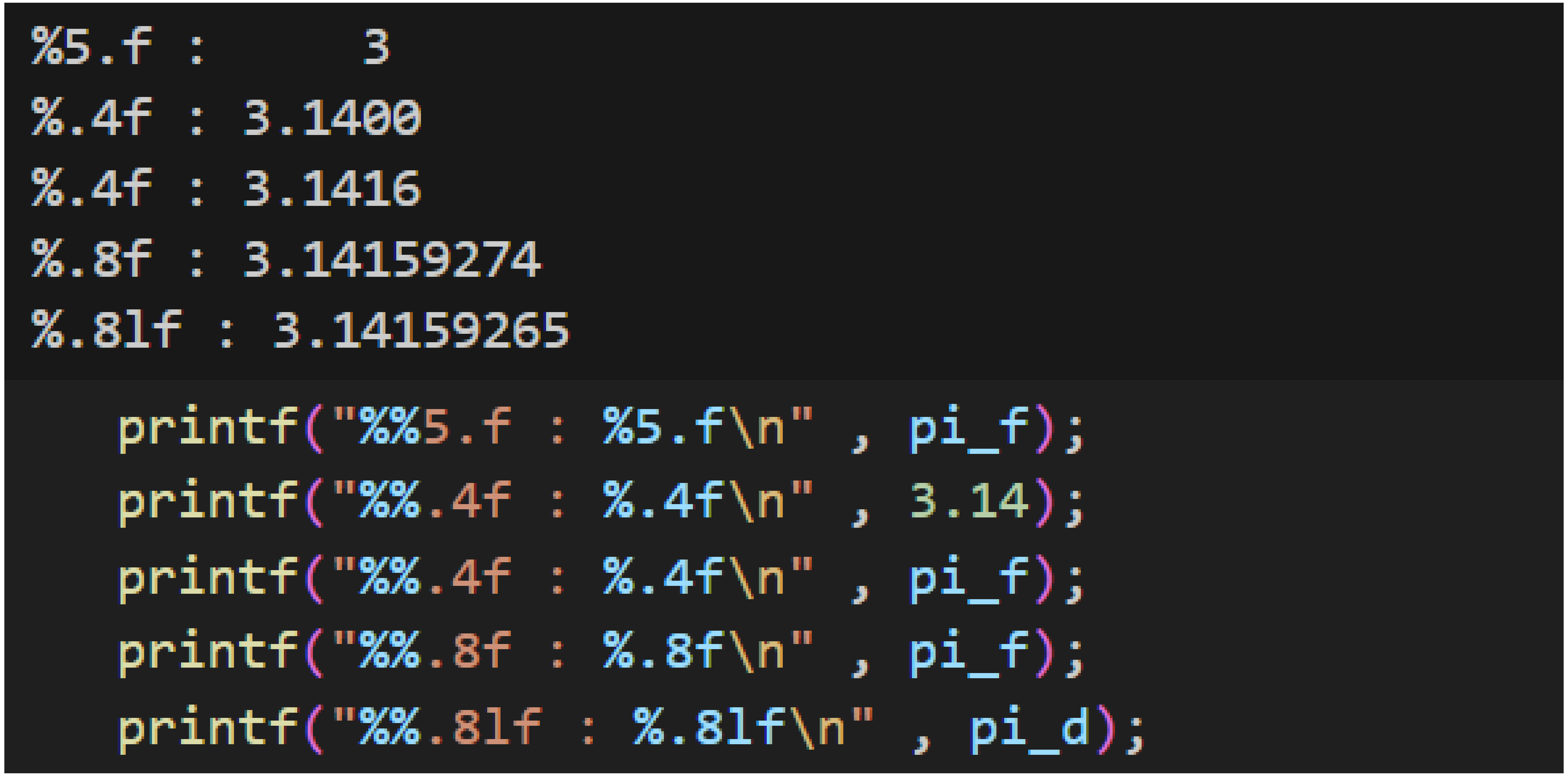
printf : output float number



```
    printf("%%.f : %.f\n" , pi_f);
    printf("%%.0f : %.f\n" , pi_f);
    printf("%%12f : %12f\n" , pi_f ) ;
    printf("%%012f : %012f\n" , pi_f ) ;
    printf("%%-12f : %-12fend.\n" , pi_f ) ;
```

```
%.f : 3
%.0f : 3
%12f :     3.141593
%012f : 00003.141593
%-12f : 3.141593    end.
```

printf : output float number



```
%5.f :     3
%.4f : 3.1400
%.4f : 3.1416
%.8f : 3.14159274
%.8lf : 3.14159265
```

```
   printf("%%5.f : %5.f\n" , pi_f);
   printf("%%.4f : %.4f\n" , 3.14);
   printf("%%.4f : %.4f\n" , pi_f);
   printf("%%.8f : %.8f\n" , pi_f);
   printf("%%.8lf : %.8lf\n" , pi_d);
```

printf : output char and string

%c >> Output character.

%nc >> To print the character with a total of at least ‘n’ space covered. (character at right side .)

%-c >> character at left side.

printf : output char and string

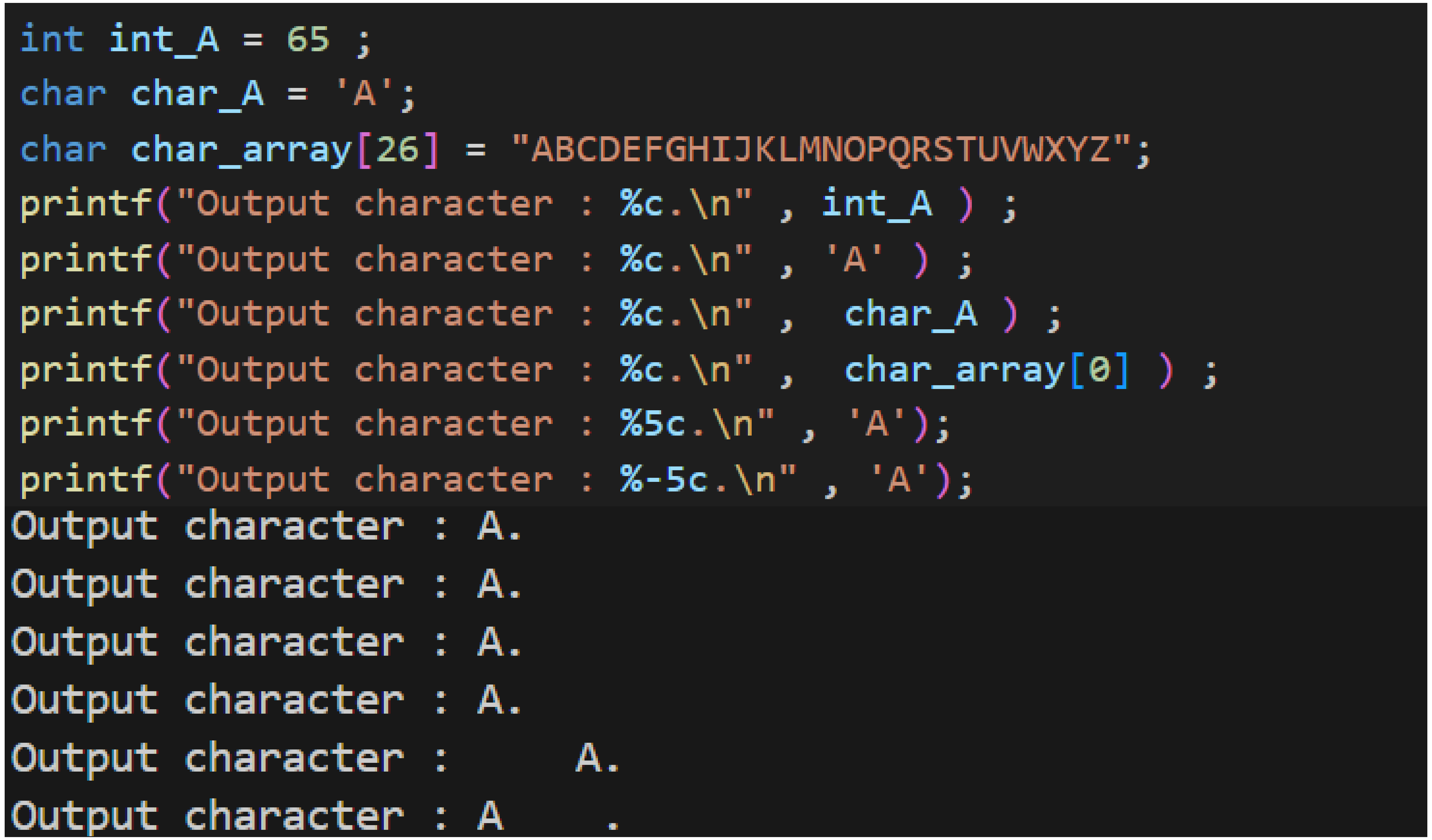
%s >> Output string (something like char array) .

%-s >> string at left

%ns >> To print the string with a total of at least ‘n’ space covered. (string at right side .)

%.ns >> Output at most ‘n’ character in string.

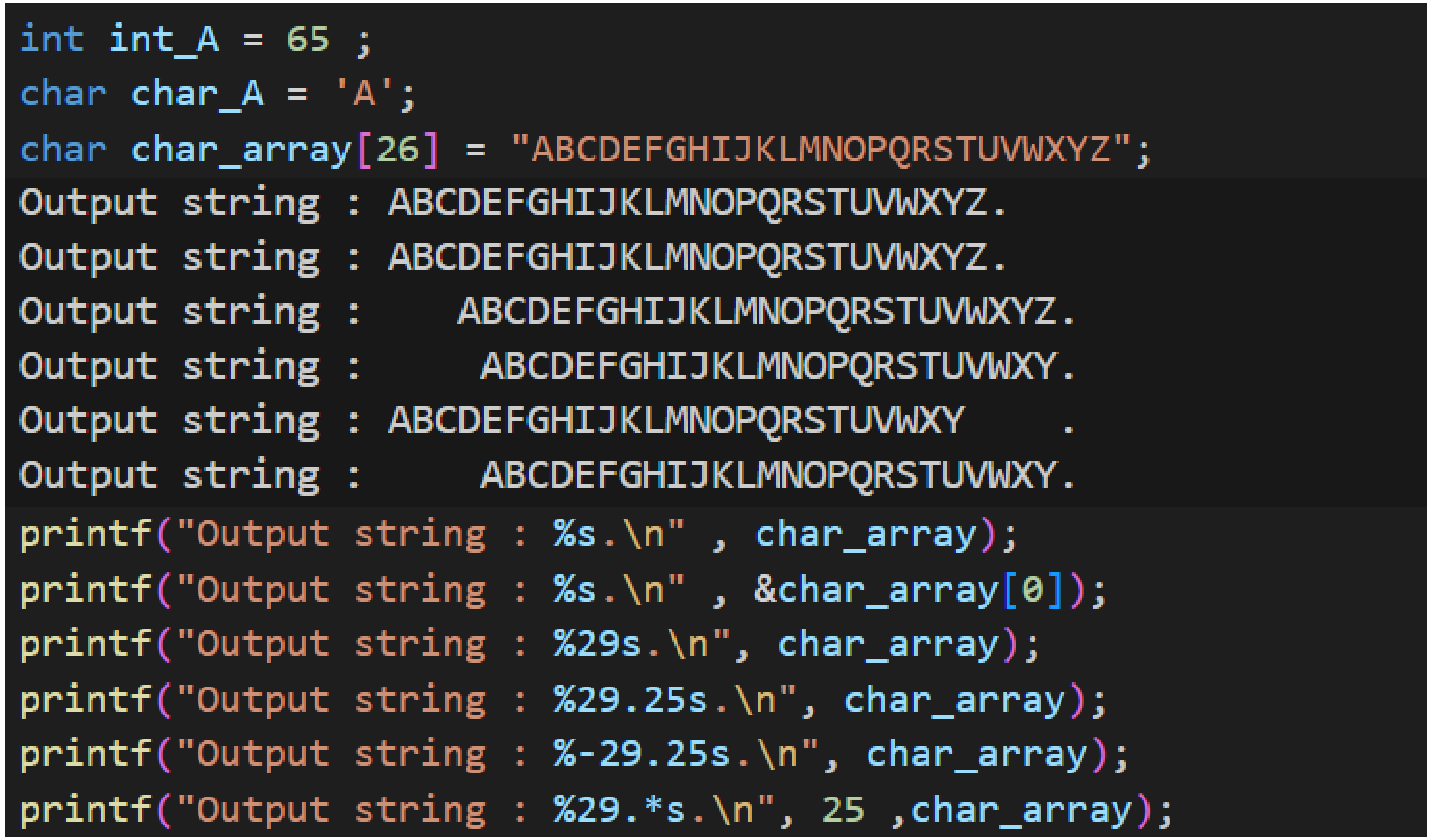
printf : output char and string



```c
int int_A = 65 ;
char char_A = 'A';
char char_array[26] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
printf("Output character : %c.\n" , int_A ) ;
printf("Output character : %c.\n" , 'A' ) ;
printf("Output character : %c.\n" ,  char_A ) ;
printf("Output character : %c.\n" ,  char_array[0] ) ;
printf("Output character : %5c.\n" , 'A');
printf("Output character : %-5c.\n" , 'A');
```

```
Output character : A.
Output character : A.
Output character : A.
Output character : A.
Output character :     A.
Output character : A    .
```

printf : output char and string



```c
int int_A = 65 ;
char char_A = 'A';
char char_array[26] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
```

```
Output string : ABCDEFGHIJKLMNOPQRSTUVWXYZ.
Output string : ABCDEFGHIJKLMNOPQRSTUVWXYZ.
Output string :    ABCDEFGHIJKLMNOPQRSTUVWXYZ.
Output string :     ABCDEFGHIJKLMNOPQRSTUVWXY.
Output string : ABCDEFGHIJKLMNOPQRSTUVWXY    .
Output string :     ABCDEFGHIJKLMNOPQRSTUVWXY.
```

```c
printf("Output string : %s.\n" , char_array);
printf("Output string : %s.\n" , &char_array[0]);
printf("Output string : %29s.\n", char_array);
printf("Output string : %29.25s.\n", char_array);
printf("Output string : %-29.25s.\n", char_array);
printf("Output string : %29.*s.\n", 25 ,char_array);
```

2. Basic input

scanf : the input syntax in C

Syntax : scanf("(%type)" , &variable);

Review all of type in C :

1. %d , %ld, %lld, %u , %lu , %llu
2. %f , %lf
3. %c
4. %s

scanf : the input syntax in C

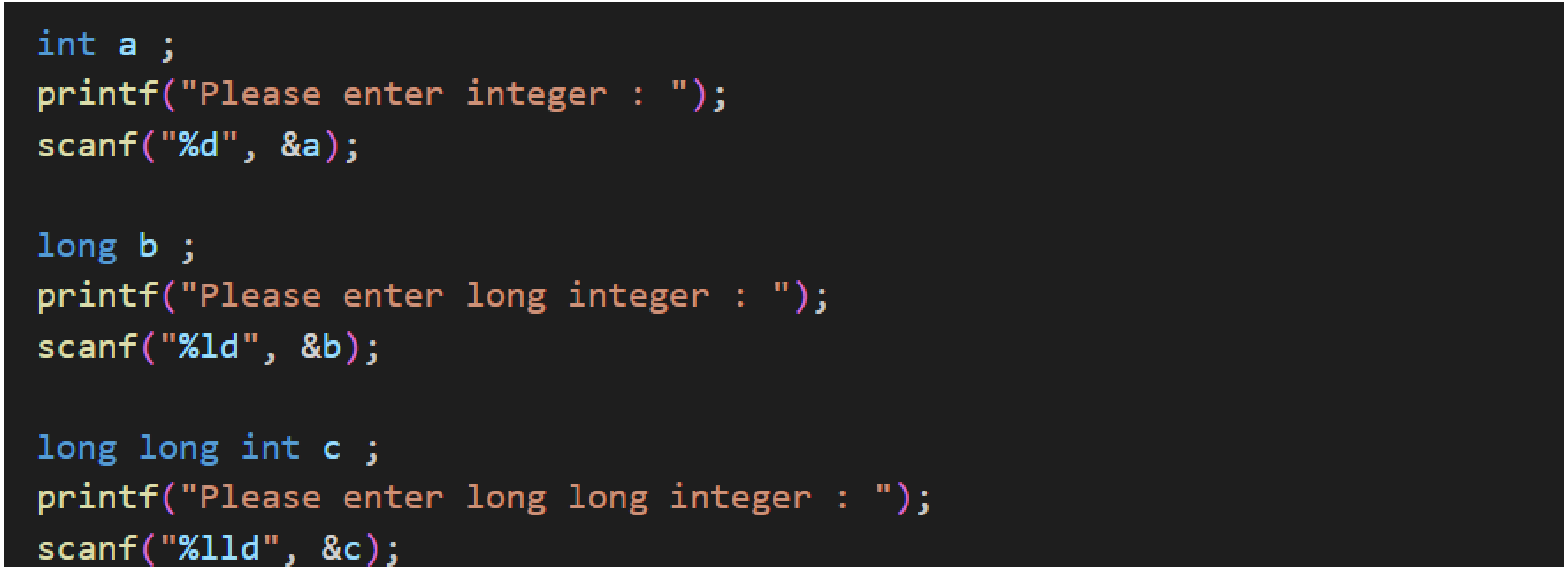
Syntax : scanf("(%type)" , &variable);

don't forget !!!

Review all of type in C :

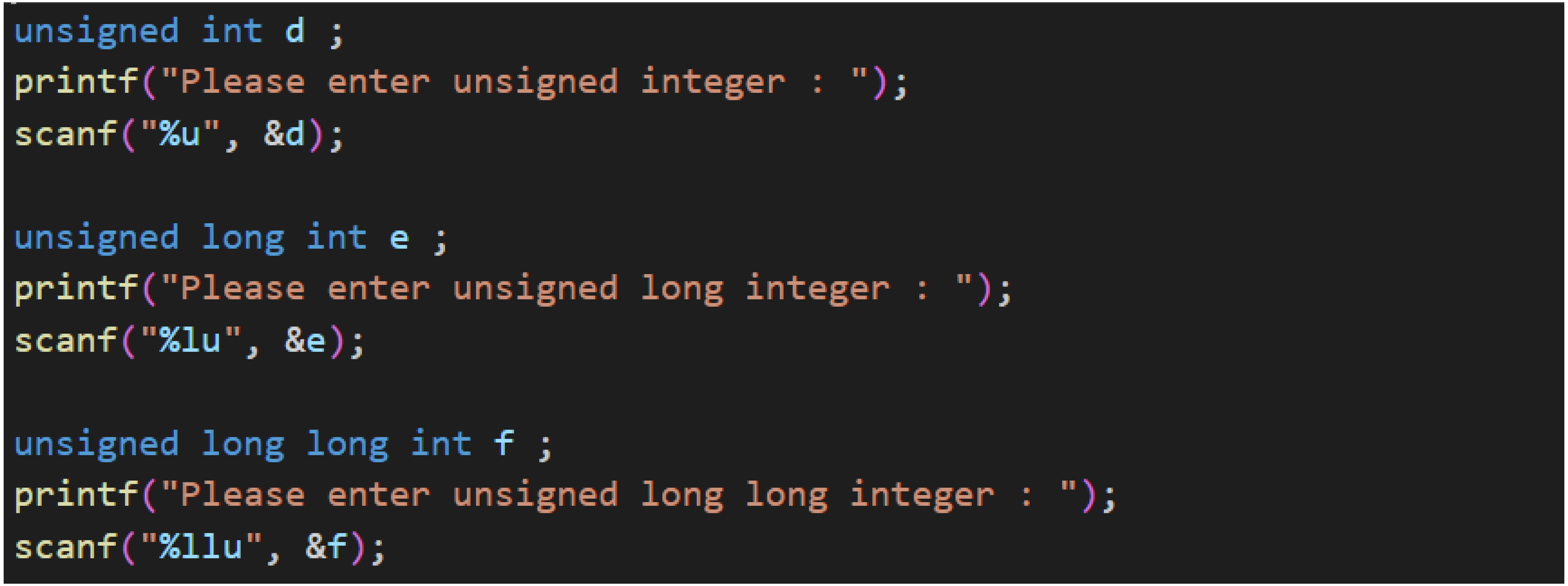
1. %d , %ld, %lld, %u , %lu , %llu
2. %f , %lf
3. %c
4. %s

Example for scanf



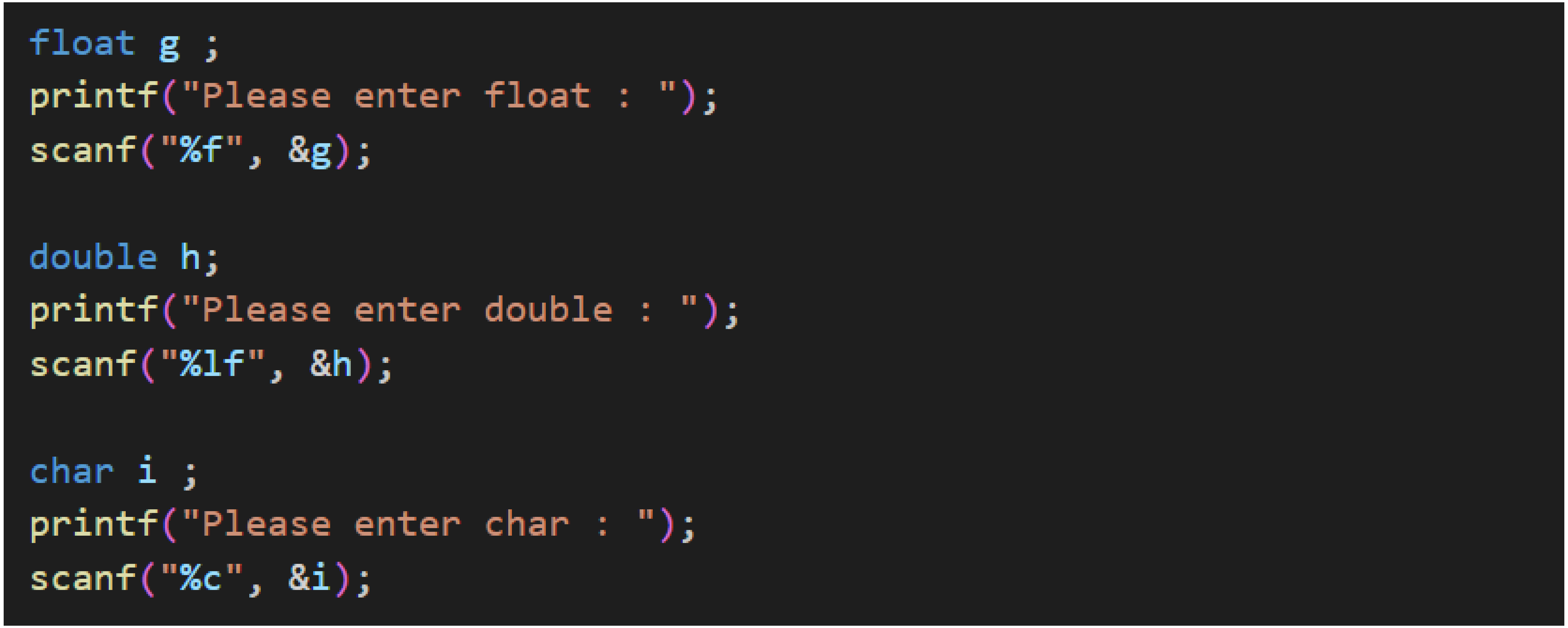
```
int a ;
printf("Please enter integer : ");
scanf("%d", &a);

long b ;
printf("Please enter long integer : ");
scanf("%ld", &b);

long long int c ;
printf("Please enter long long integer : ");
scanf("%lld", &c);
```

Example for scanf



```c
unsigned int d ;
printf("Please enter unsigned integer : ");
scanf("%u", &d);

unsigned long int e ;
printf("Please enter unsigned long integer : ");
scanf("%lu", &e);

unsigned long long int f ;
printf("Please enter unsigned long long integer : ");
scanf("%llu", &f);
```

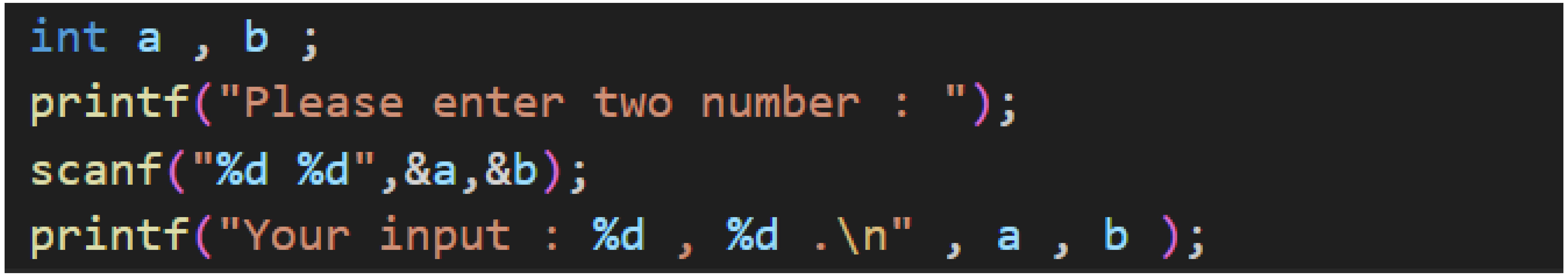
Example for scanf



```c
float g ;
printf("Please enter float : ");
scanf("%f", &g);

double h;
printf("Please enter double : ");
scanf("%lf", &h);

char i ;
printf("Please enter char : ");
scanf("%c", &i);
```

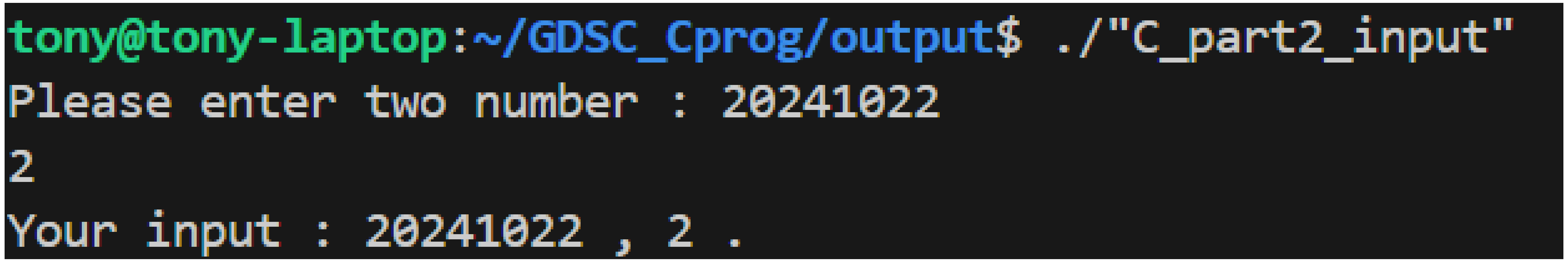
scanf : multiple input (single scanf)

Use single scanf to input , you can do this ...



```c
int a , b ;
printf("Please enter two number : ");
scanf("%d %d",&a,&b);
printf("Your input : %d , %d .\n" , a , b );
```

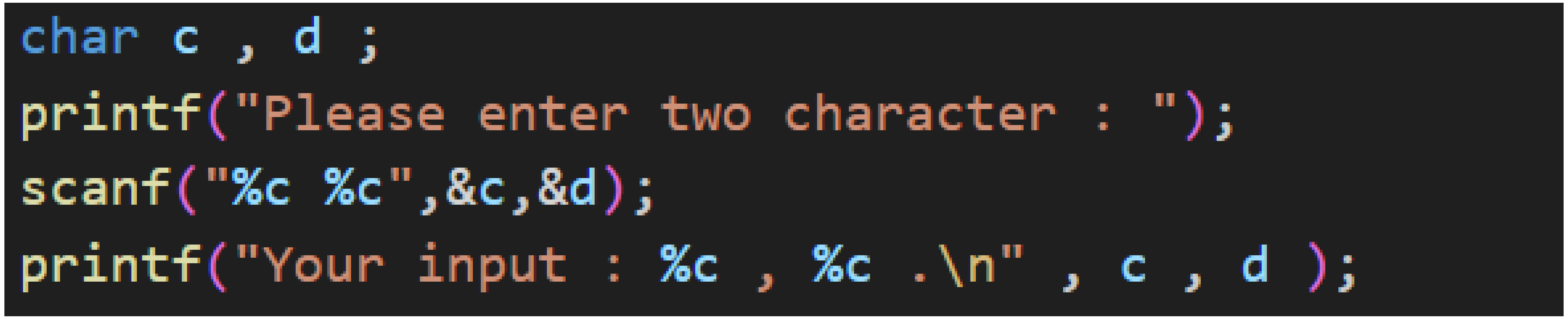
output



```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part2_input"
Please enter two number : 20241022
2
Your input : 20241022 , 2 .
```

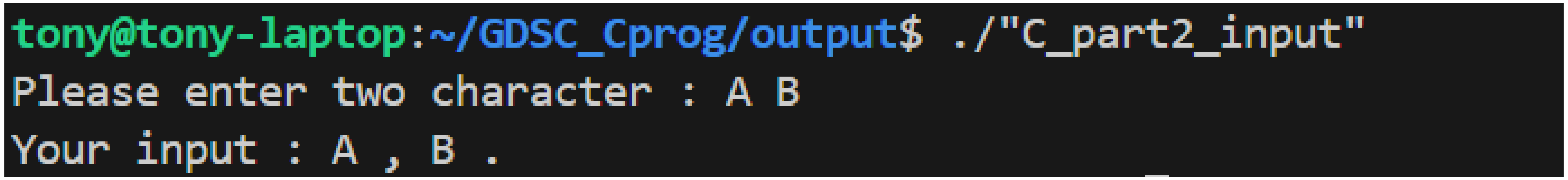
scanf : multiple input(single scanf)

For other situation , you also can use this method for multiple input



```c
char c , d ;
printf("Please enter two character : ");
scanf("%c %c",&c,&d);
printf("Your input : %c , %c .\n" , c , d );
```

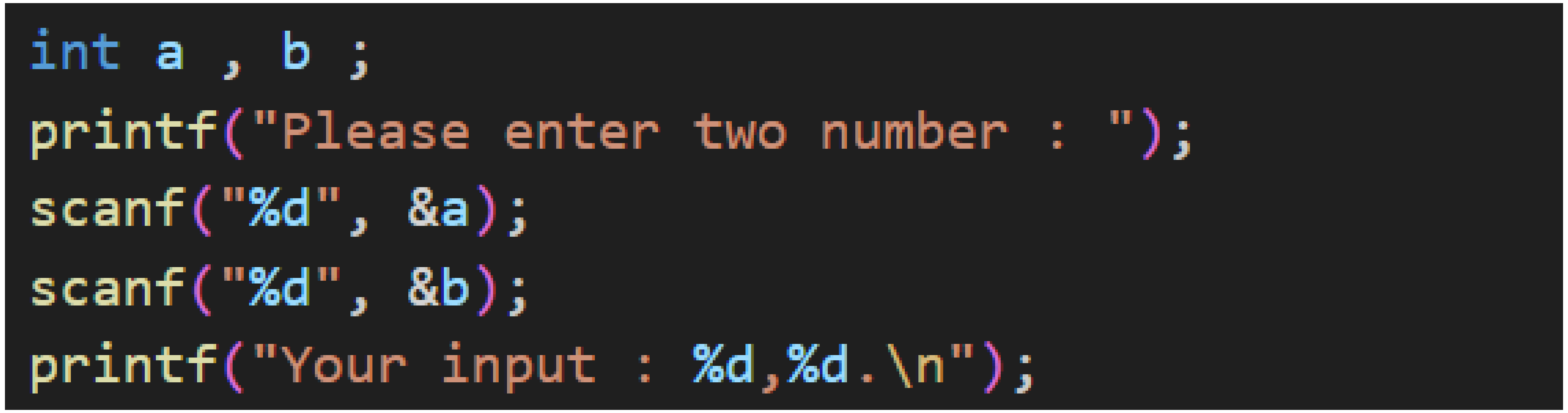
output



```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part2_input"
Please enter two character : A B
Your input : A , B .
```

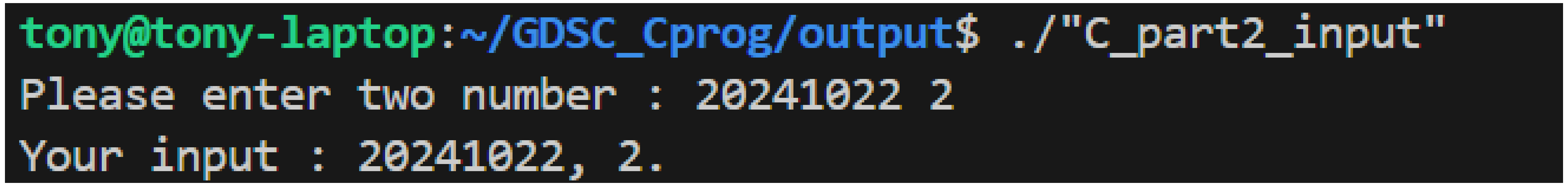
scanf : multiple input(multiple scanf)

Use multiple scanf need to be careful , it can work for integer.



```c
int a , b ;
printf("Please enter two number : ");
scanf("%d", &a);
scanf("%d", &b);
printf("Your input : %d,%d.\n");
```

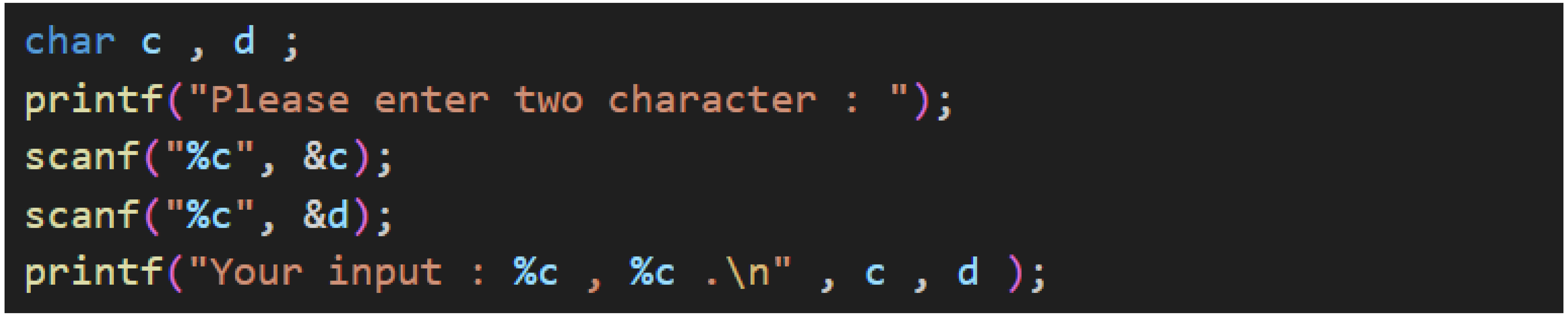
output



```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part2_input"
Please enter two number : 20241022 2
Your input : 20241022, 2.
```

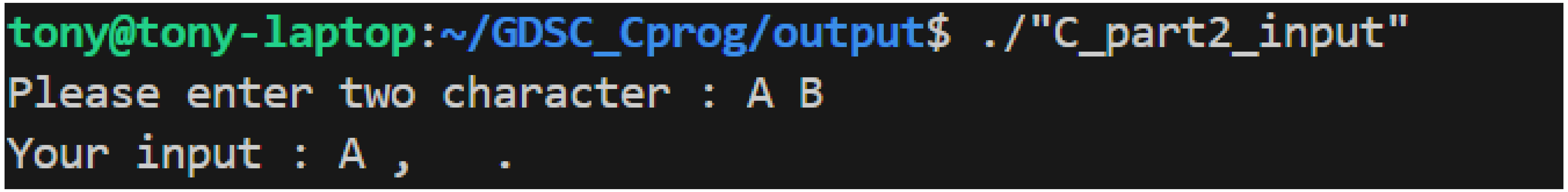
scanf : multiple input(multiple scanf)

But if you need to input character , careful to use mutiple scanf



```c
char c , d ;
printf("Please enter two character : ");
scanf("%c", &c);
scanf("%c", &d);
printf("Your input : %c , %c .\n" , c , d );
```

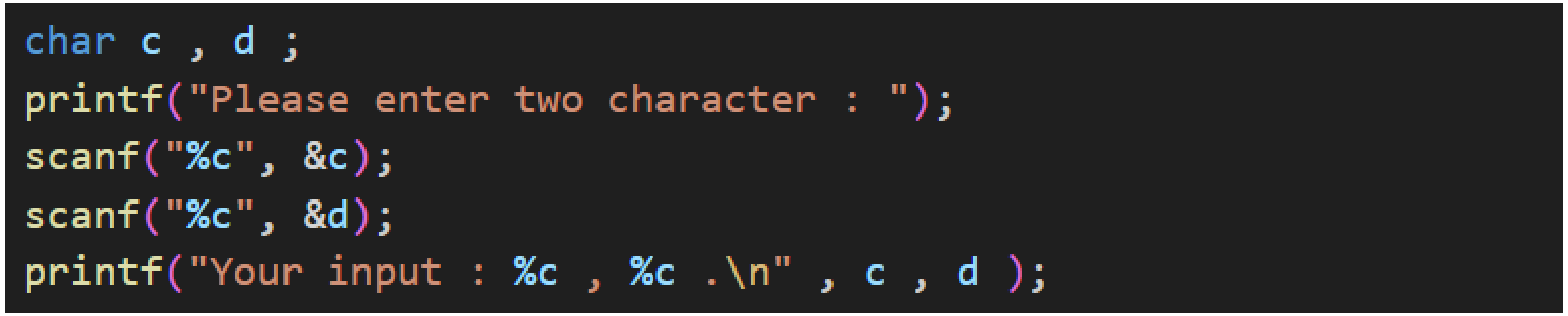
output



```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part2_input"
Please enter two character : A B
Your input : A ,   .
```

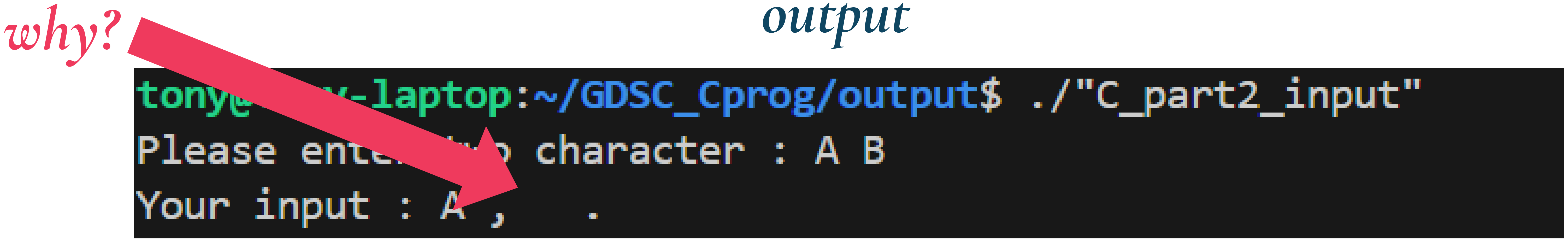
scanf : multiple input(multiple scanf)

But if you need to input character , careful to use mutiple scanf



```
char c , d ;
printf("Please enter two character : ");
scanf("%c", &c);
scanf("%c", &d);
printf("Your input : %c , %c .\n" , c , d );
```

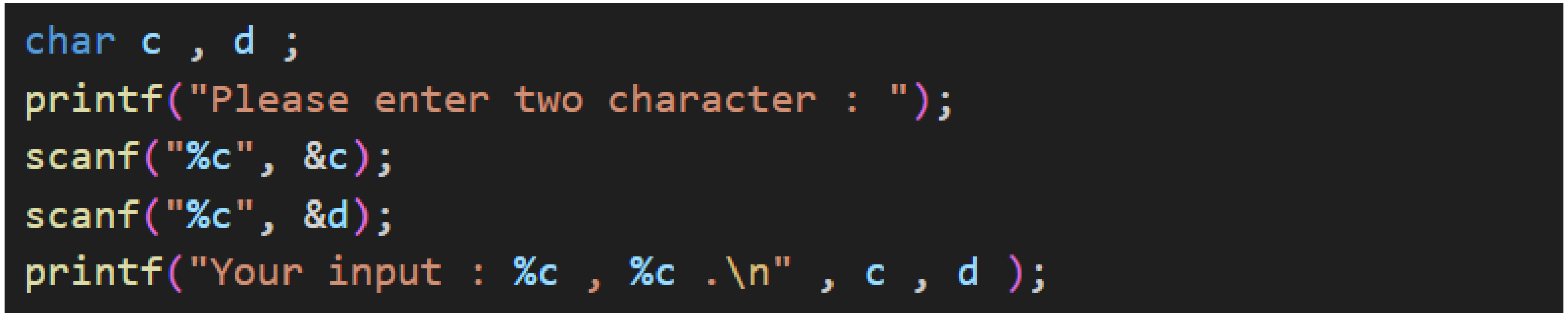
output



```
tony@[illegible]-laptop:~/GDSC_Cprog/output$ ./"C_part2_input"
Please enter two character : A B
Your input : A ,   .
```

scanf : multiple input(multiple scanf)

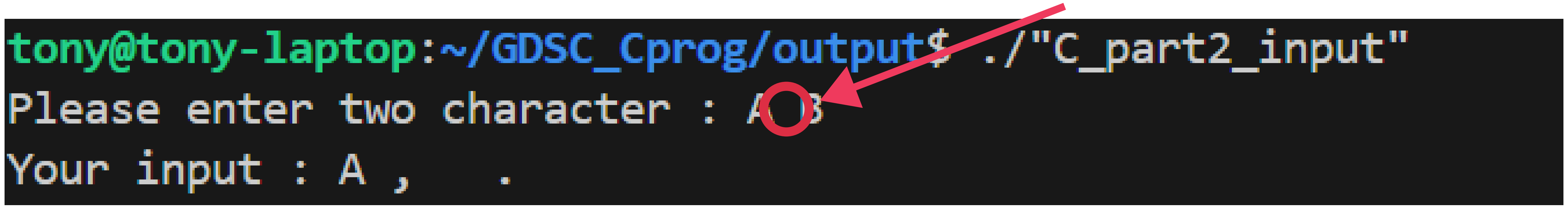
But if you need to input character , careful to use mutiple scanf



```c
char c , d ;
printf("Please enter two character : ");
scanf("%c", &c);
scanf("%c", &d);
printf("Your input : %c , %c .\n" , c , d );
```

output

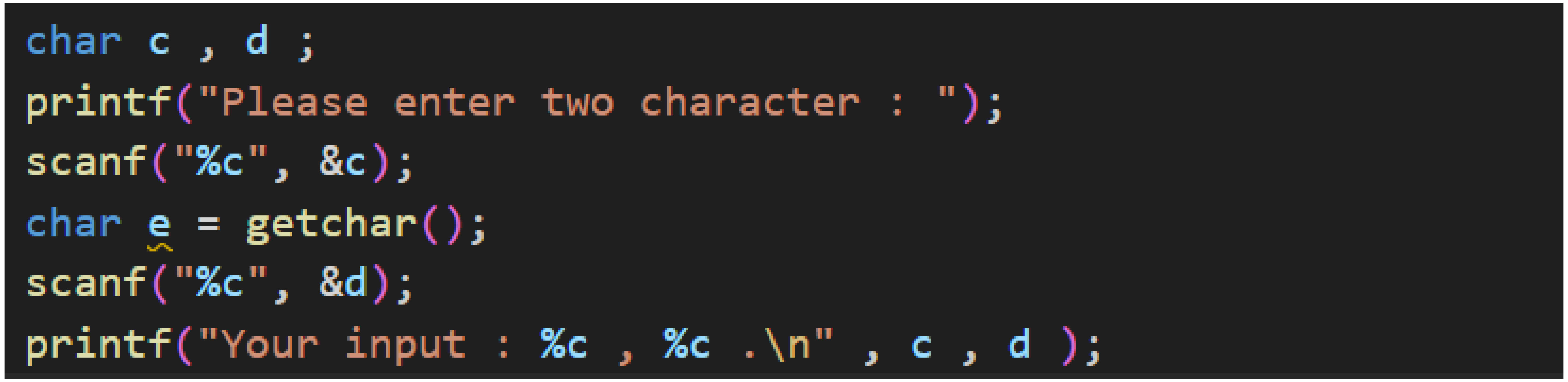
There is a space!!! how to solve?



```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part2_input"
Please enter two character : A B
Your input : A ,   .
```

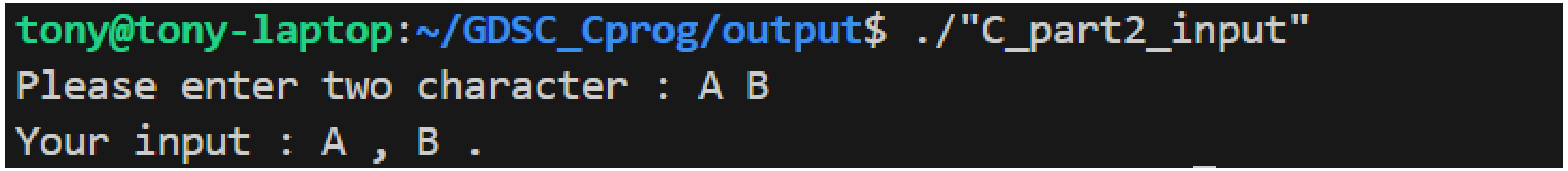
scanf : multiple input(multiple scanf)

Use getchar to solve this , or add one scanf



```c
char c , d ;
printf("Please enter two character : ");
scanf("%c", &c);
char e = getchar();
scanf("%c", &d);
printf("Your input : %c , %c .\n" , c , d );
```

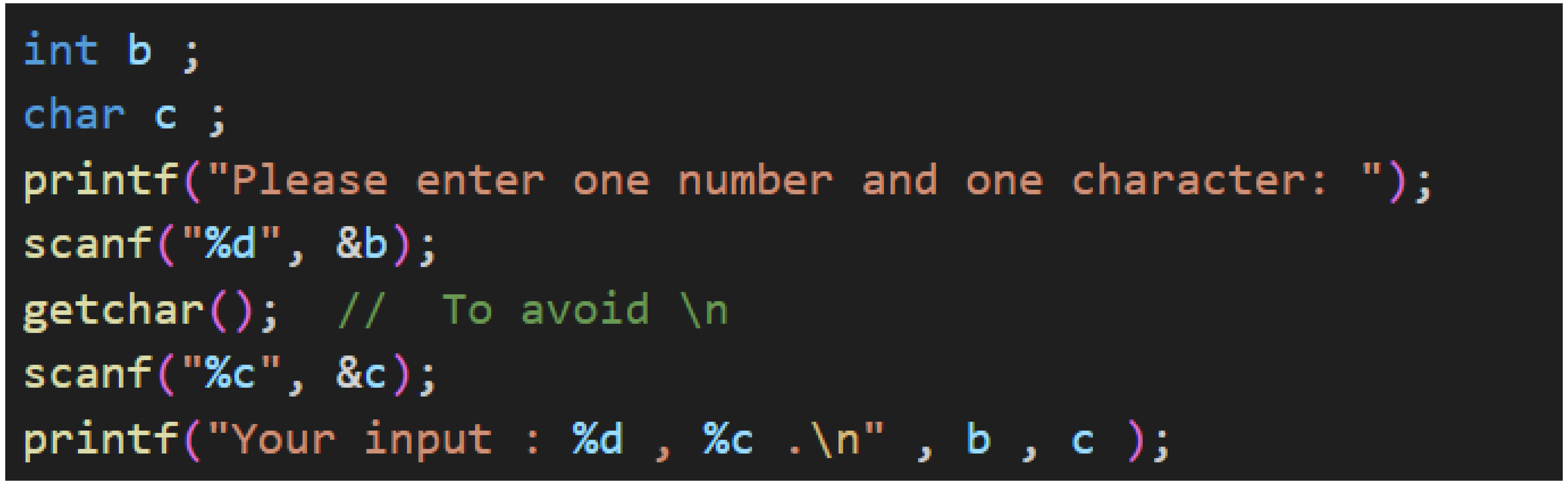
output



```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part2_input"
Please enter two character : A B
Your input : A , B .
```

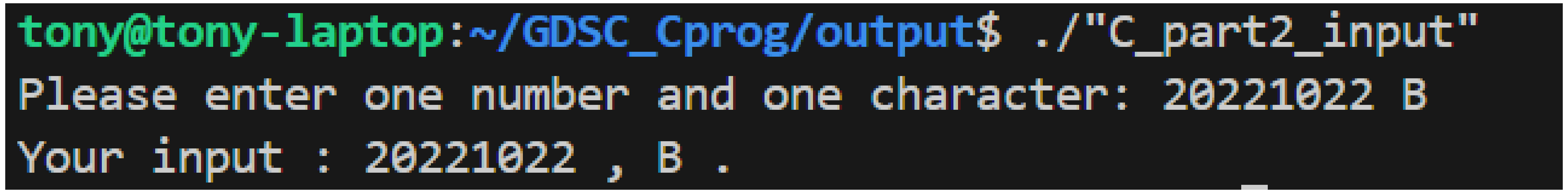
scanf : multiple input(multiple scanf)

getchar() can use less variable.



```
int b ;
char c ;
printf("Please enter one number and one character: ");
scanf("%d", &b);
getchar();  //  To avoid \n
scanf("%c", &c);
printf("Your input : %d , %c .\n" , b , c );
```

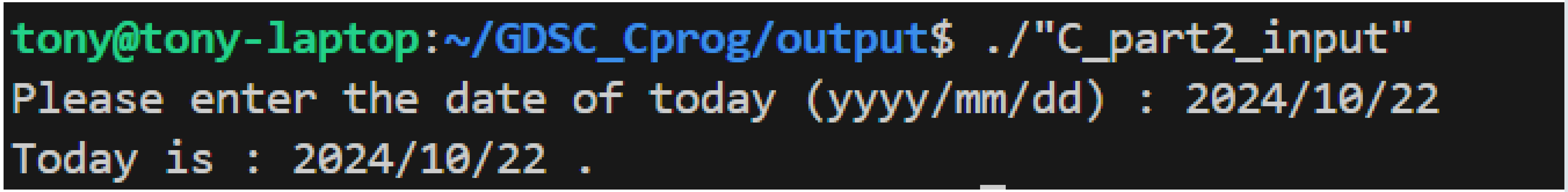
output



```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part2_input"
Please enter one number and one character: 20221022 B
Your input : 20221022 , B .
```

scanf : specific input format (date)

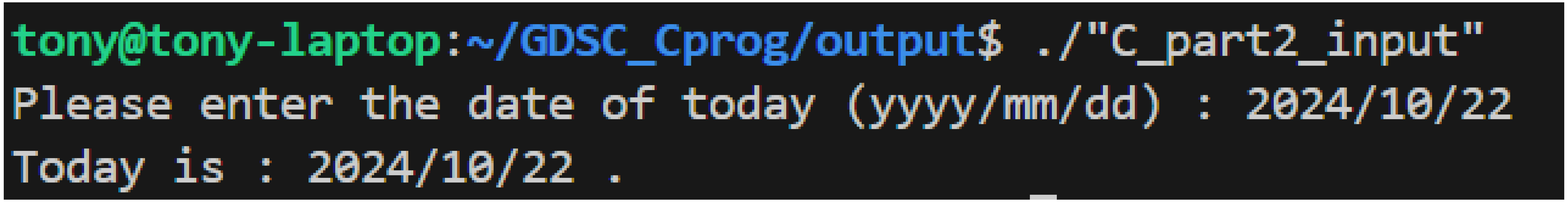
How to do this ?



```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part2_input"
Please enter the date of today (yyyy/mm/dd) : 2024/10/22
Today is : 2024/10/22 .
```

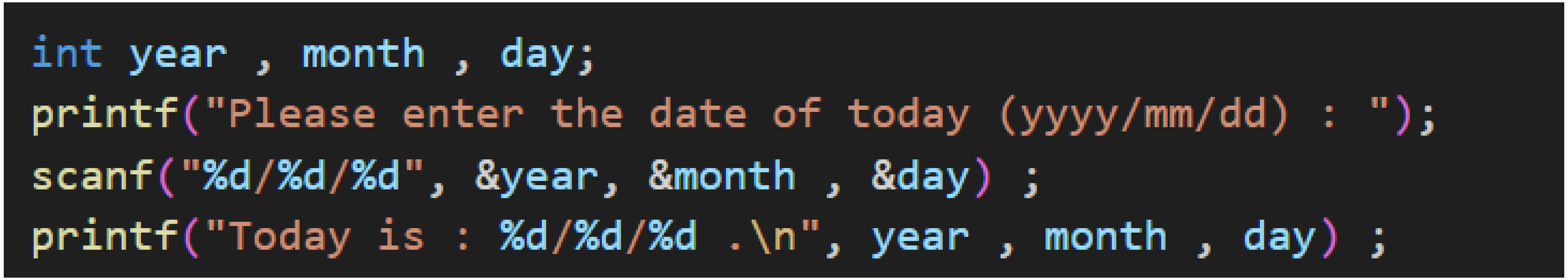
scanf : specific input format (date)

How to do this ?



```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part2_input"
Please enter the date of today (yyyy/mm/dd) : 2024/10/22
Today is : 2024/10/22 .
```

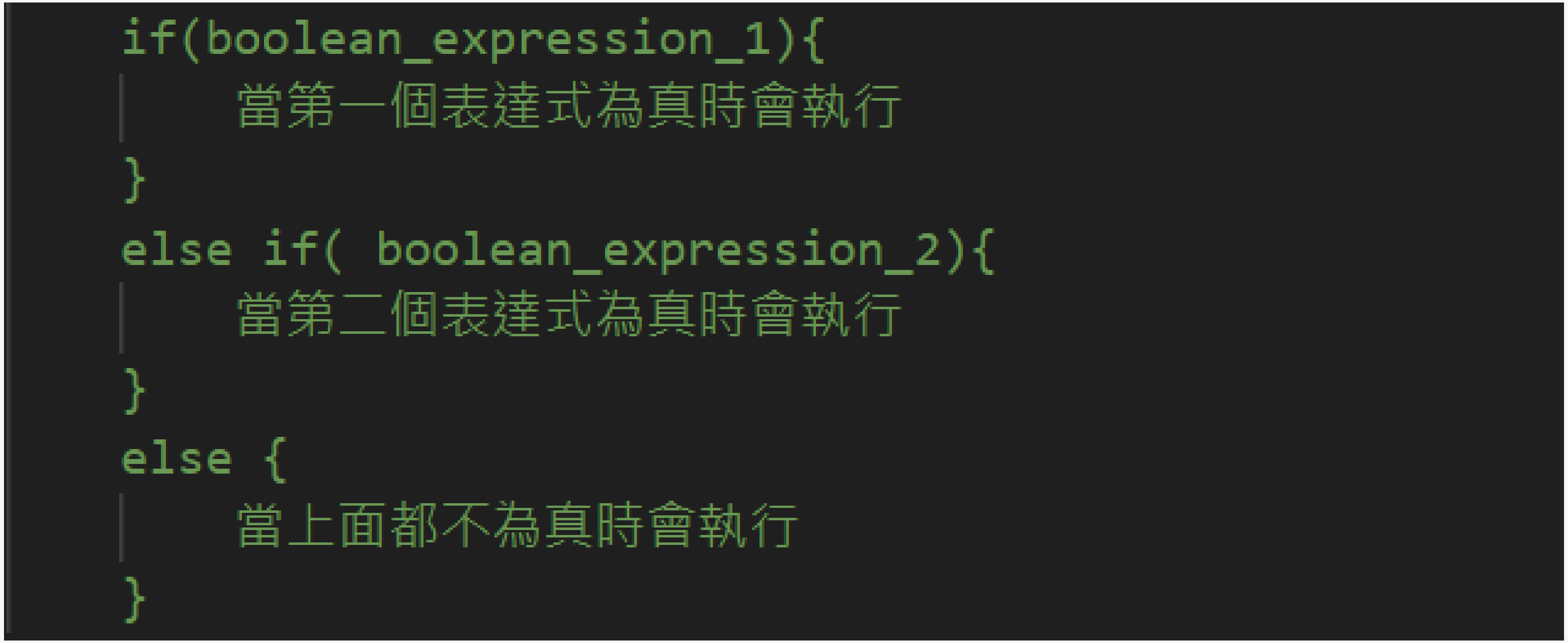
you can ...



```c
int year , month , day;
printf("Please enter the date of today (yyyy/mm/dd) : ");
scanf("%d/%d/%d", &year, &month , &day) ;
printf("Today is : %d/%d/%d .\n", year , month , day) ;
```

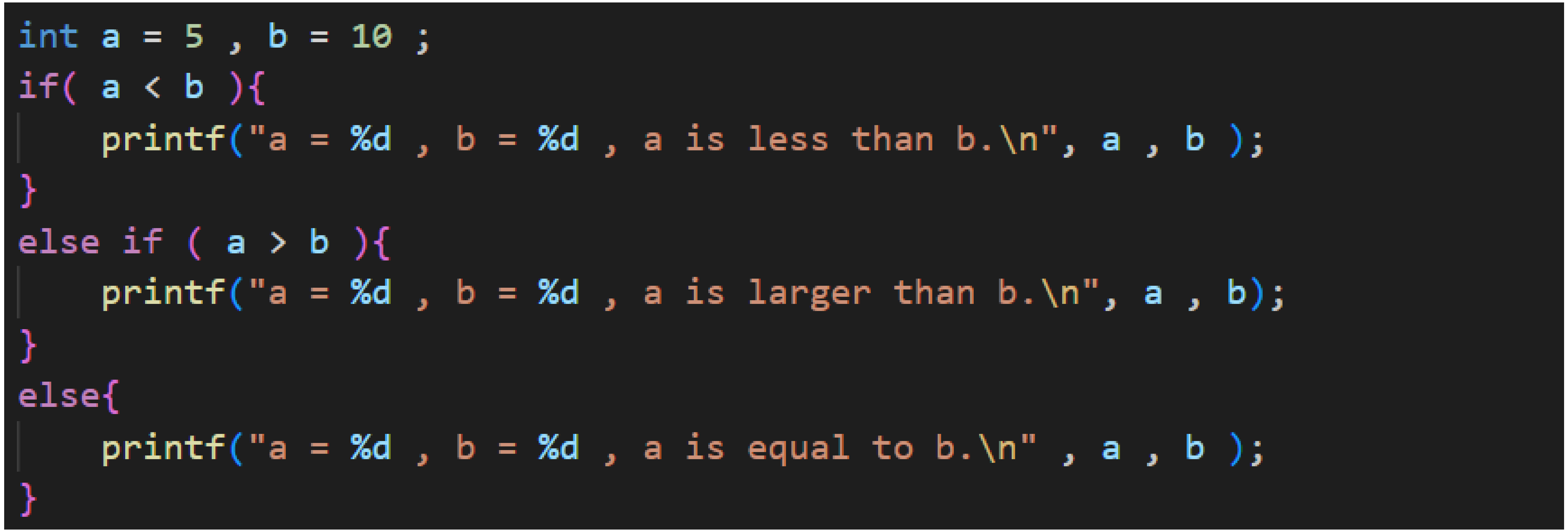
3. if / else

if / else : the syntax in C

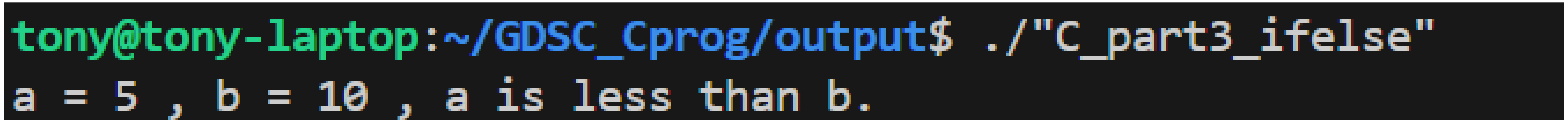


```
if(boolean_expression_1){
        當第一個表達式為真時會執行
}
else if( boolean_expression_2){
        當第二個表達式為真時會執行
}
else {
        當上面都不為真時會執行
}
```

if / else : the syntax in C

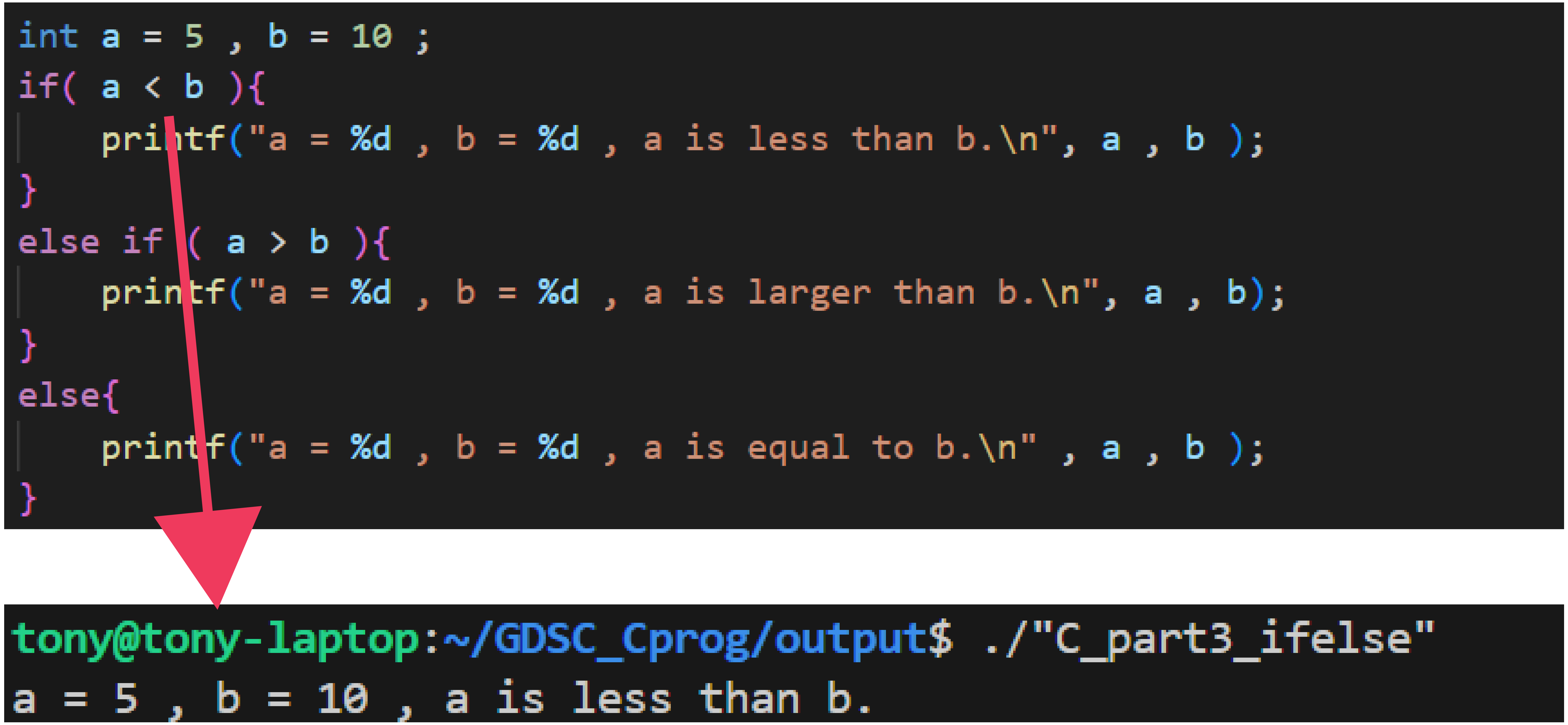


```c
int a = 5 , b = 10 ;
if( a < b ){
    printf("a = %d , b = %d , a is less than b.\n", a , b );
}
else if ( a > b ){
    printf("a = %d , b = %d , a is larger than b.\n", a , b);
}
else{
    printf("a = %d , b = %d , a is equal to b.\n" , a , b );
}
```



```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part3_ifelse"
a = 5 , b = 10 , a is less than b.
```

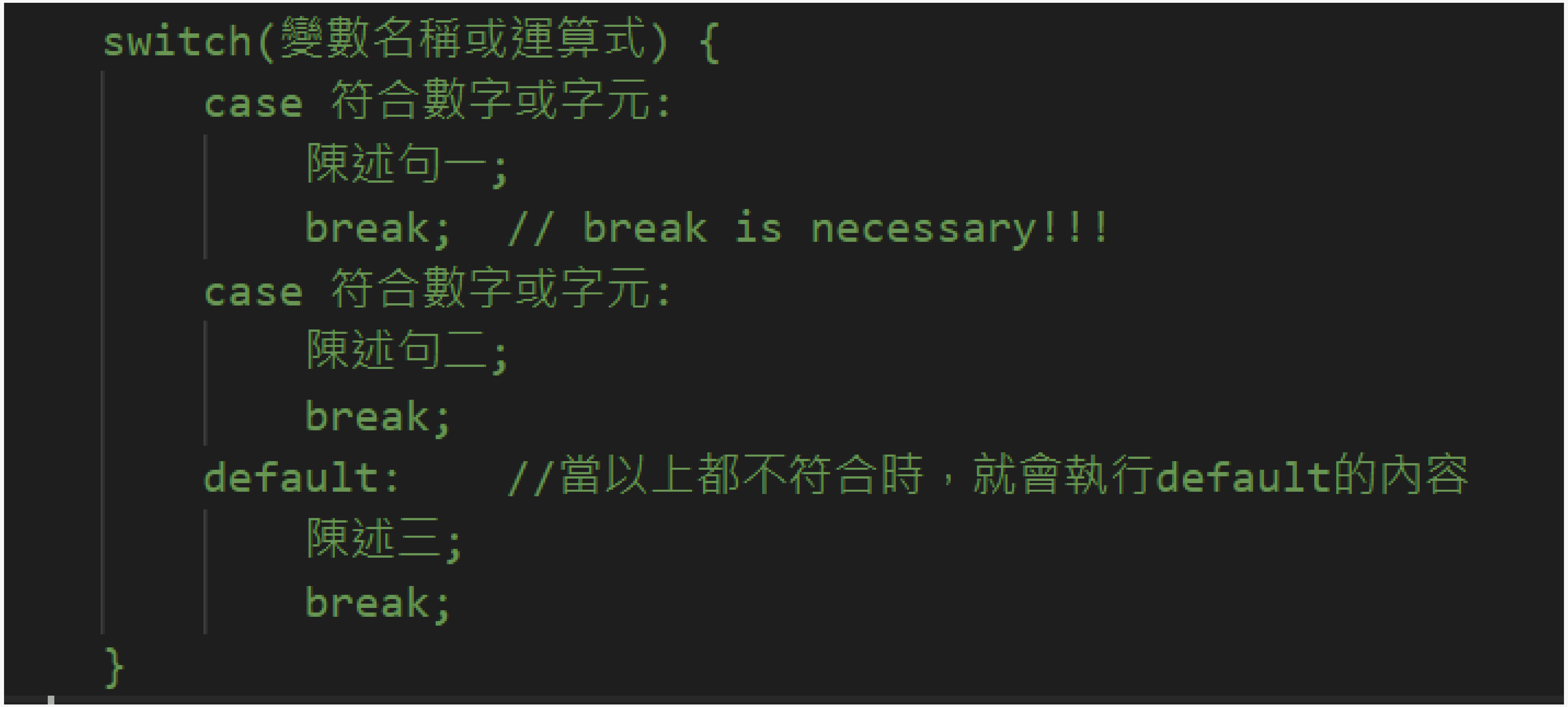
if / else : the syntax in C



```c
int a = 5 , b = 10 ;
if( a < b ){
    printf("a = %d , b = %d , a is less than b.\n", a , b );
}
else if ( a > b ){
    printf("a = %d , b = %d , a is larger than b.\n", a , b);
}
else{
    printf("a = %d , b = %d , a is equal to b.\n" , a , b );
}
```

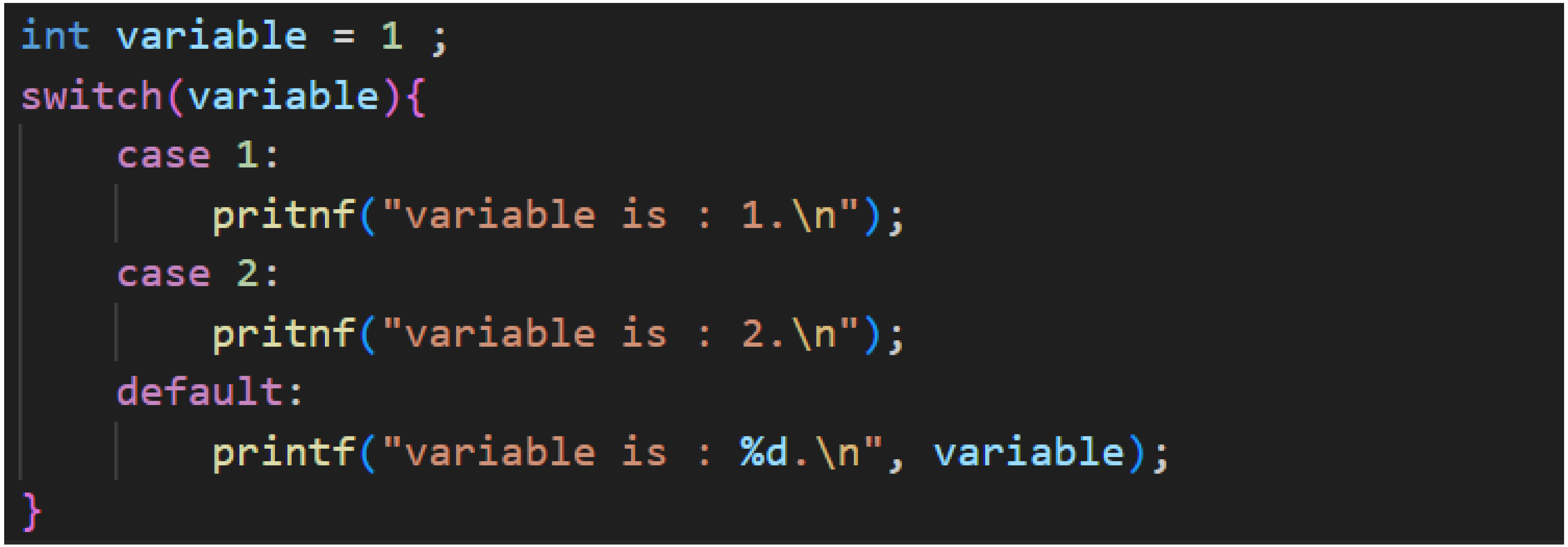
```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part3_ifelse"
a = 5 , b = 10 , a is less than b.
```

Switch : the syntax in C

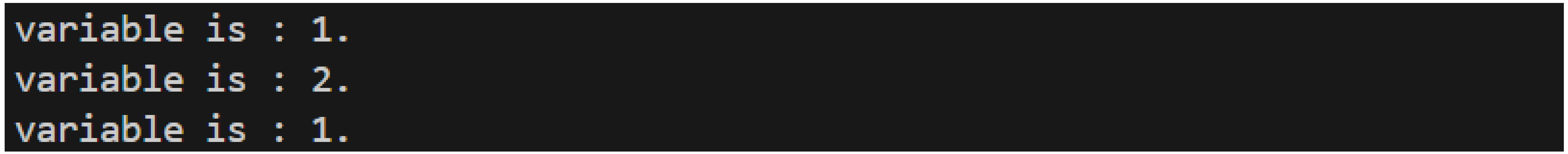


```
switch(變數名稱或運算式) {
    case 符合數字或字元:
        陳述句一;
        break;  // break is necessary!!!
    case 符合數字或字元:
        陳述句二;
        break;
    default:    //當以上都不符合時，就會執行default的內容
        陳述三;
        break;
}
```

Switch : the syntax in C

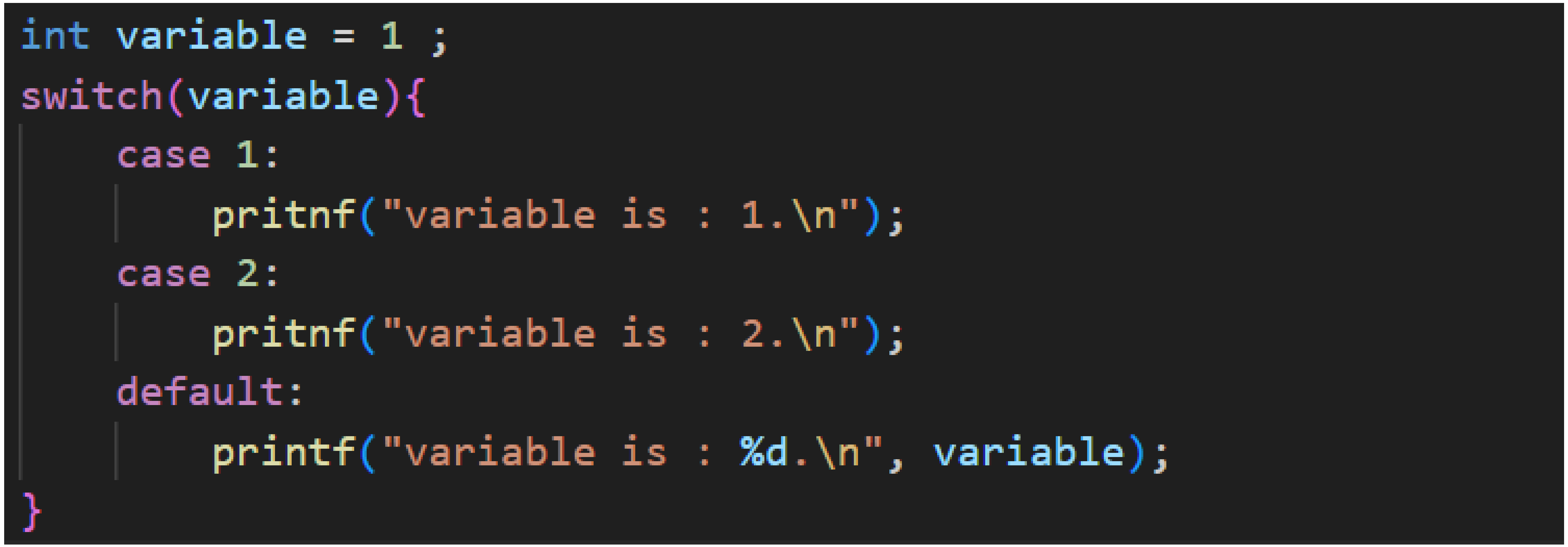


```c
int variable = 1 ;
switch(variable){
    case 1:
        pritnf("variable is : 1.\n");
    case 2:
        pritnf("variable is : 2.\n");
    default:
        printf("variable is : %d.\n", variable);
}
```

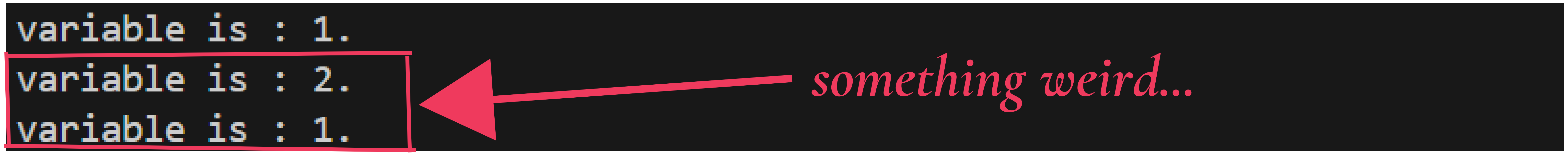


```
variable is : 1.
variable is : 2.
variable is : 1.
```

Switch : the syntax in C

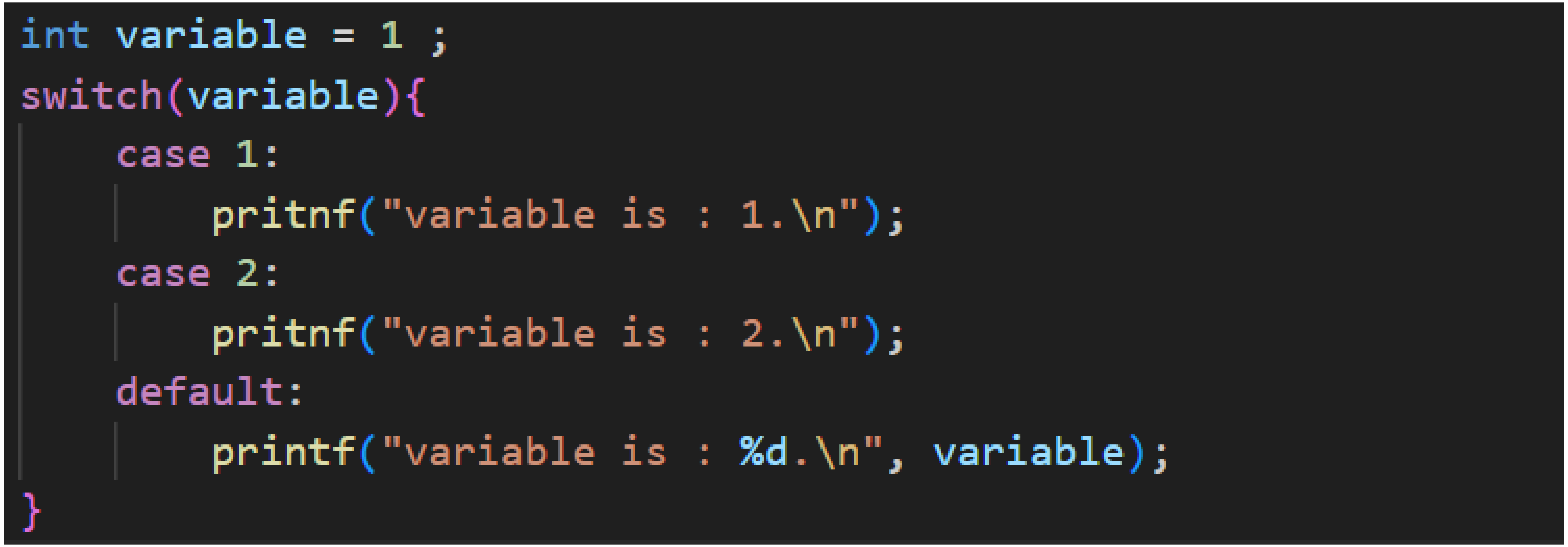


```c
int variable = 1 ;
switch(variable){
    case 1:
        pritnf("variable is : 1.\n");
    case 2:
        pritnf("variable is : 2.\n");
    default:
        printf("variable is : %d.\n", variable);
}
```

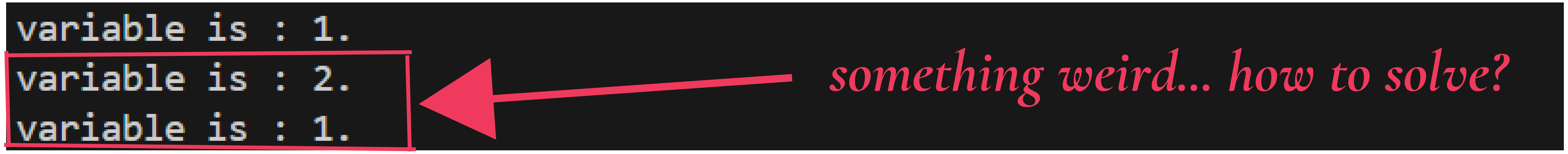


```
variable is : 1.
variable is : 2.
variable is : 1.
```

Switch : the syntax in C

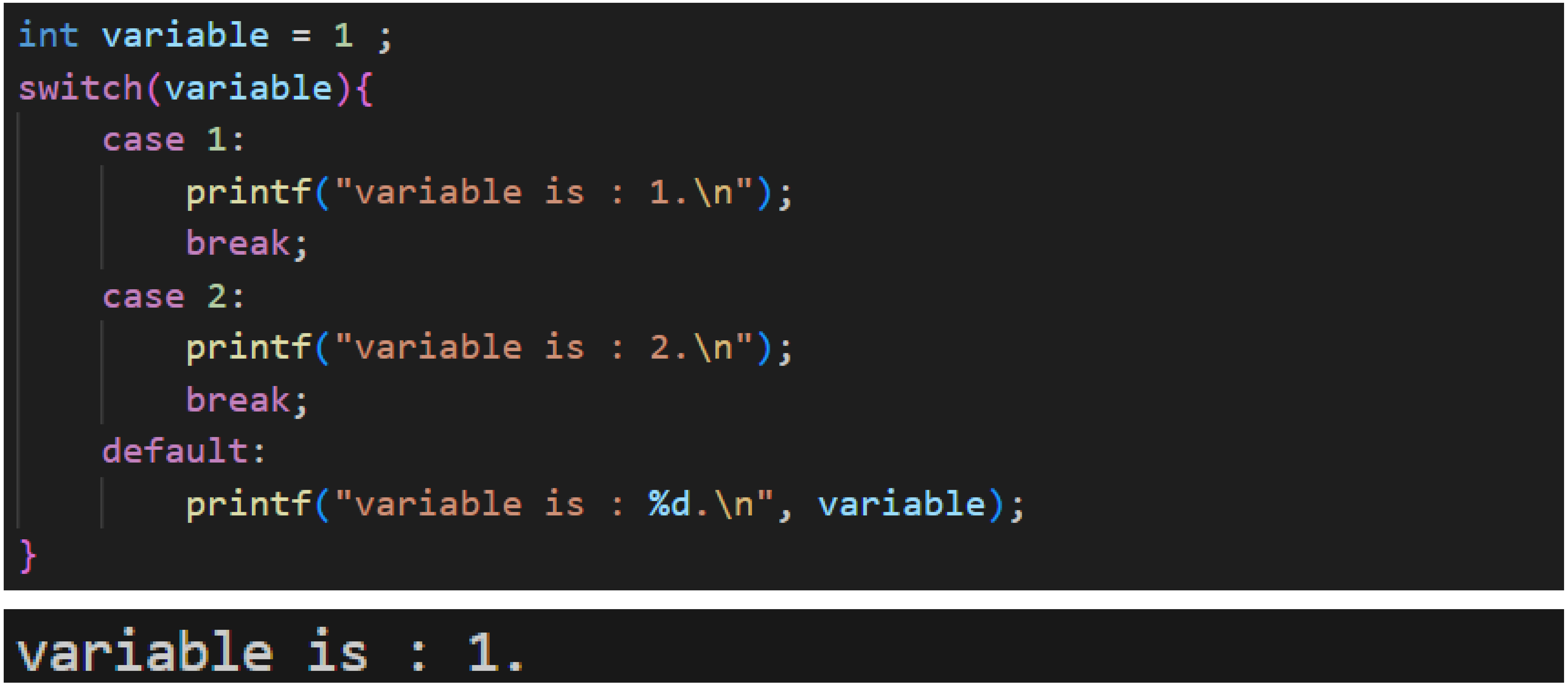


```c
int variable = 1 ;
switch(variable){
    case 1:
        pritnf("variable is : 1.\n");
    case 2:
        pritnf("variable is : 2.\n");
    default:
        printf("variable is : %d.\n", variable);
}
```



```
variable is : 1.
variable is : 2.
variable is : 1.
```

Switch : the syntax in C

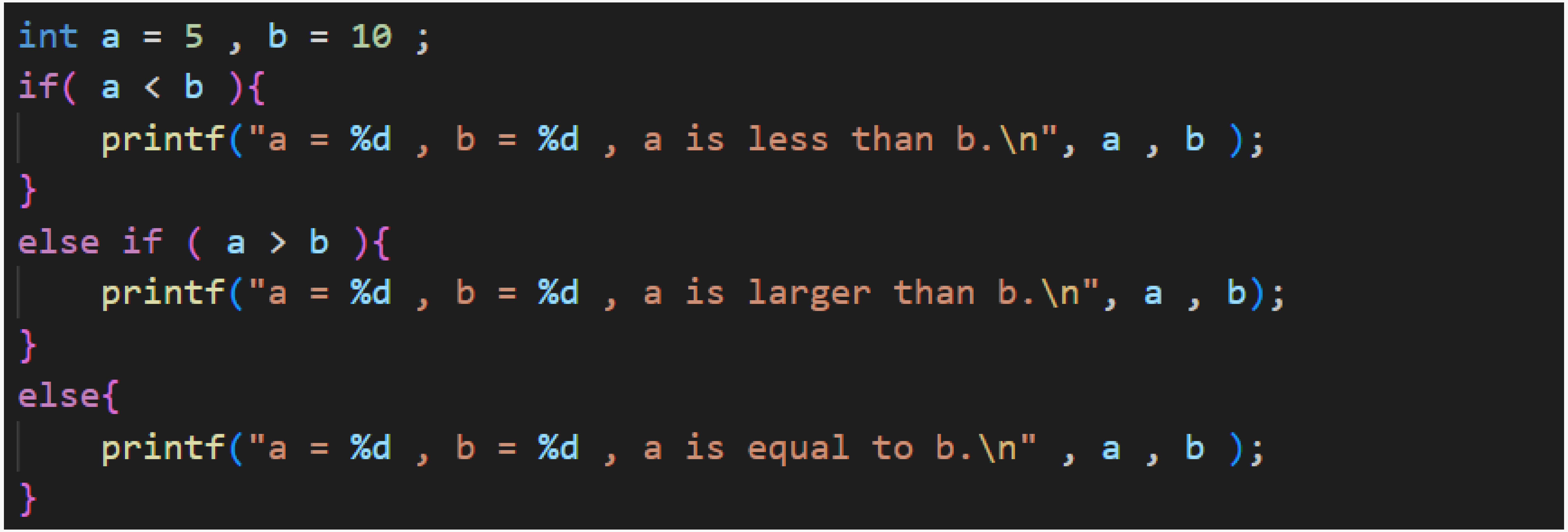


```c
int variable = 1 ;
switch(variable){
    case 1:
        printf("variable is : 1.\n");
        break;
    case 2:
        printf("variable is : 2.\n");
        break;
    default:
        printf("variable is : %d.\n", variable);
}
```

```
variable is : 1.
```

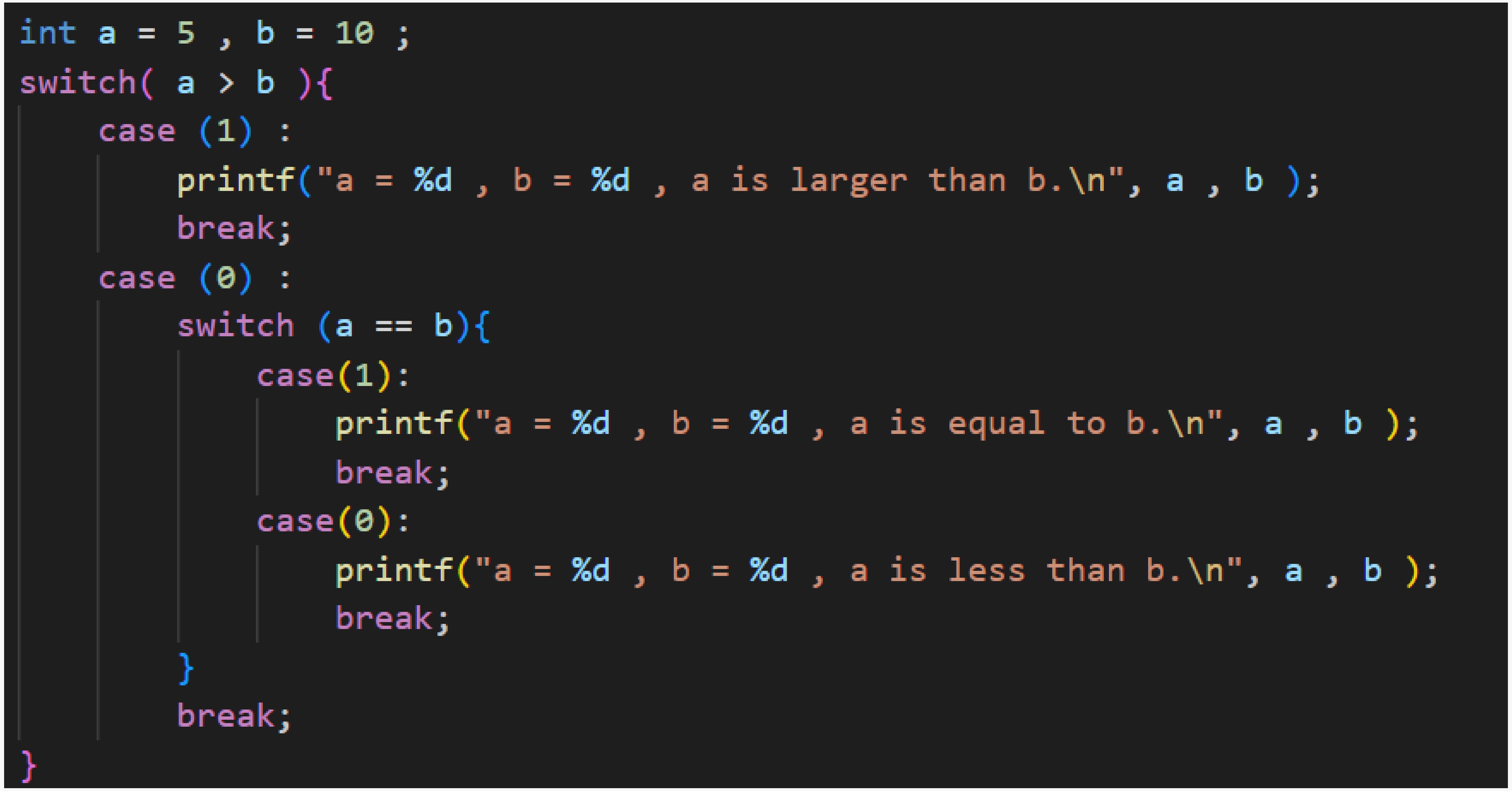
Switch : the syntax in C

How do we use switch to replace if/else in C ?



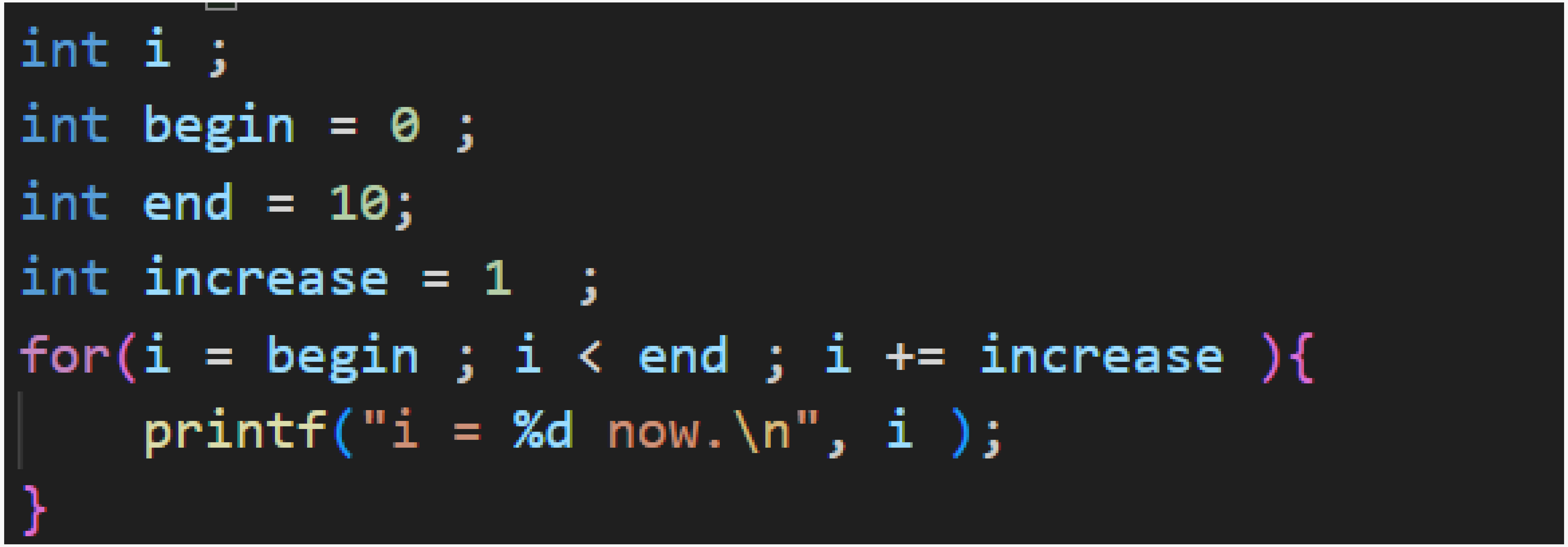
```
int a = 5 , b = 10 ;
if( a < b ){
    printf("a = %d , b = %d , a is less than b.\n", a , b );
}
else if ( a > b ){
    printf("a = %d , b = %d , a is larger than b.\n", a , b);
}
else{
    printf("a = %d , b = %d , a is equal to b.\n" , a , b );
}
```

Switch : the syntax in C


```c
int a = 5 , b = 10 ;
switch( a > b ){
    case (1) :
        printf("a = %d , b = %d , a is larger than b.\n", a , b );
        break;
    case (0) :
        switch (a == b){
            case(1):
                printf("a = %d , b = %d , a is equal to b.\n", a , b );
                break;
            case(0):
                printf("a = %d , b = %d , a is less than b.\n", a , b );
                break;
        }
        break;
}
```

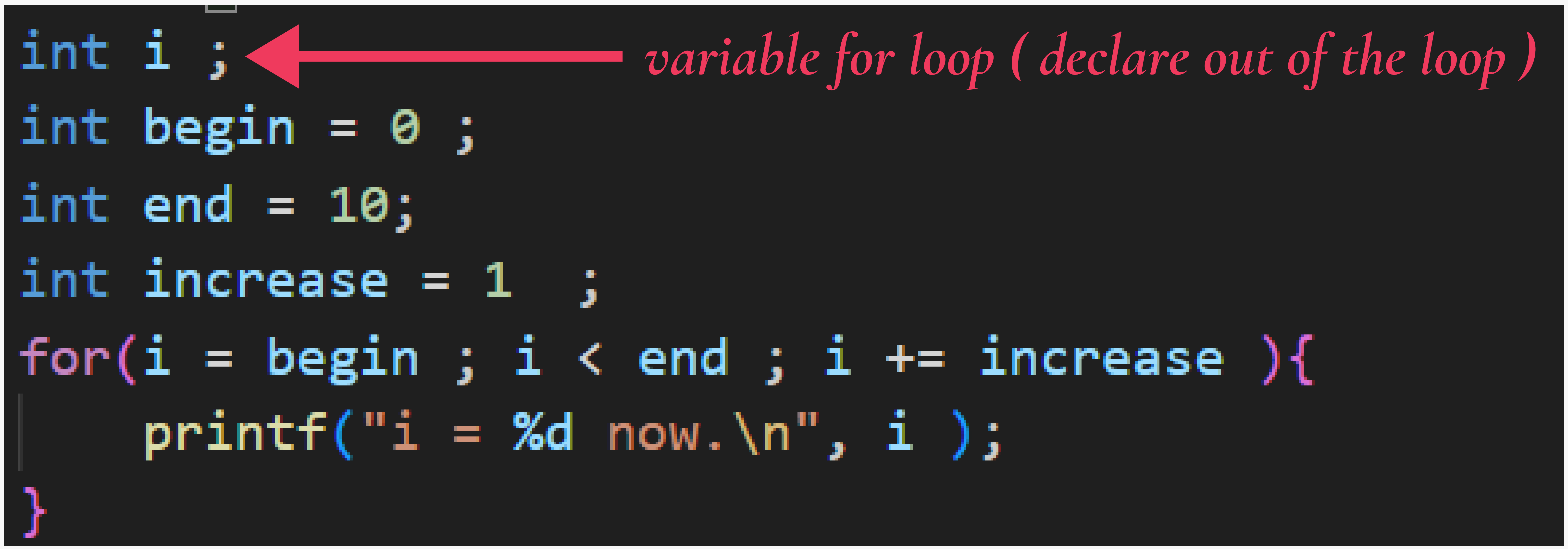
4. for/while loop

for loop : the syntax in C



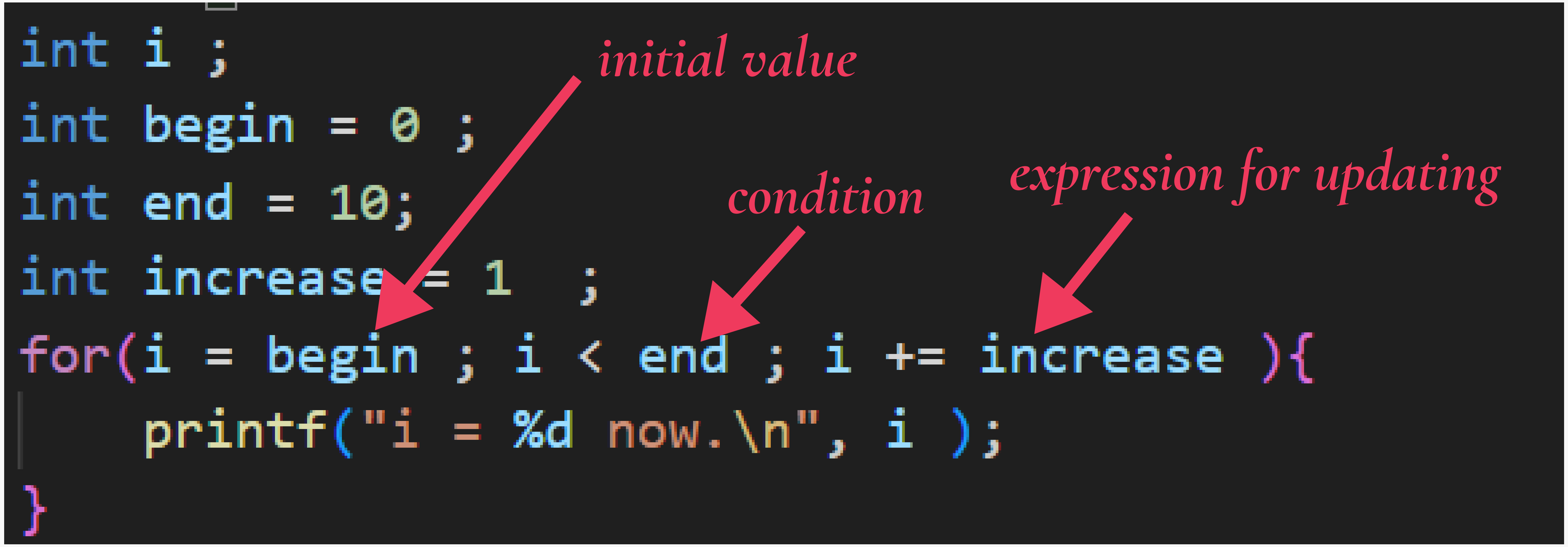
```c
int i ;
int begin = 0 ;
int end = 10;
int increase = 1  ;
for(i = begin ; i < end ; i += increase ){
    printf("i = %d now.\n", i );
}
```

for loop : the syntax in C



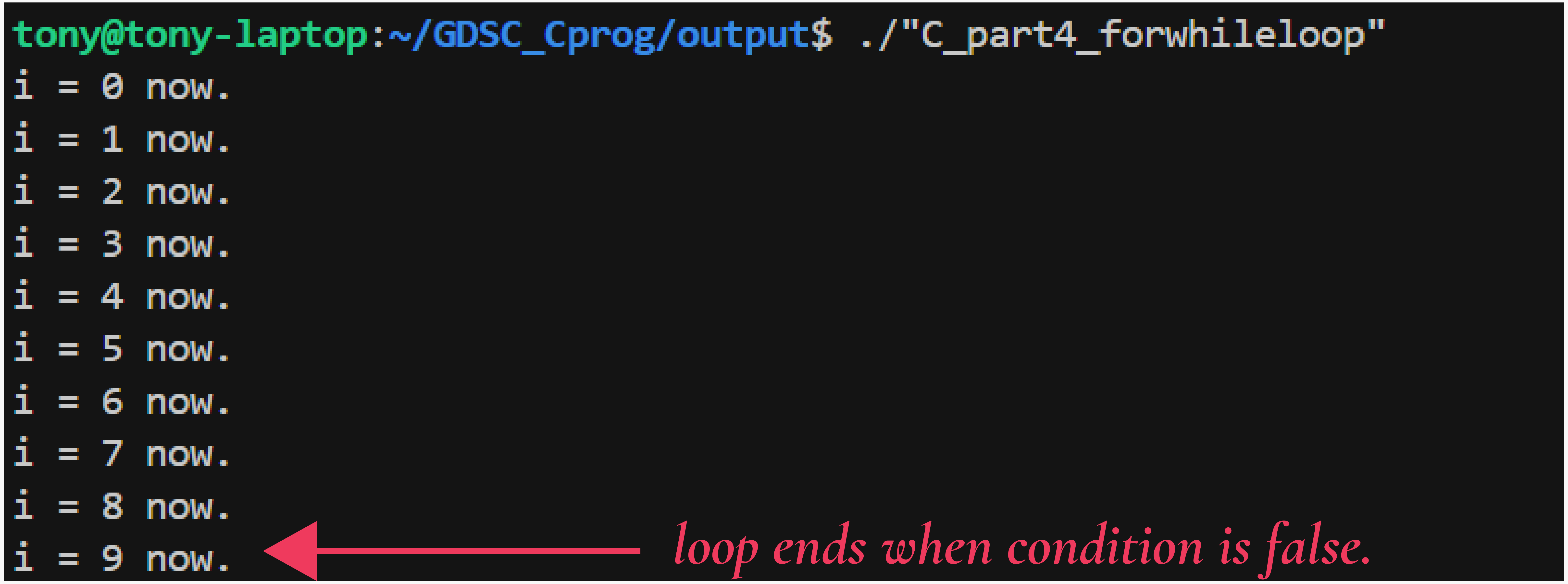
```
int i ;
int begin = 0 ;
int end = 10;
int increase = 1   ;
for(i = begin ; i < end ; i += increase ){
    printf("i = %d now.\n", i );
}
```

for loop : the syntax in C



```
int i ;
int begin = 0 ;
int end = 10;
int increase = 1  ;
for(i = begin ; i < end ; i += increase ){
    printf("i = %d now.\n", i );
}
```

for loop : the syntax in C

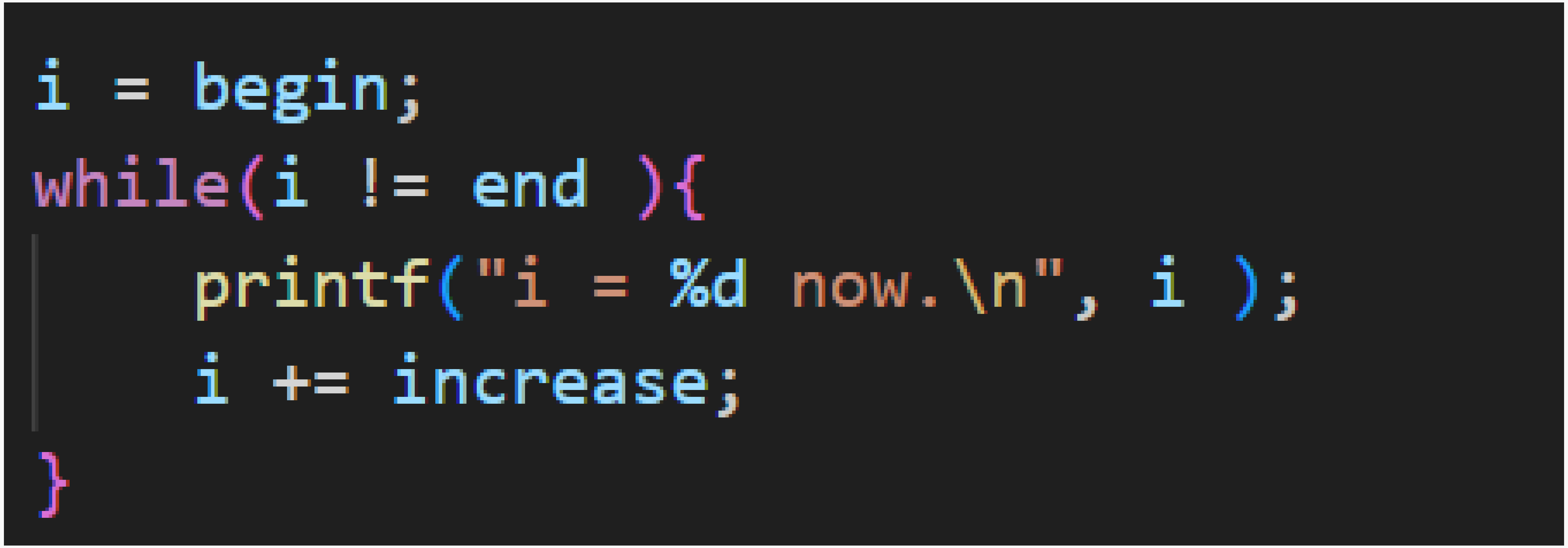


```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part4_forwhileloop"
i = 0 now.
i = 1 now.
i = 2 now.
i = 3 now.
i = 4 now.
i = 5 now.
i = 6 now.
i = 7 now.
i = 8 now.
i = 9 now.
```

loop ends when condition is false.

while loop : the syntax in C

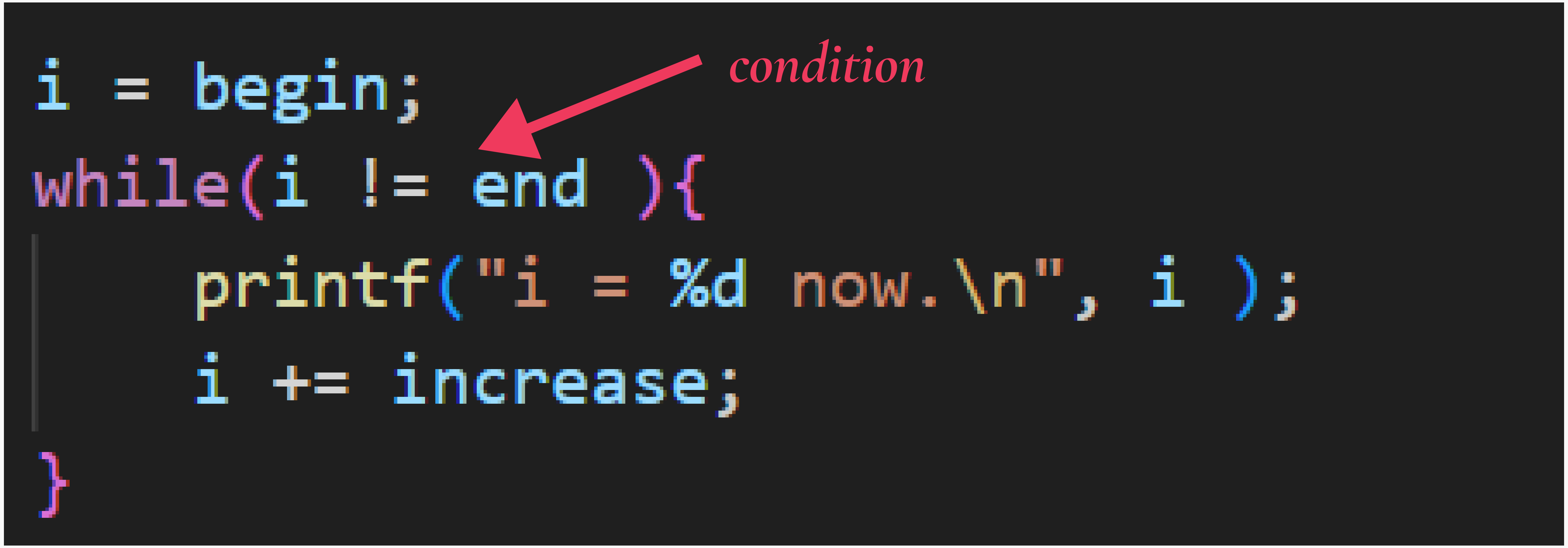
There are two types of while loop , while and do while.



```
i = begin;
while(i != end ){
    printf("i = %d now.\n", i );
    i += increase;
}
```

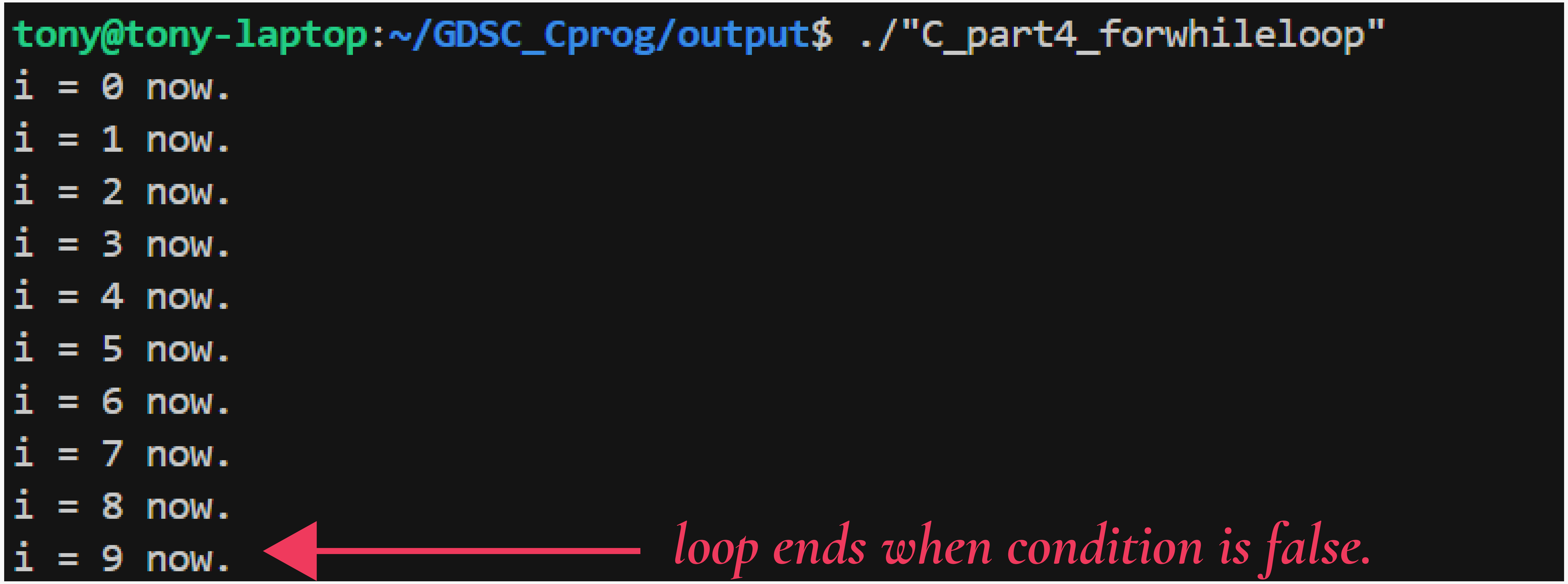
while loop : the syntax in C

There are two types of while loop , while and do while.



```
i = begin;
while(i != end ){
    printf("i = %d now.\n", i );
    i += increase;
}
```

while loop : the syntax in C

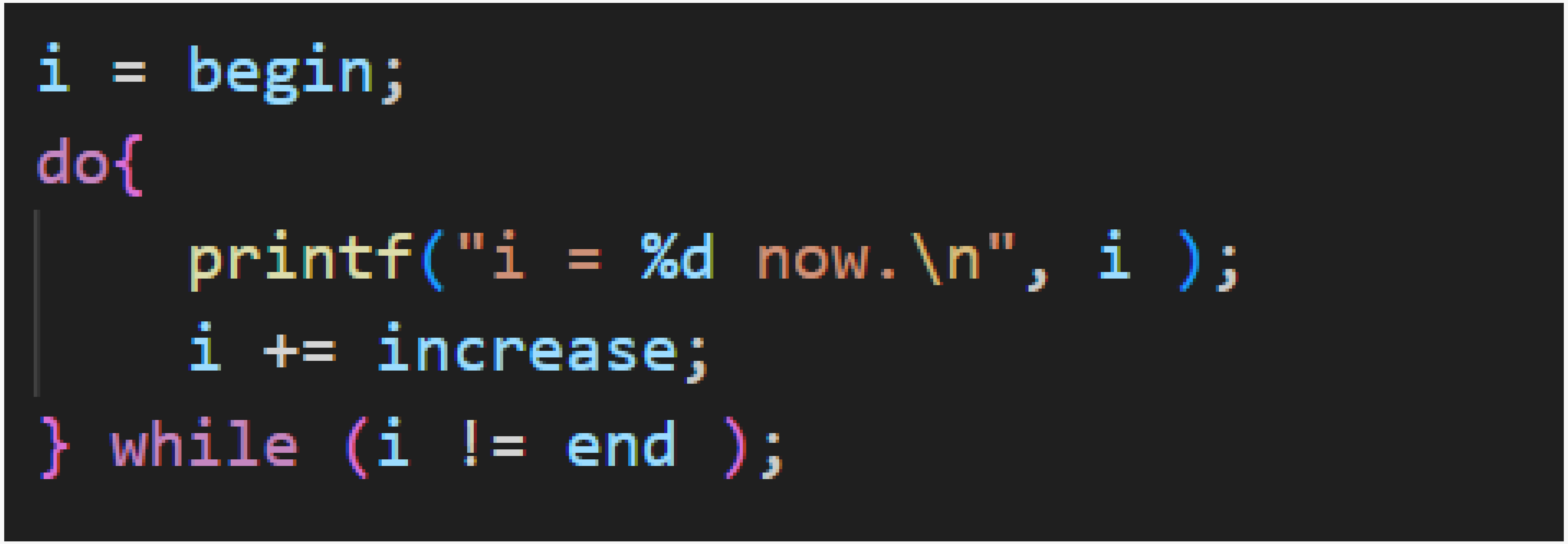


```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part4_forwhileloop"
i = 0 now.
i = 1 now.
i = 2 now.
i = 3 now.
i = 4 now.
i = 5 now.
i = 6 now.
i = 7 now.
i = 8 now.
i = 9 now.
```

loop ends when condition is false.

while loop : the syntax in C

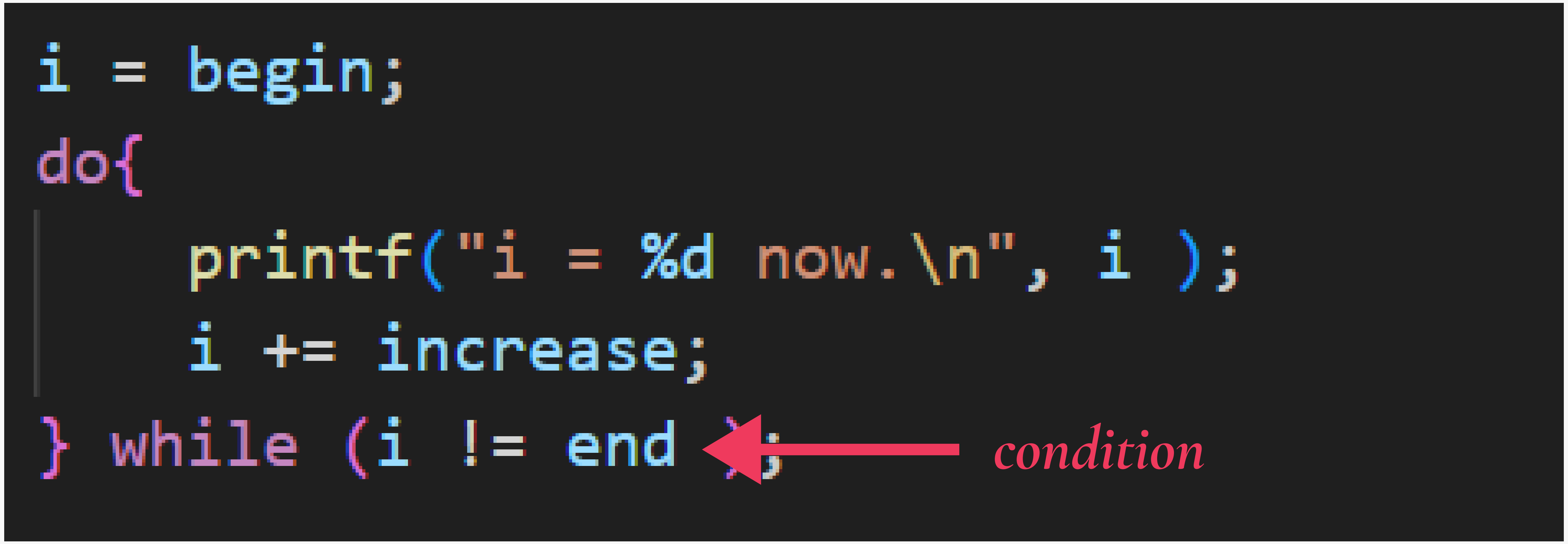
There are two types of while loop , while and do while.



```
i = begin;
do{
    printf("i = %d now.\n", i );
    i += increase;
} while (i != end );
```

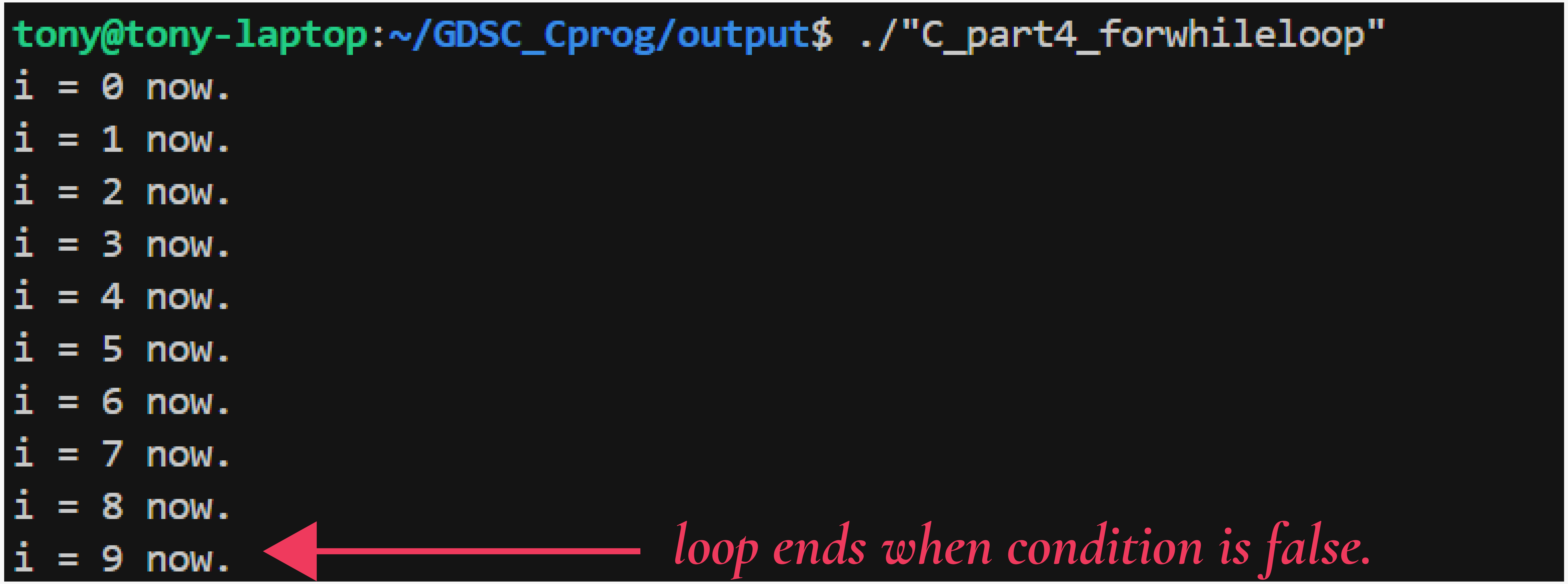
while loop : the syntax in C

There are two types of while loop , while and do while.



```
i = begin;
do{
    printf("i = %d now.\n", i );
    i += increase;
} while (i != end );
```

while loop : the syntax in C



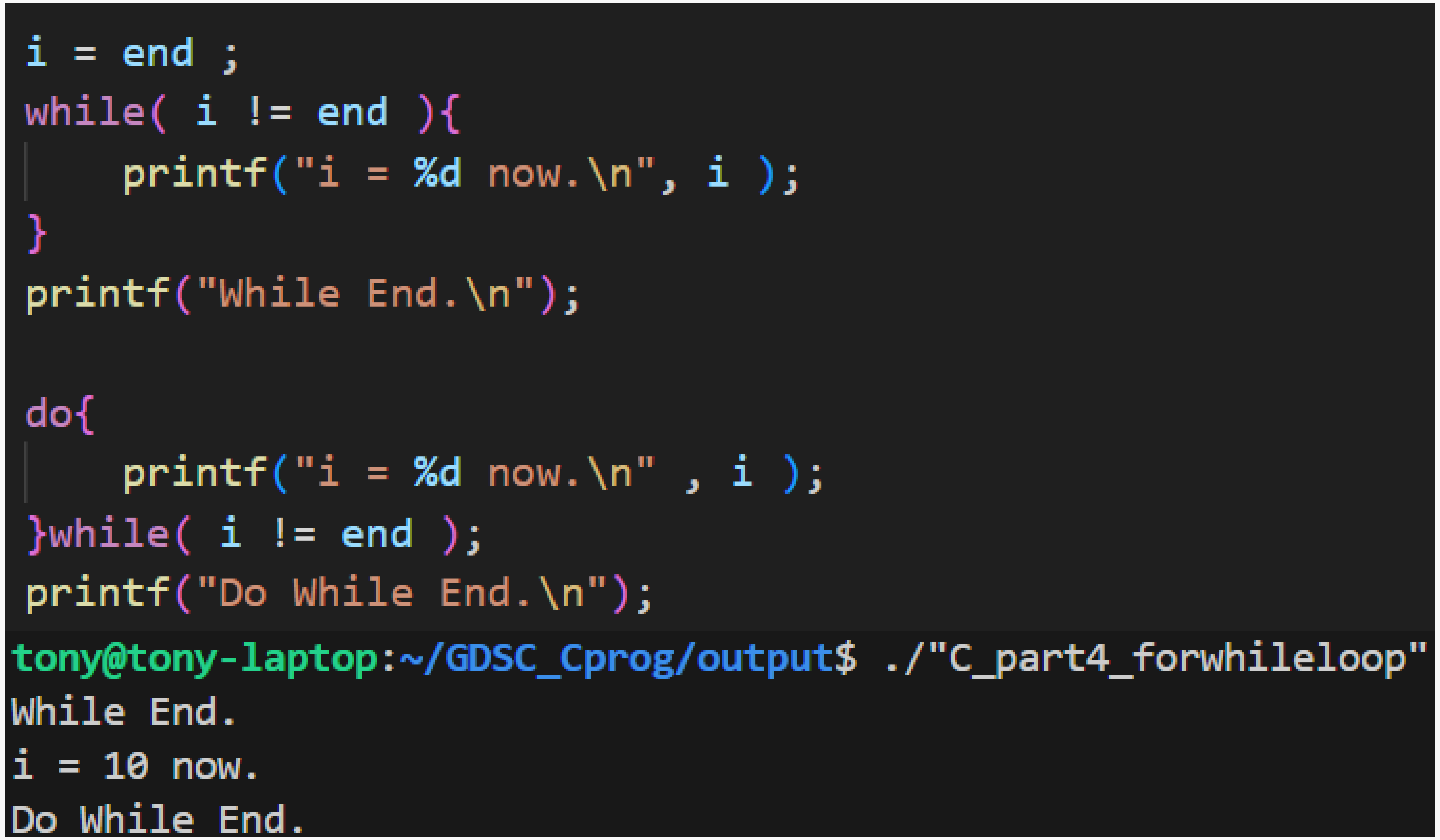
```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part4_forwhileloop"
i = 0 now.
i = 1 now.
i = 2 now.
i = 3 now.
i = 4 now.
i = 5 now.
i = 6 now.
i = 7 now.
i = 8 now.
i = 9 now.
```

loop ends when condition is false.

difference between while and do while

while : check condition and run loop once when looping.
do while : run loop once and check condition when looping.

difference between while and do while

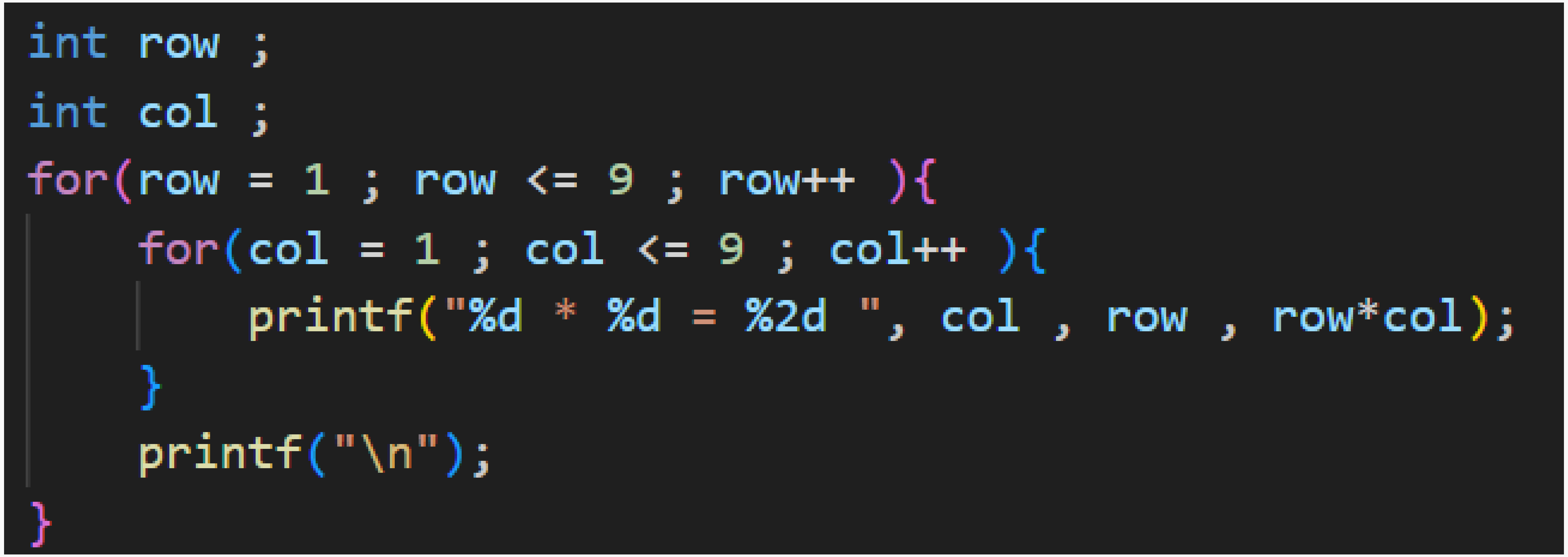


```c
i = end ;
while( i != end ){
    printf("i = %d now.\n", i );
}
printf("While End.\n");

do{
    printf("i = %d now.\n" , i );
}while( i != end );
printf("Do While End.\n");
```

```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part4_forwhileloop"
While End.
i = 10 now.
Do While End.
```

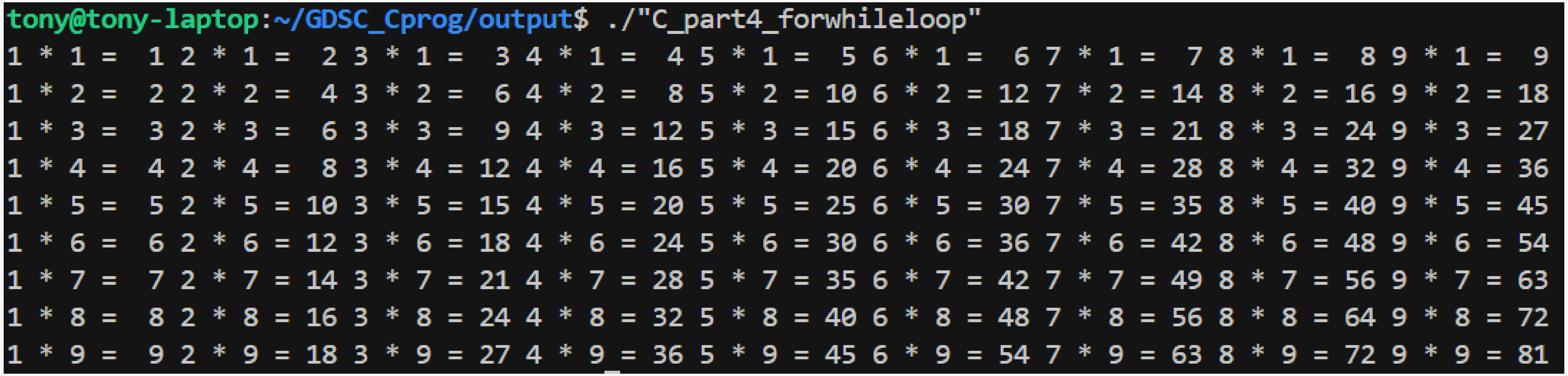
for loop : nested loop

Definition of nested loop : place other loops inside a loop.



```
int row ;
int col ;
for(row = 1 ; row <= 9 ; row++ ){
    for(col = 1 ; col <= 9 ; col++ ){
        printf("%d * %d = %2d ", col , row , row*col);
    }
    printf("\n");
}
```

for loop : nested loop



```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_part4_forwhileloop"
1 * 1 =  1 2 * 1 =  2 3 * 1 =  3 4 * 1 =  4 5 * 1 =  5 6 * 1 =  6 7 * 1 =  7 8 * 1 =  8 9 * 1 =  9
1 * 2 =  2 2 * 2 =  4 3 * 2 =  6 4 * 2 =  8 5 * 2 = 10 6 * 2 = 12 7 * 2 = 14 8 * 2 = 16 9 * 2 = 18
1 * 3 =  3 2 * 3 =  6 3 * 3 =  9 4 * 3 = 12 5 * 3 = 15 6 * 3 = 18 7 * 3 = 21 8 * 3 = 24 9 * 3 = 27
1 * 4 =  4 2 * 4 =  8 3 * 4 = 12 4 * 4 = 16 5 * 4 = 20 6 * 4 = 24 7 * 4 = 28 8 * 4 = 32 9 * 4 = 36
1 * 5 =  5 2 * 5 = 10 3 * 5 = 15 4 * 5 = 20 5 * 5 = 25 6 * 5 = 30 7 * 5 = 35 8 * 5 = 40 9 * 5 = 45
1 * 6 =  6 2 * 6 = 12 3 * 6 = 18 4 * 6 = 24 5 * 6 = 30 6 * 6 = 36 7 * 6 = 42 8 * 6 = 48 9 * 6 = 54
1 * 7 =  7 2 * 7 = 14 3 * 7 = 21 4 * 7 = 28 5 * 7 = 35 6 * 7 = 42 7 * 7 = 49 8 * 7 = 56 9 * 7 = 63
1 * 8 =  8 2 * 8 = 16 3 * 8 = 24 4 * 8 = 32 5 * 8 = 40 6 * 8 = 48 7 * 8 = 56 8 * 8 = 64 9 * 8 = 72
1 * 9 =  9 2 * 9 = 18 3 * 9 = 27 4 * 9 = 36 5 * 9 = 45 6 * 9 = 54 7 * 9 = 63 8 * 9 = 72 9 * 9 = 81
```

Topic : Logic training

1. Number base

2. LCM and GCD

3. Prime Number

•••••

1. Number base

Number base

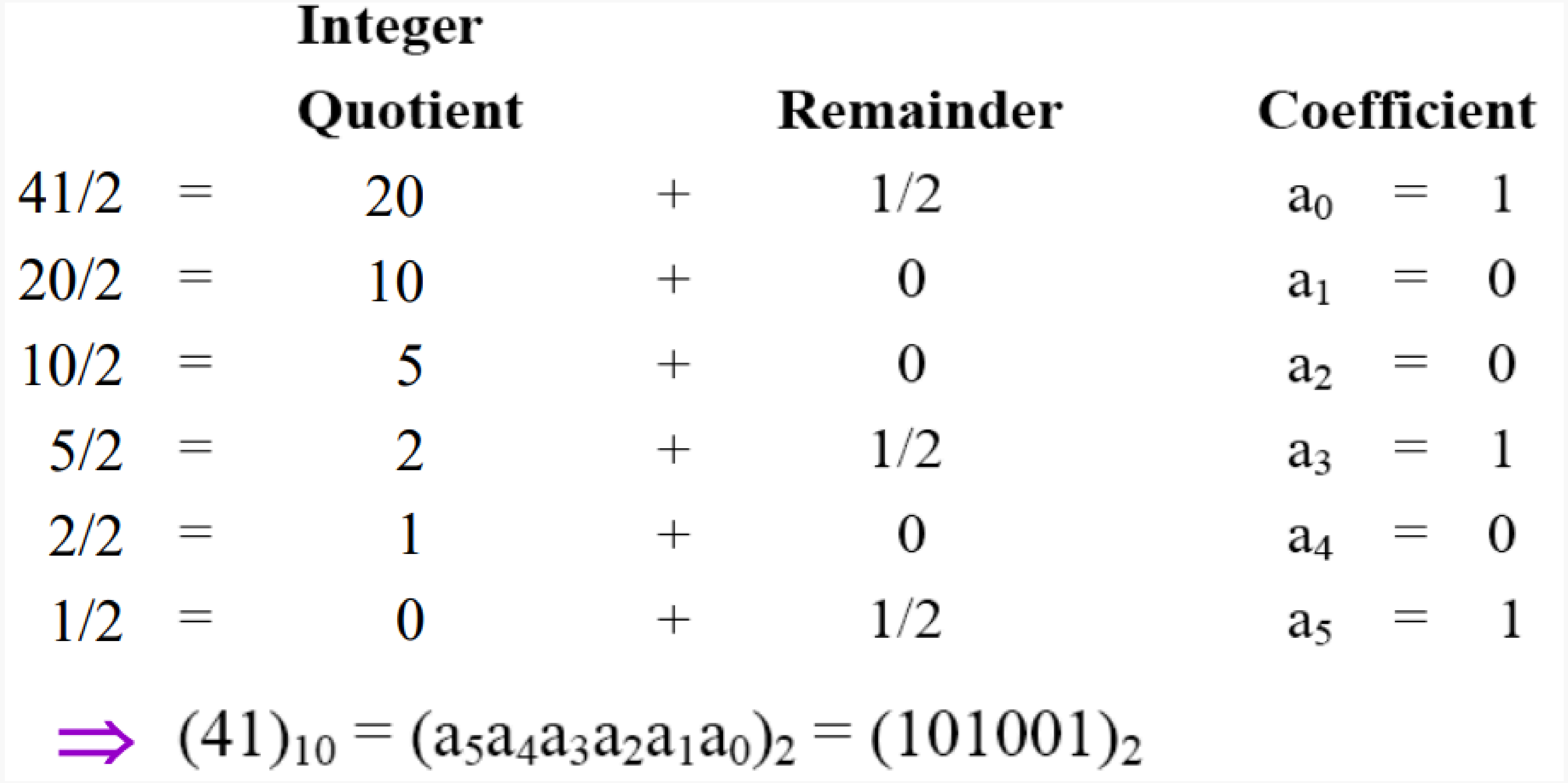
Def : A number base is the combination of digits that a system of counting uses to represent numbers.

Example:

1. the base 10 system : 7 = 7 * 10^0 => 7
2. the base 2 system : 111 = 1 * 2^2 + 1 * 2^1 + 1 * 2^0 => 7
3. the base 3 system : 21 = 2 * 3^1 + 1 * 3^0 => 7
4. the base 4 system : 13 = 1 * 4^1 + 3 * 4^0 => 7

Number base

If we have number 41 in base 10 , how to convert it to base 2 ?

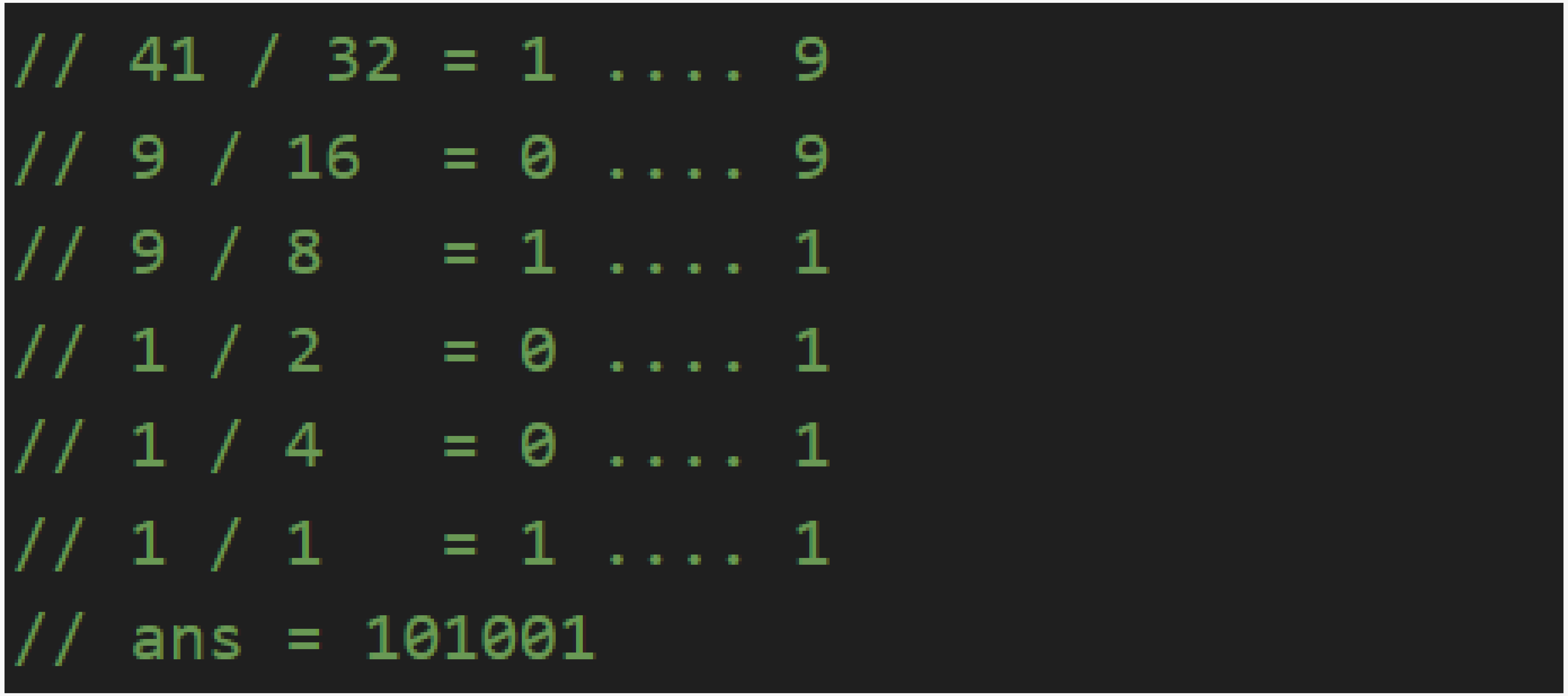


		Integer Quotient		Remainder	Coefficient
$41/2$	$=$	20	$+$	$1/2$	$a_0 = 1$
$20/2$	$=$	10	$+$	0	$a_1 = 0$
$10/2$	$=$	5	$+$	0	$a_2 = 0$
$5/2$	$=$	2	$+$	$1/2$	$a_3 = 1$
$2/2$	$=$	1	$+$	0	$a_4 = 0$
$1/2$	$=$	0	$+$	$1/2$	$a_5 = 1$

$$\Rightarrow (41)_{10} = (a_5a_4a_3a_2a_1a_0)_2 = (101001)_2$$

Number base

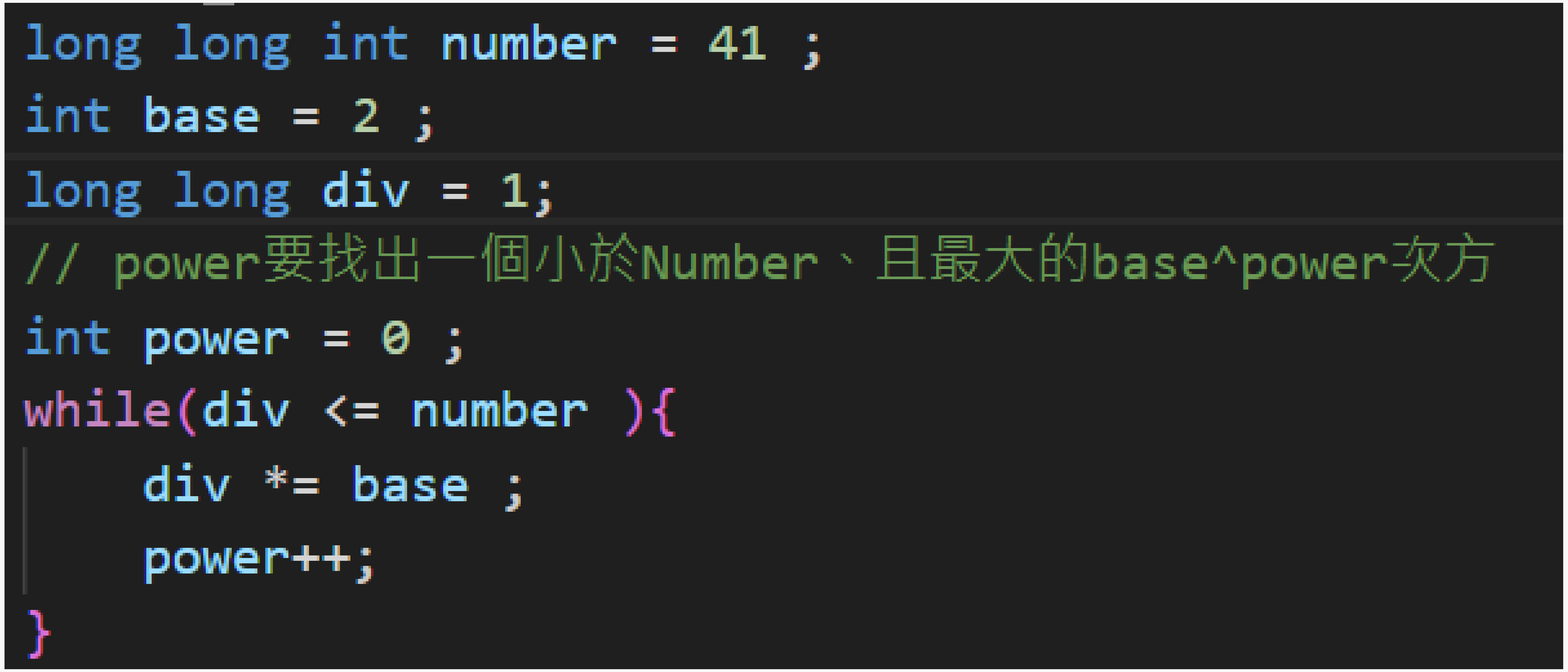
If we have number 41 in base 10 , how to convert it to base 2 ?



```
// 41 / 32 = 1 .... 9
// 9 / 16  = 0 .... 9
// 9 / 8   = 1 .... 1
// 1 / 2   = 0 .... 1
// 1 / 4   = 0 .... 1
// 1 / 1   = 1 .... 1
// ans = 101001
```

Number base

If we have number 41 in base 10 , how to convert it to base 2 ?

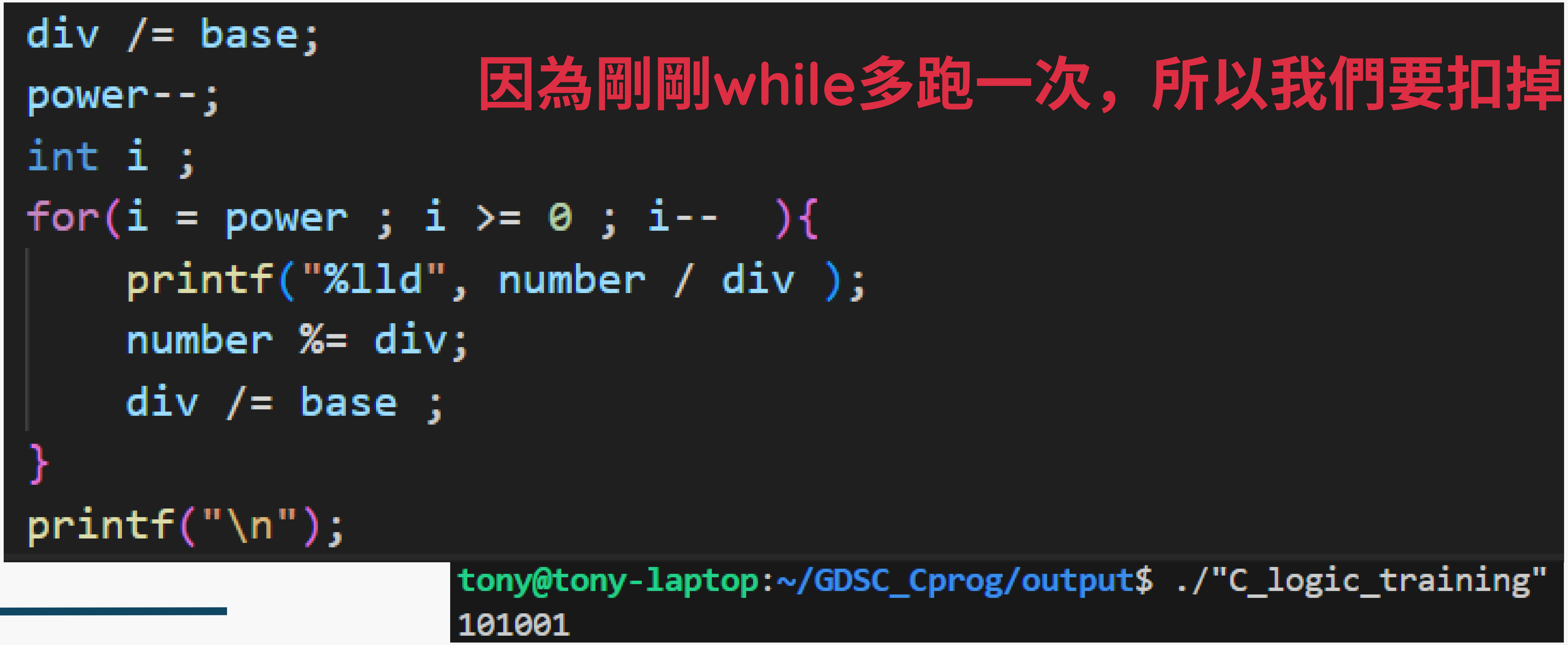


```
long long int number = 41 ;
int base = 2 ;
long long div = 1;
// power要找出一個小於Number、且最大的base^power次方
int power = 0 ;
while(div <= number ){
    div *= base ;
    power++;
}
```

請問這段code跑出來，power是多少

Number base

If we have number 41 in base 10 , how to convert it to base 2 ?



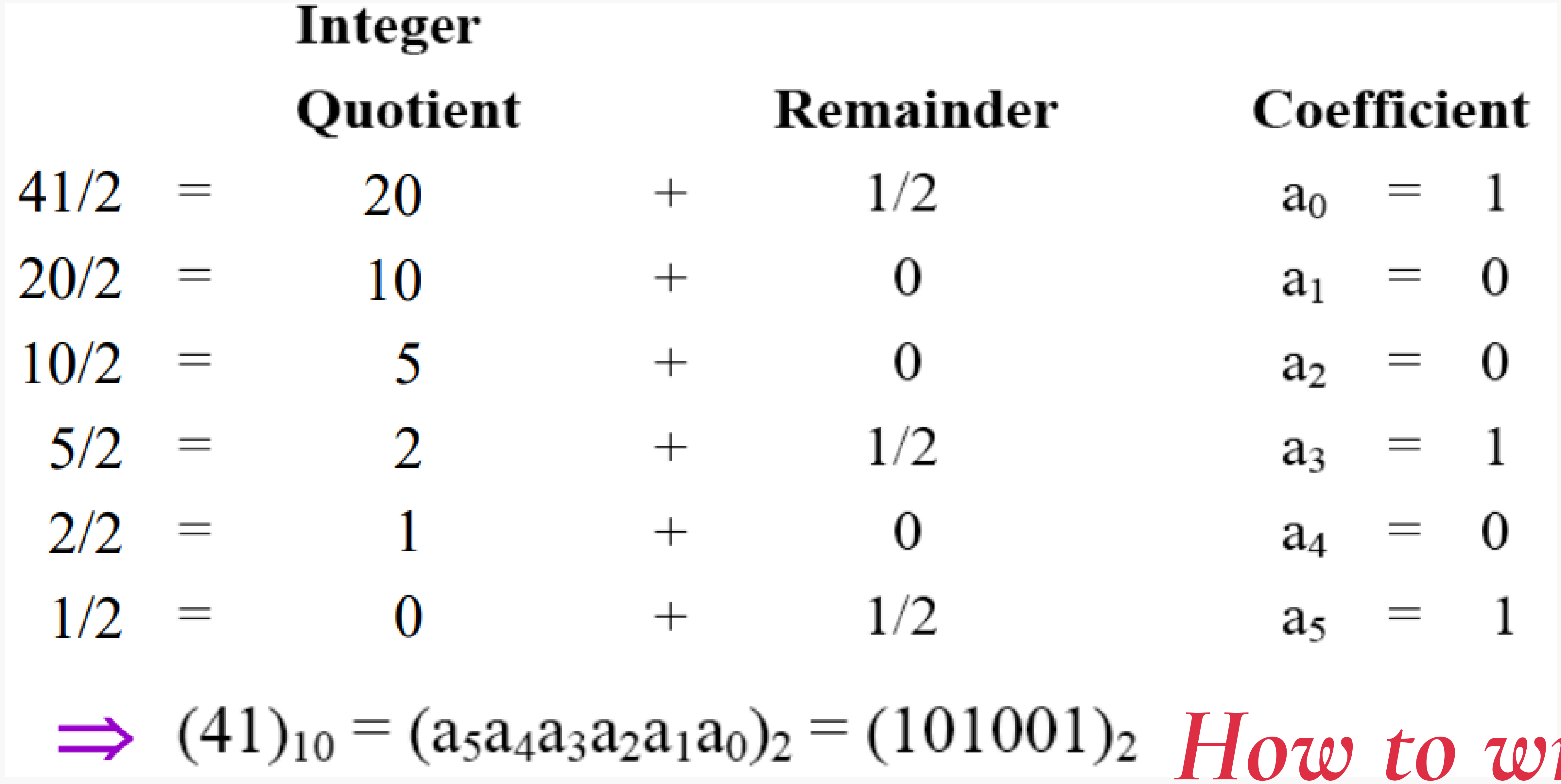
```
div /= base;
power--;
int i ;
for(i = power ; i >= 0 ; i--  ){
    printf("%lld", number / div );
    number %= div;
    div /= base ;
}
printf("\n");
```

因為剛剛while多跑一次，所以我們要扣掉

```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_logic_training"
101001
```

Number base

If we have number 41 in base 10 , how to convert it to base 2 ?

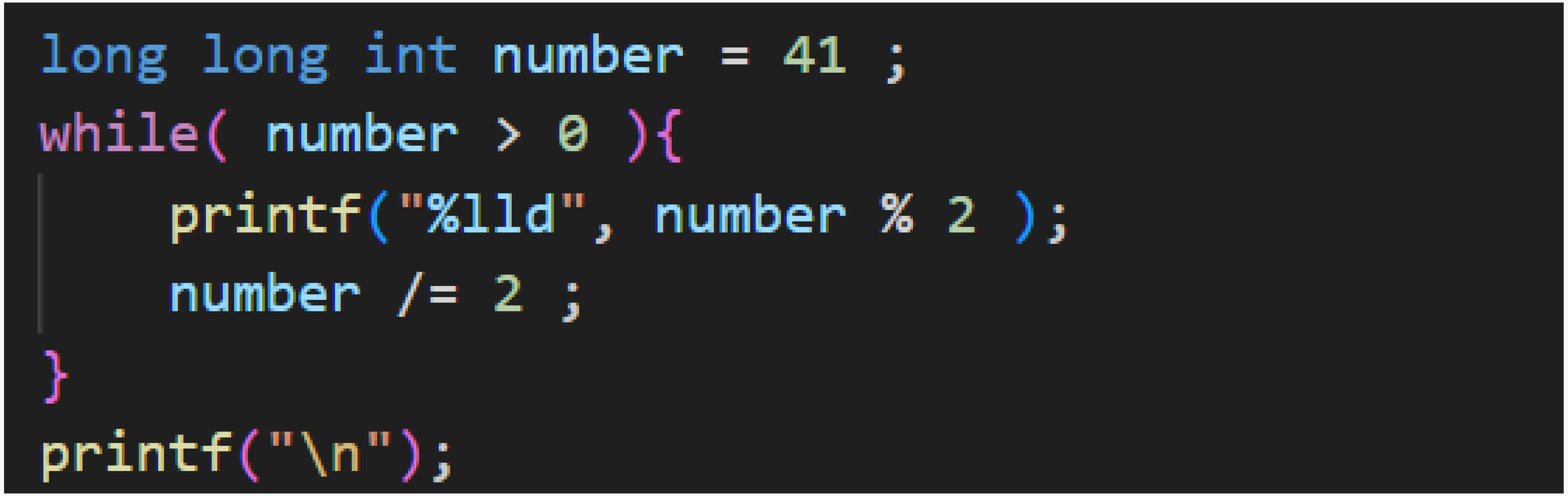


		Integer Quotient		Remainder	Coefficient		
41/2	=	20	+	1/2	a_0	=	1
20/2	=	10	+	0	a_1	=	0
10/2	=	5	+	0	a_2	=	0
5/2	=	2	+	1/2	a_3	=	1
2/2	=	1	+	0	a_4	=	0
1/2	=	0	+	1/2	a_5	=	1

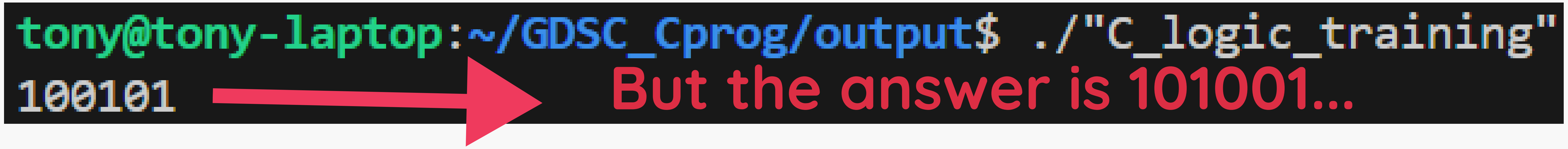
$\Rightarrow (41)_{10} = (a_5a_4a_3a_2a_1a_0)_2 = (101001)_2$

How to write it in C?

Number base



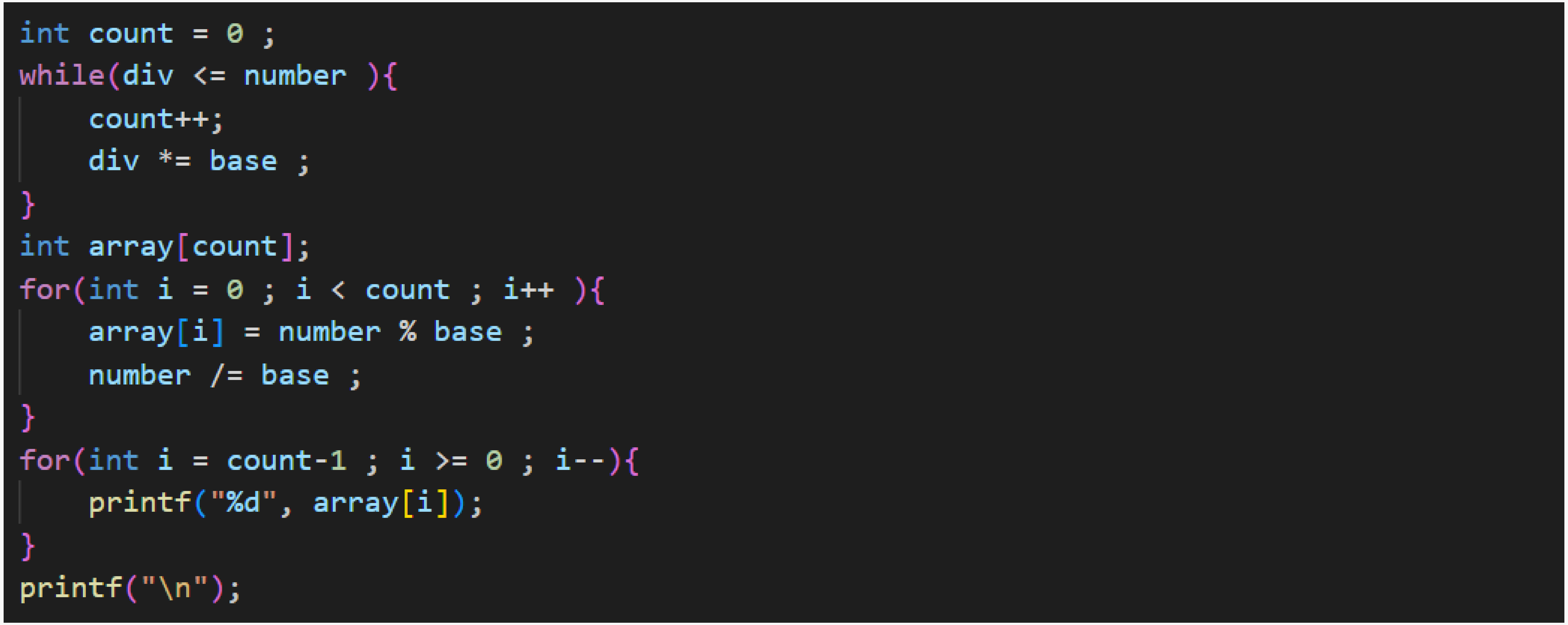
```c
long long int number = 41 ;
while( number > 0 ){
    printf("%lld", number % 2 );
    number /= 2 ;
}
printf("\n");
```



```
tony@tony-laptop:~/GDSC_Cprog/output$ ./"C_logic_training"
100101
```

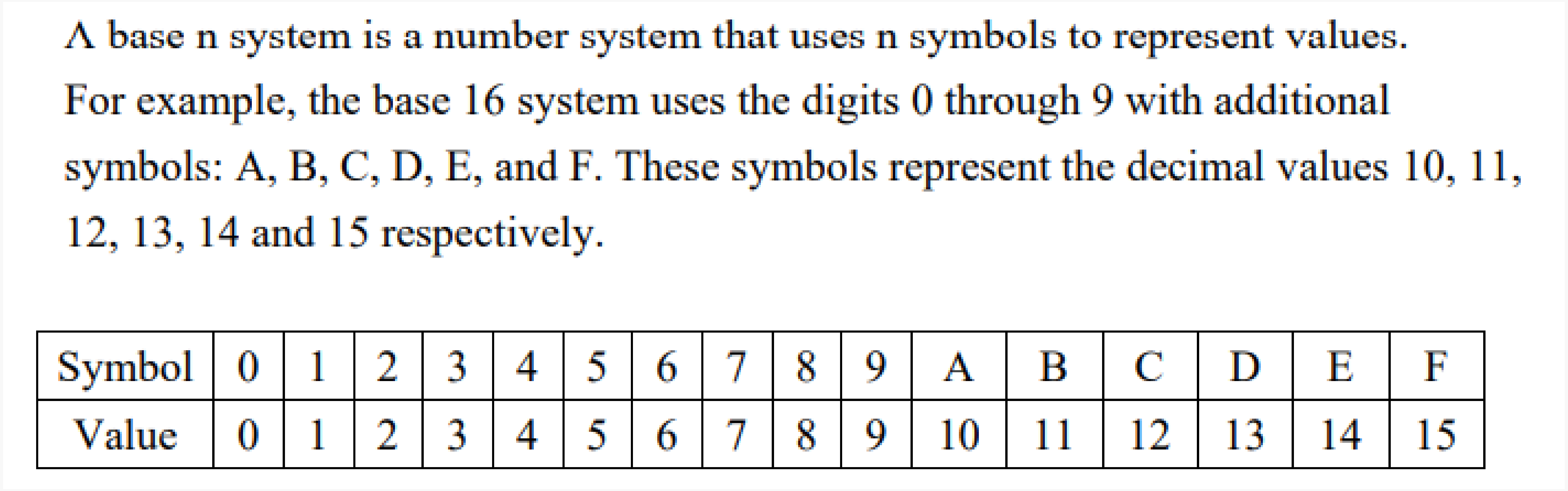
But the answer is 101001...

Number base : use array to do



```c
int count = 0 ;
while(div <= number ){
    count++;
    div *= base ;
}
int array[count];
for(int i = 0 ; i < count ; i++ ){
    array[i] = number % base ;
    number /= base ;
}
for(int i = count-1 ; i >= 0 ; i--){
    printf("%d", array[i]);
}
printf("\n");
```

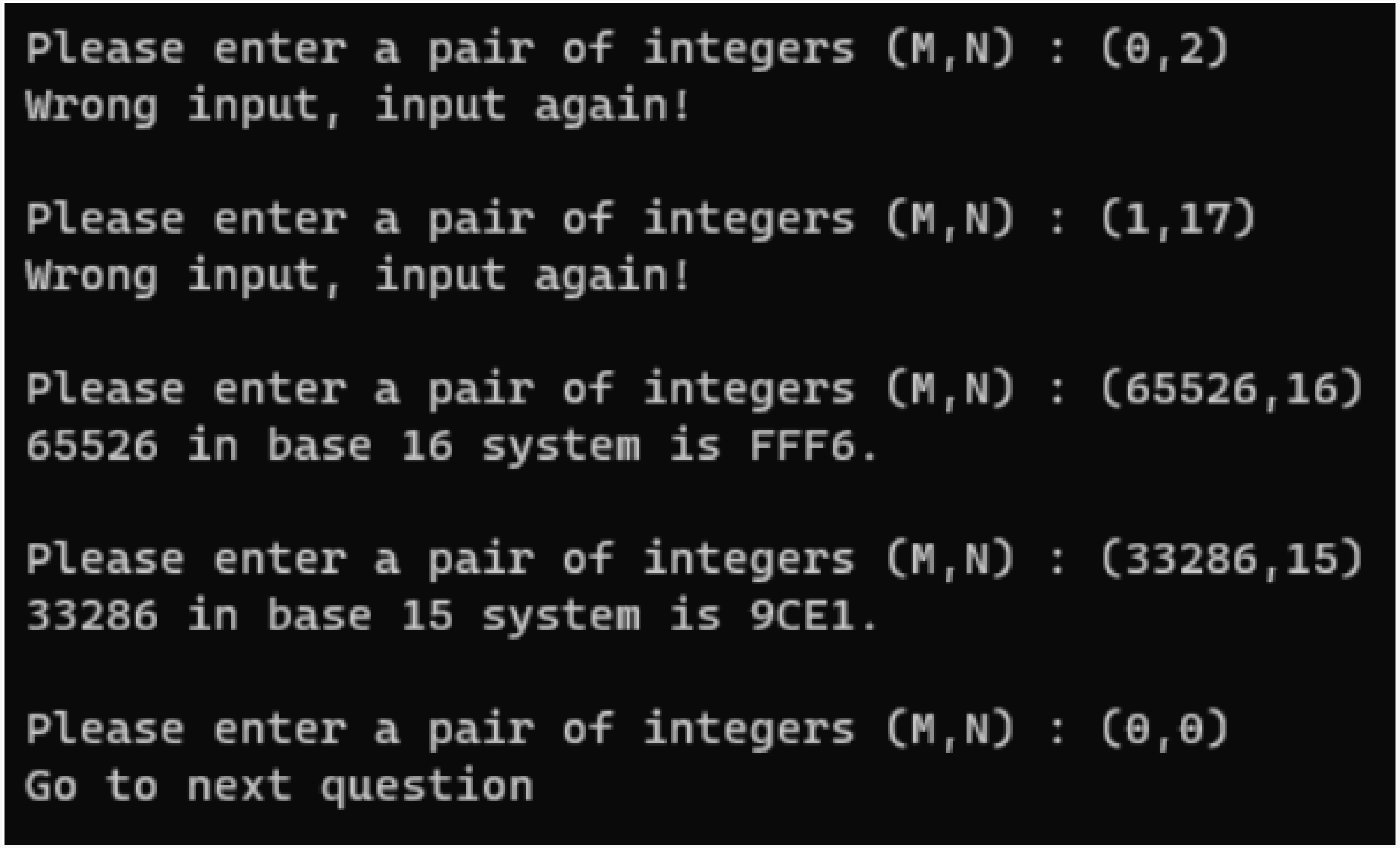
Number base : example



A base n system is a number system that uses n symbols to represent values. For example, the base 16 system uses the digits 0 through 9 with additional symbols: A, B, C, D, E, and F. These symbols represent the decimal values 10, 11, 12, 13, 14 and 15 respectively.

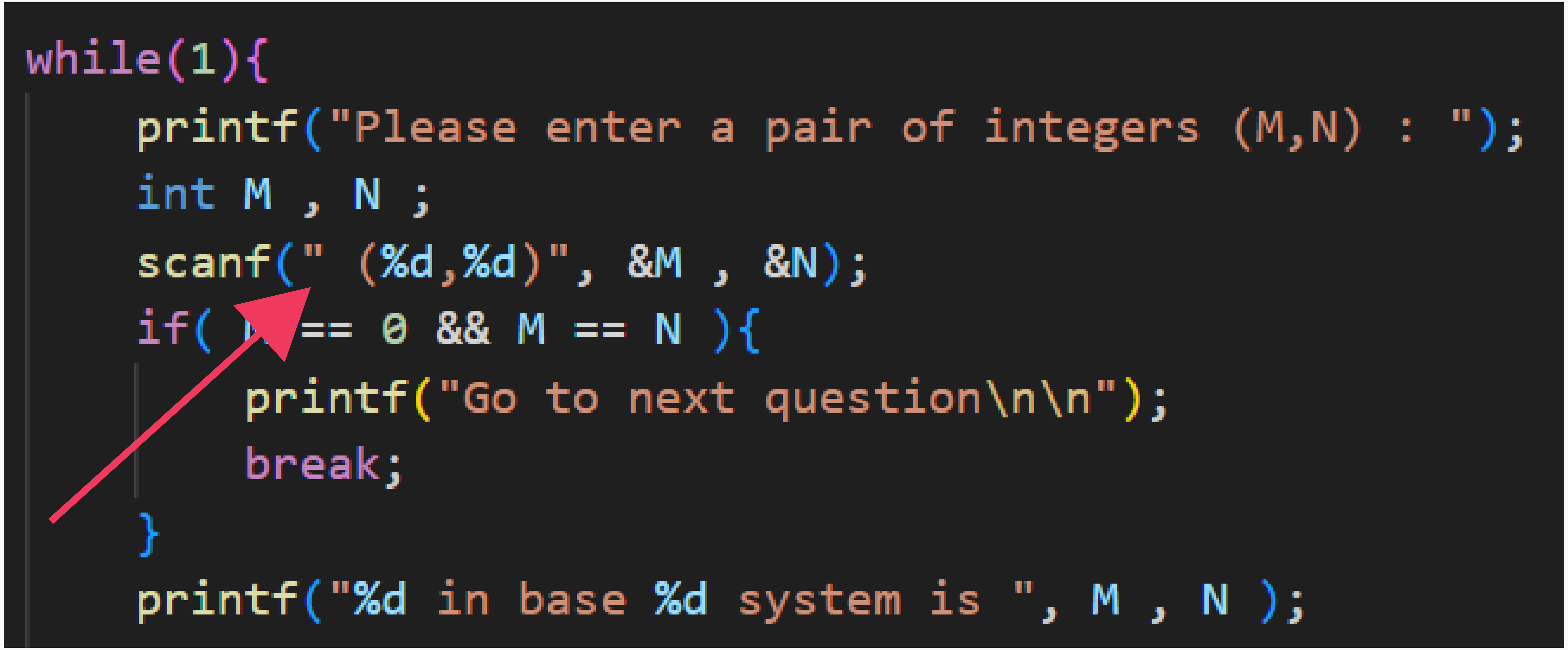
Symbol	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Number base : example



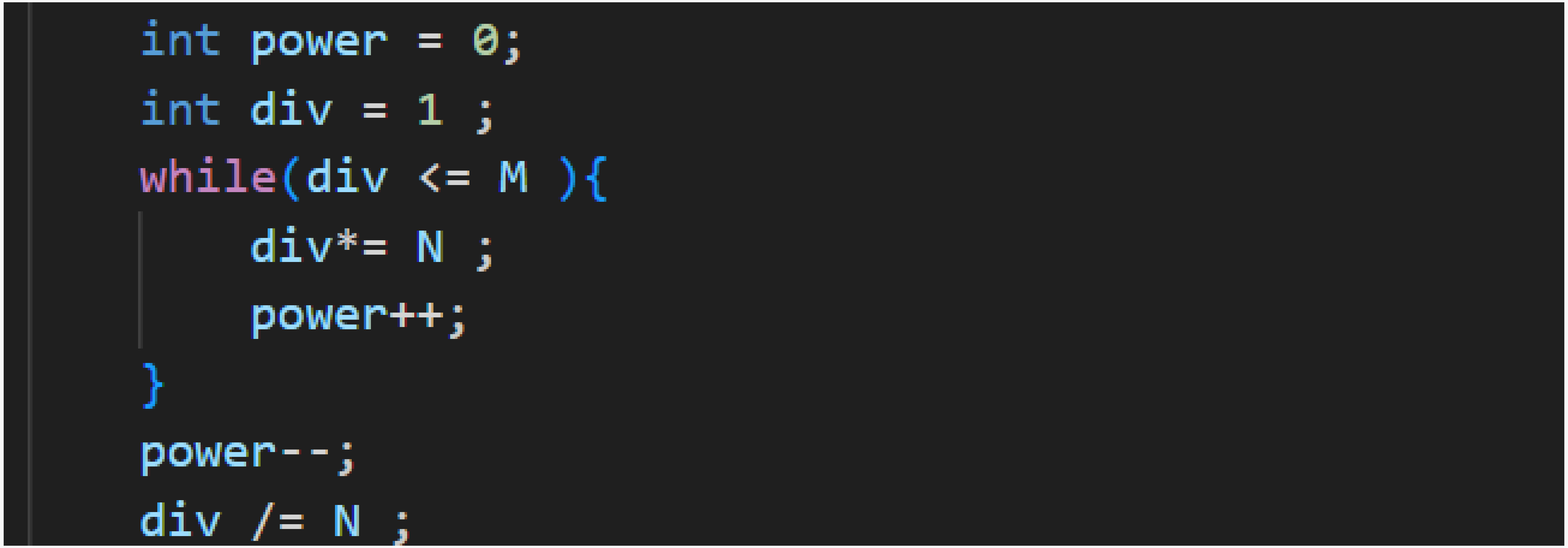
```
Please enter a pair of integers (M,N) : (0,2)
Wrong input, input again!

Please enter a pair of integers (M,N) : (1,17)
Wrong input, input again!

Please enter a pair of integers (M,N) : (65526,16)
65526 in base 16 system is FFF6.

Please enter a pair of integers (M,N) : (33286,15)
33286 in base 15 system is 9CE1.

Please enter a pair of integers (M,N) : (0,0)
Go to next question
```

Number base : example



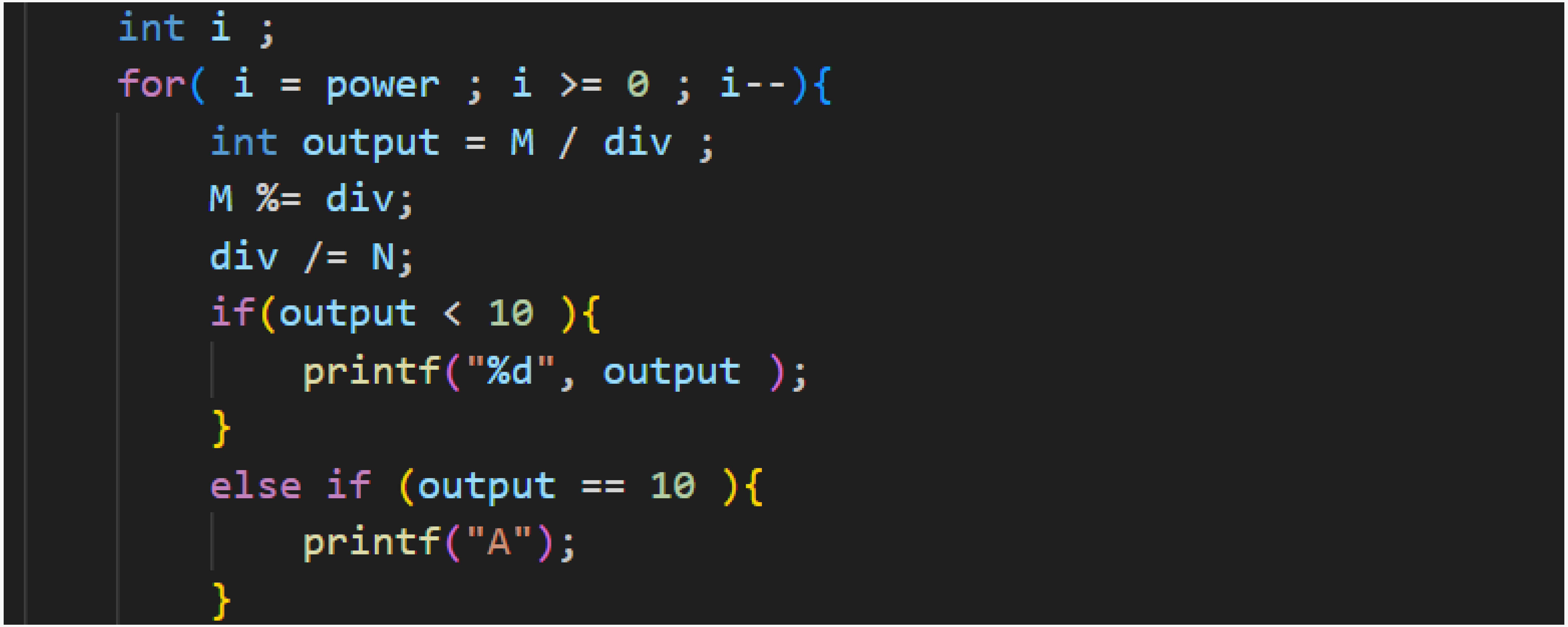
```c
while(1){
    printf("Please enter a pair of integers (M,N) : ");
    int M , N ;
    scanf(" (%d,%d)", &M , &N);
    if( N == 0 && M == N ){
        printf("Go to next question\n\n");
        break;
    }
    printf("%d in base %d system is ", M , N );
```

Number base : example



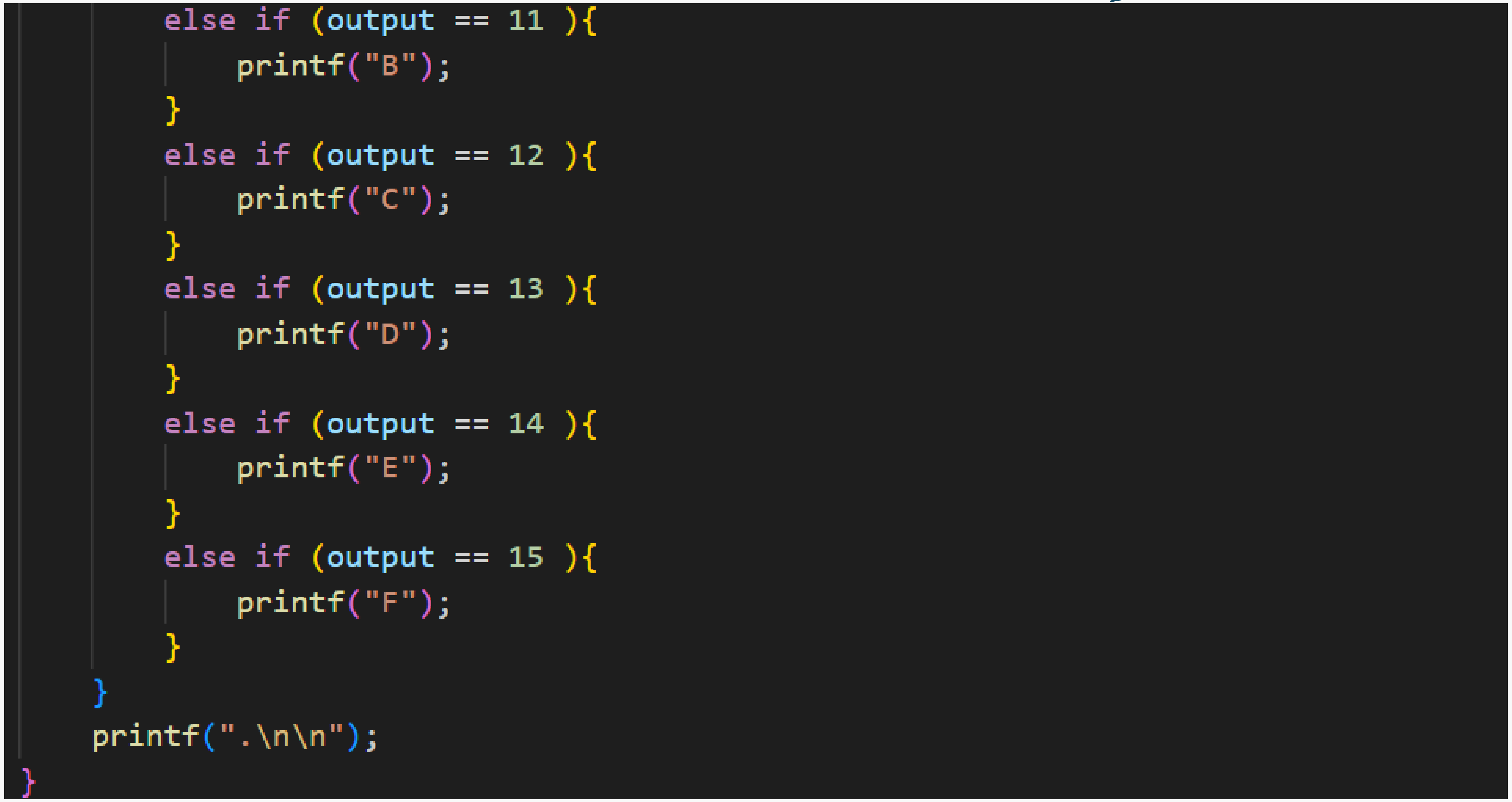
```
int power = 0;
int div = 1 ;
while(div <= M ){
    div*= N ;
    power++;
}
power--;
div /= N ;
```

Number base : example



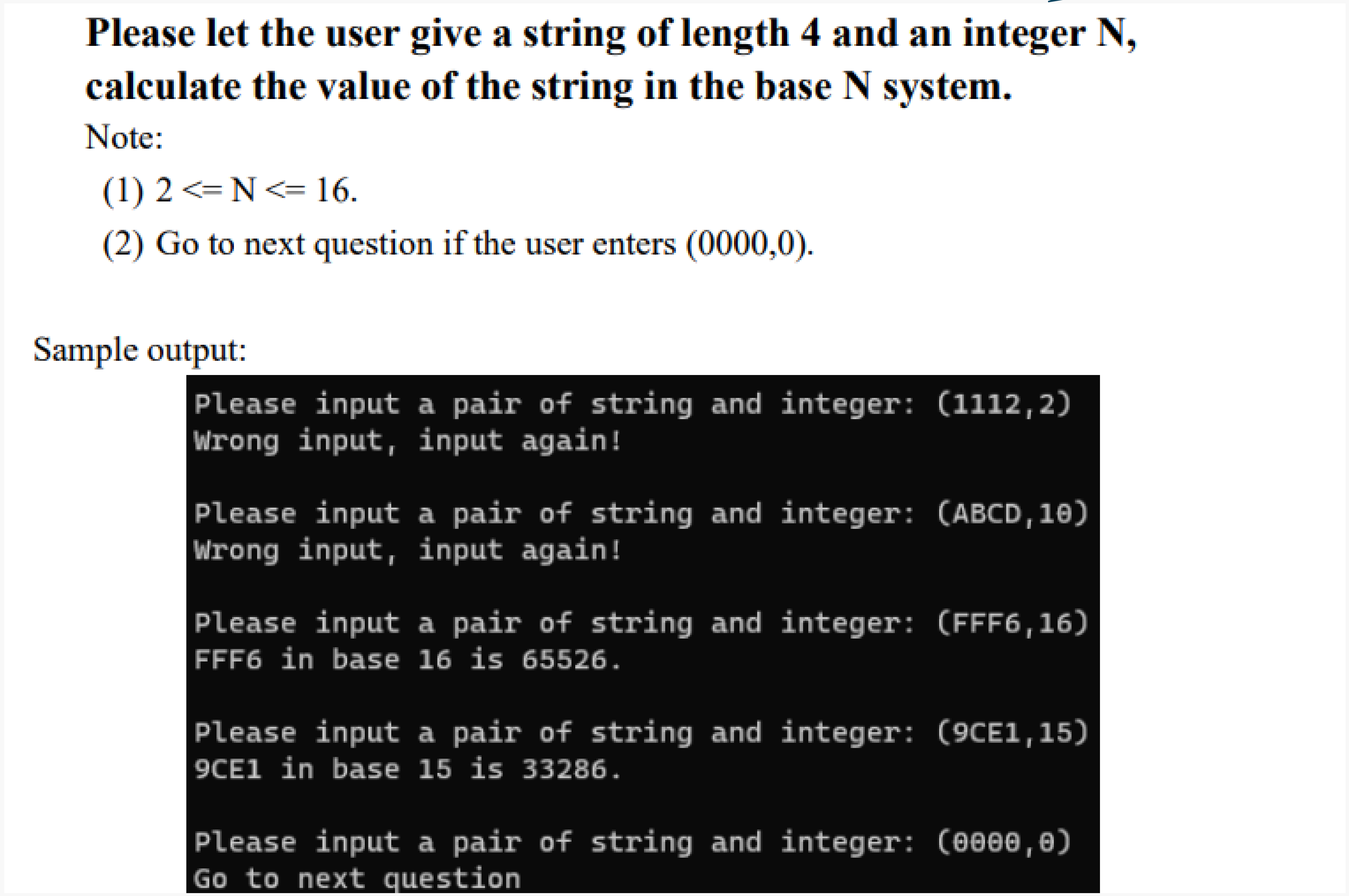
```c
int i ;
for( i = power ; i >= 0 ; i--){
    int output = M / div ;
    M %= div;
    div /= N;
    if(output < 10 ){
        printf("%d", output );
    }
    else if (output == 10 ){
        printf("A");
    }
```

Number base : example



```
        else if (output == 11 ){
            printf("B");
        }
        else if (output == 12 ){
            printf("C");
        }
        else if (output == 13 ){
            printf("D");
        }
        else if (output == 14 ){
            printf("E");
        }
        else if (output == 15 ){
            printf("F");
        }
    }
    printf(".\n\n");
}
```

Number base : example



Please let the user give a string of length 4 and an integer N, calculate the value of the string in the base N system.

Note:

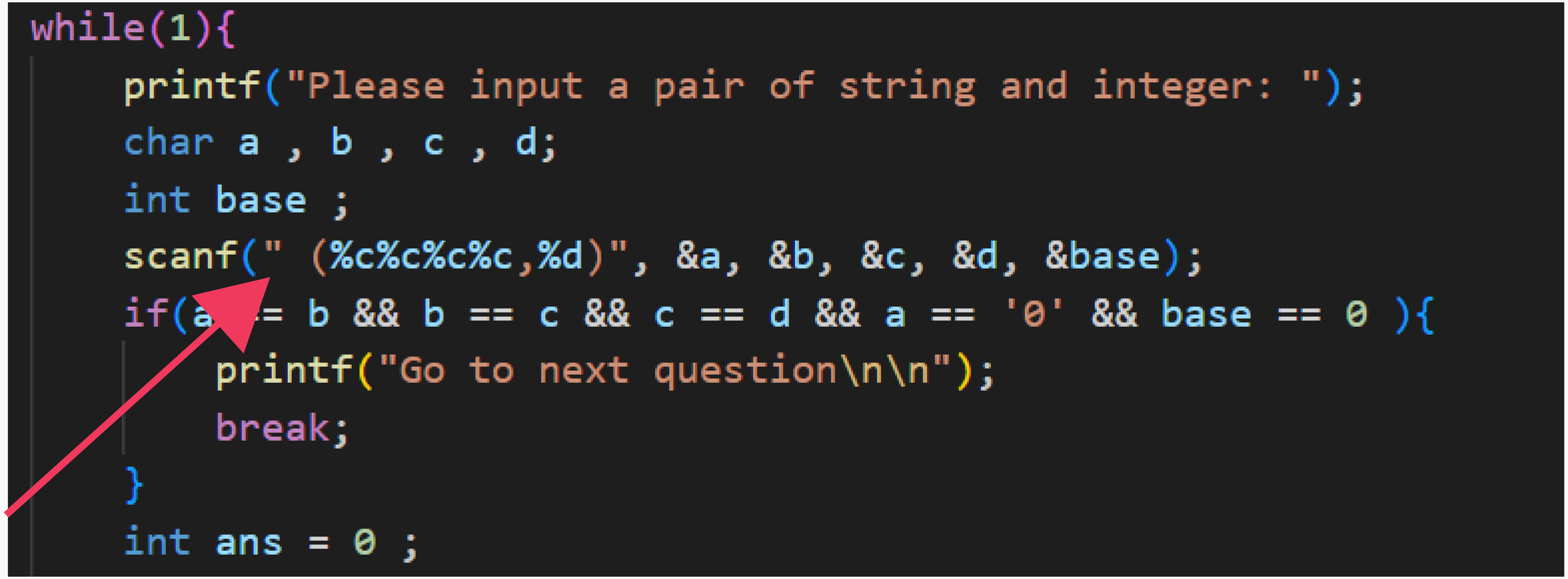
(1) 2 <= N <= 16.

(2) Go to next question if the user enters (0000,0).

Sample output:

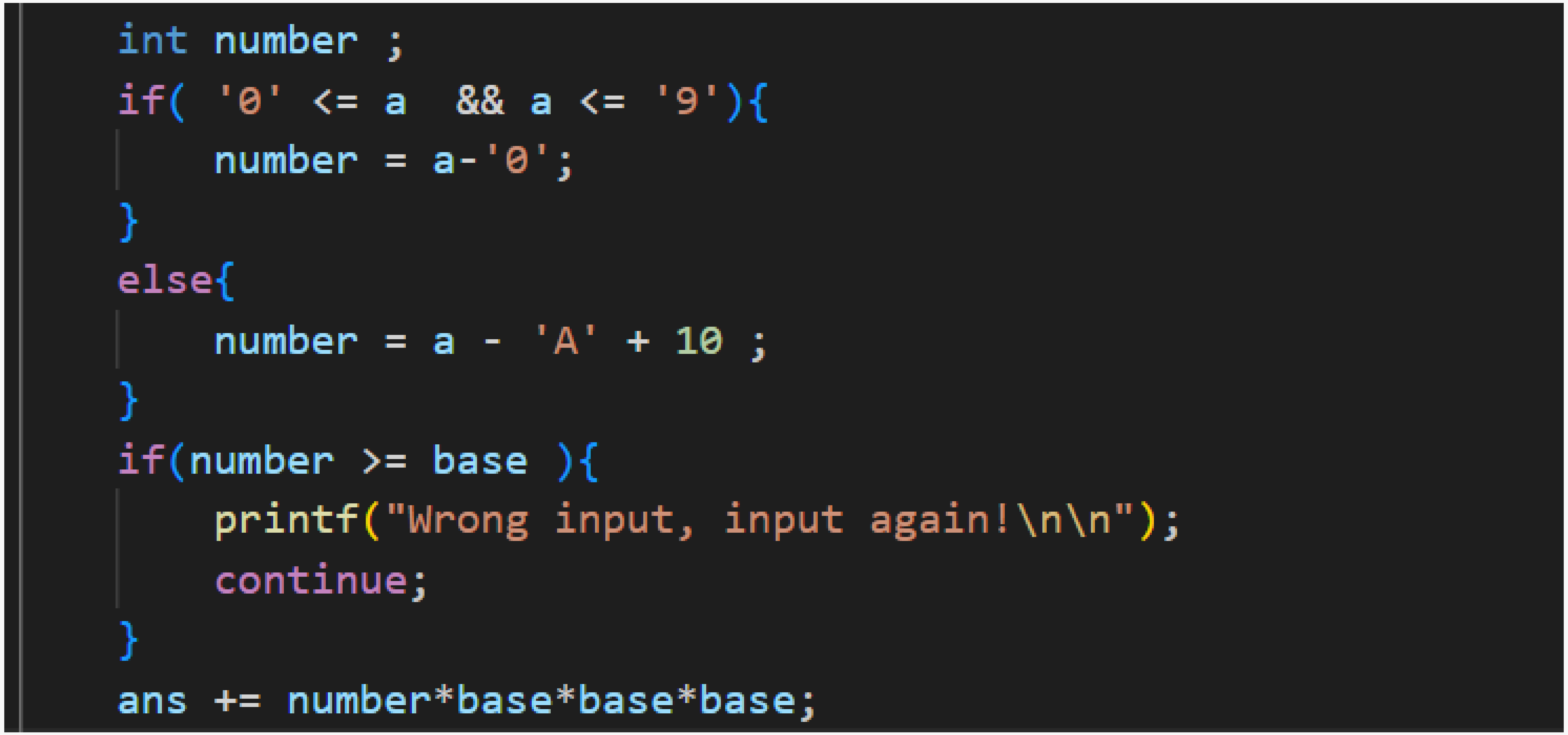
```
Please input a pair of string and integer: (1112,2)
Wrong input, input again!

Please input a pair of string and integer: (ABCD,10)
Wrong input, input again!

Please input a pair of string and integer: (FFF6,16)
FFF6 in base 16 is 65526.

Please input a pair of string and integer: (9CE1,15)
9CE1 in base 15 is 33286.

Please input a pair of string and integer: (0000,0)
Go to next question
```

Number base : example



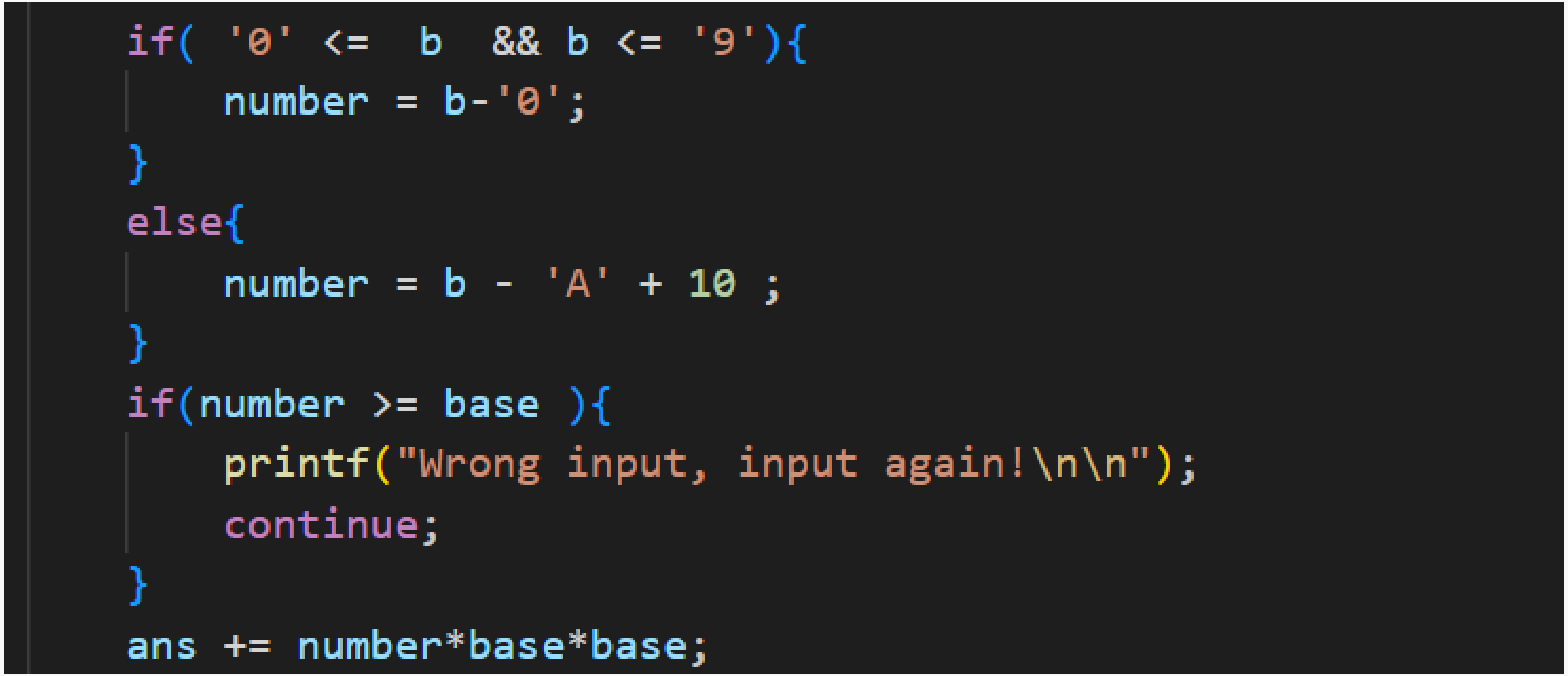
```
while(1){
    printf("Please input a pair of string and integer: ");
    char a , b , c , d;
    int base ;
    scanf(" (%c%c%c%c,%d)", &a, &b, &c, &d, &base);
    if(a == b && b == c && c == d && a == '0' && base == 0 ){
        printf("Go to next question\n\n");
        break;
    }
    int ans = 0 ;
```

Number base : example



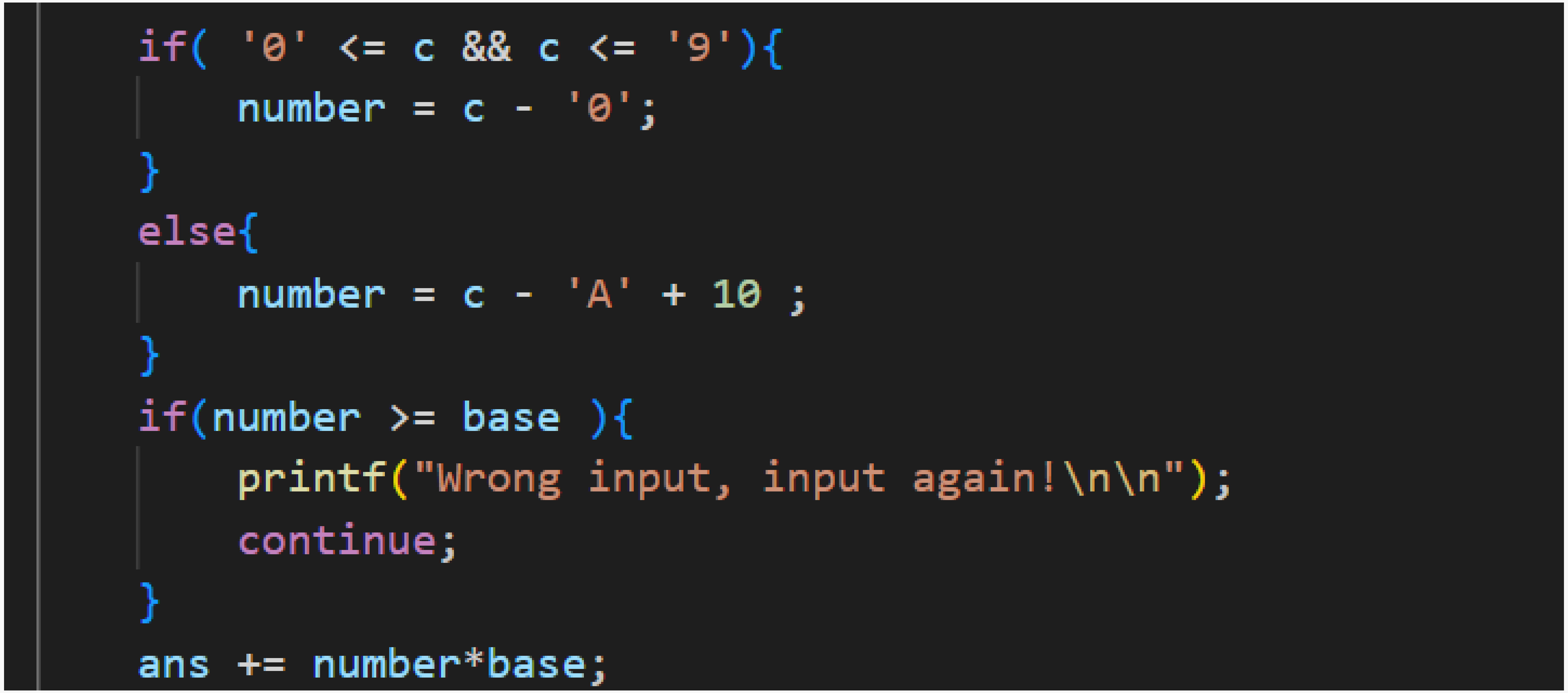
```
int number ;
if( '0' <= a  && a <= '9'){
    number = a-'0';
}
else{
    number = a - 'A' + 10 ;
}
if(number >= base ){
    printf("Wrong input, input again!\n\n");
    continue;
}
ans += number*base*base*base;
```

Number base : example



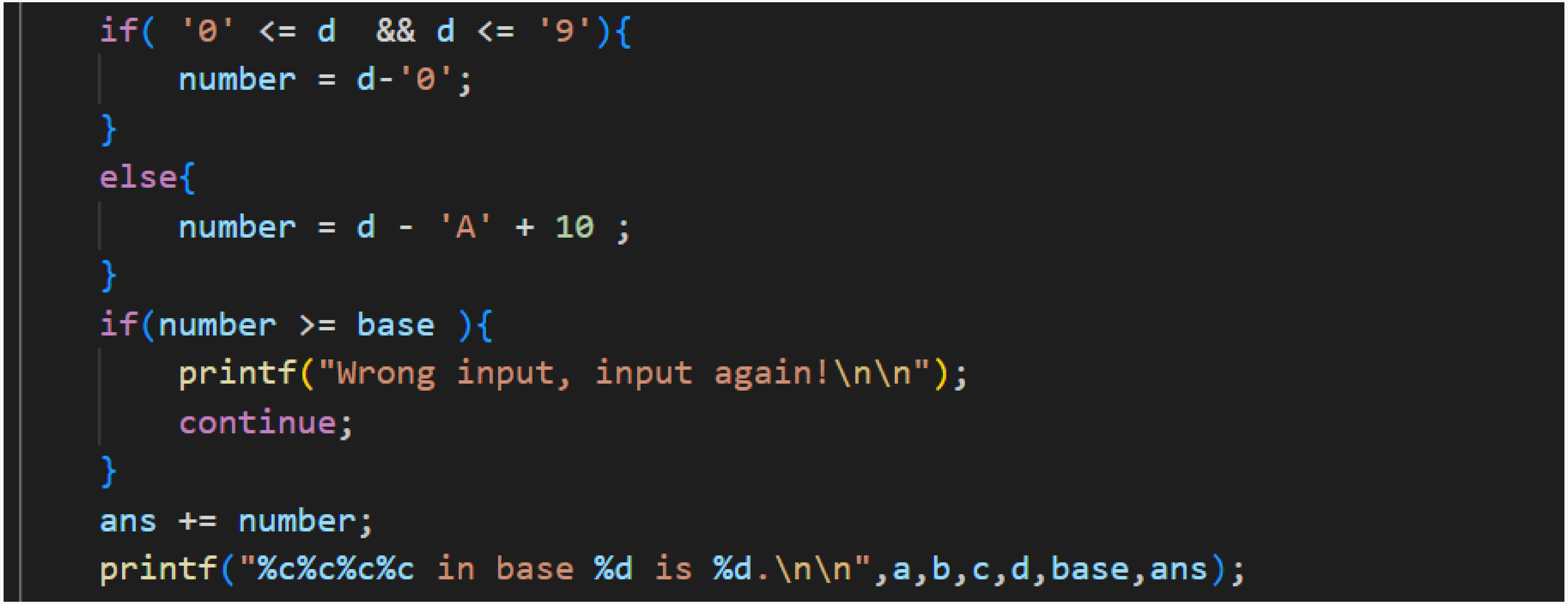
```
if( '0' <=  b  && b <= '9'){
    number = b-'0';
}
else{
    number = b - 'A' + 10 ;
}
if(number >= base ){
    printf("Wrong input, input again!\n\n");
    continue;
}
ans += number*base*base;
```

Number base : example



```
if( '0' <= c && c <= '9'){
    number = c - '0';
}
else{
    number = c - 'A' + 10 ;
}
if(number >= base ){
    printf("Wrong input, input again!\n\n");
    continue;
}
ans += number*base;
```

Number base : example

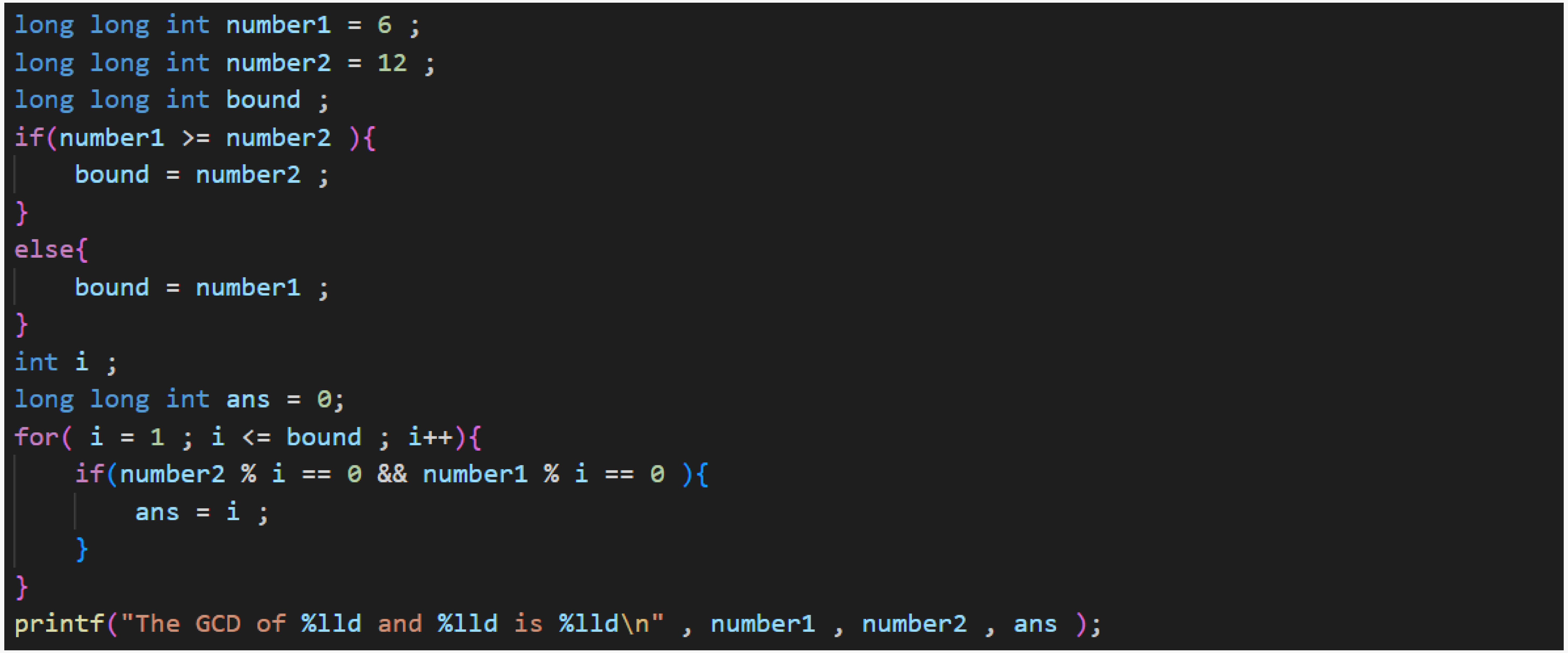


```
if( '0' <= d  && d <= '9'){
    number = d-'0';
}
else{
    number = d - 'A' + 10 ;
}
if(number >= base ){
    printf("Wrong input, input again!\n\n");
    continue;
}
ans += number;
printf("%c%c%c%c in base %d is %d.\n\n",a,b,c,d,base,ans);
```

2. LCM and GCD

LCM and GCD

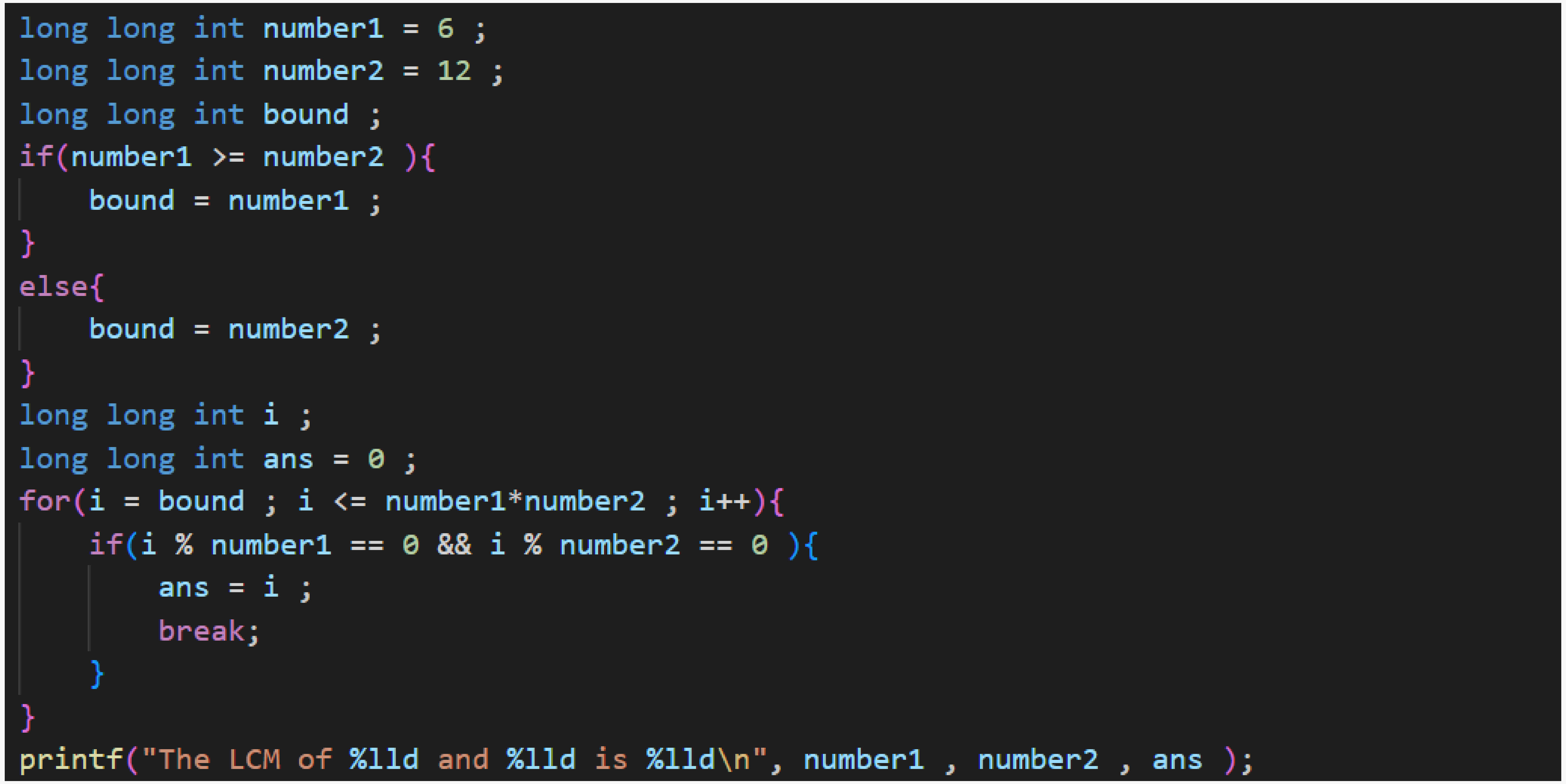
GCD : the Greatest Common Divisor of two number



```
long long int number1 = 6 ;
long long int number2 = 12 ;
long long int bound ;
if(number1 >= number2 ){
    bound = number2 ;
}
else{
    bound = number1 ;
}
int i ;
long long int ans = 0;
for( i = 1 ; i <= bound ; i++){
    if(number2 % i == 0 && number1 % i == 0 ){
        ans = i ;
    }
}
printf("The GCD of %lld and %lld is %lld\n" , number1 , number2 , ans );
```

LCM and GCD

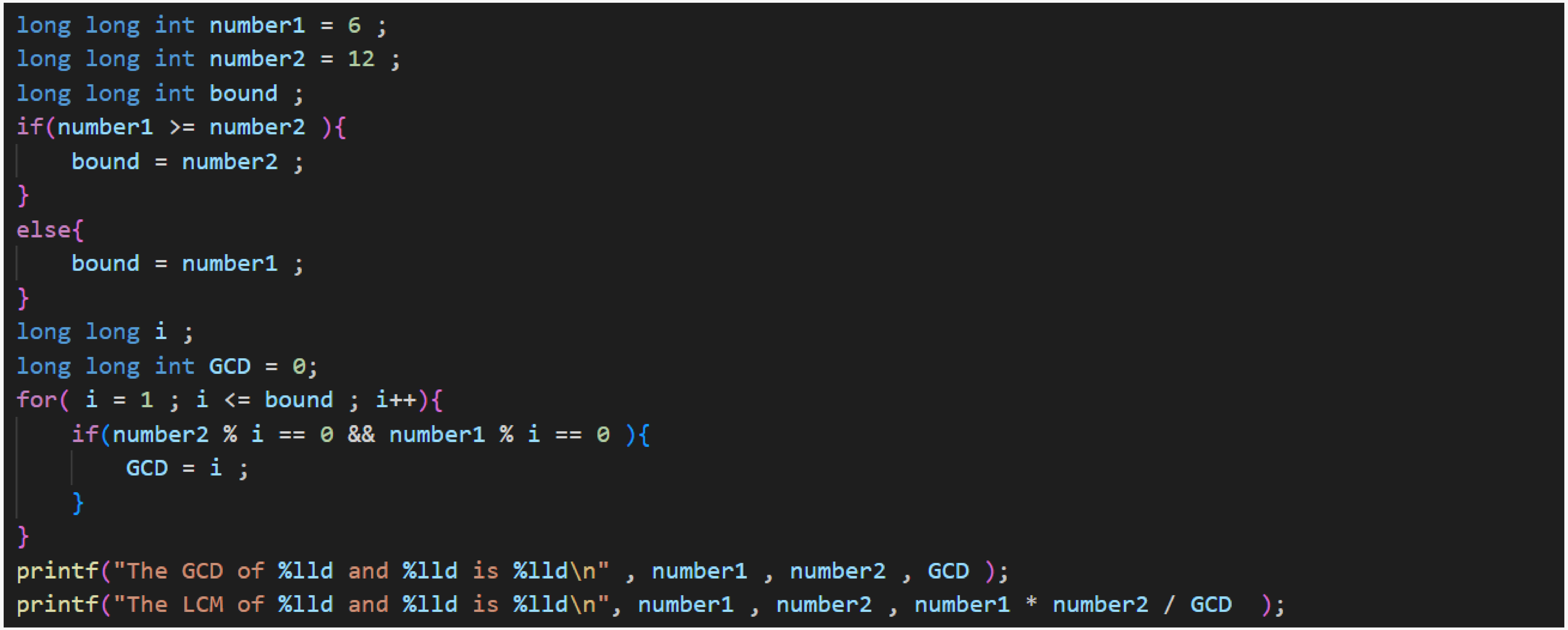
LCM : the Least Common multiple of two number



```c
long long int number1 = 6 ;
long long int number2 = 12 ;
long long int bound ;
if(number1 >= number2 ){
    bound = number1 ;
}
else{
    bound = number2 ;
}
long long int i ;
long long int ans = 0 ;
for(i = bound ; i <= number1*number2 ; i++){
    if(i % number1 == 0 && i % number2 == 0 ){
        ans = i ;
        break;
    }
}
printf("The LCM of %lld and %lld is %lld\n", number1 , number2 , ans );
```

LCM and GCD

LCM = number1 * number2 / GCD

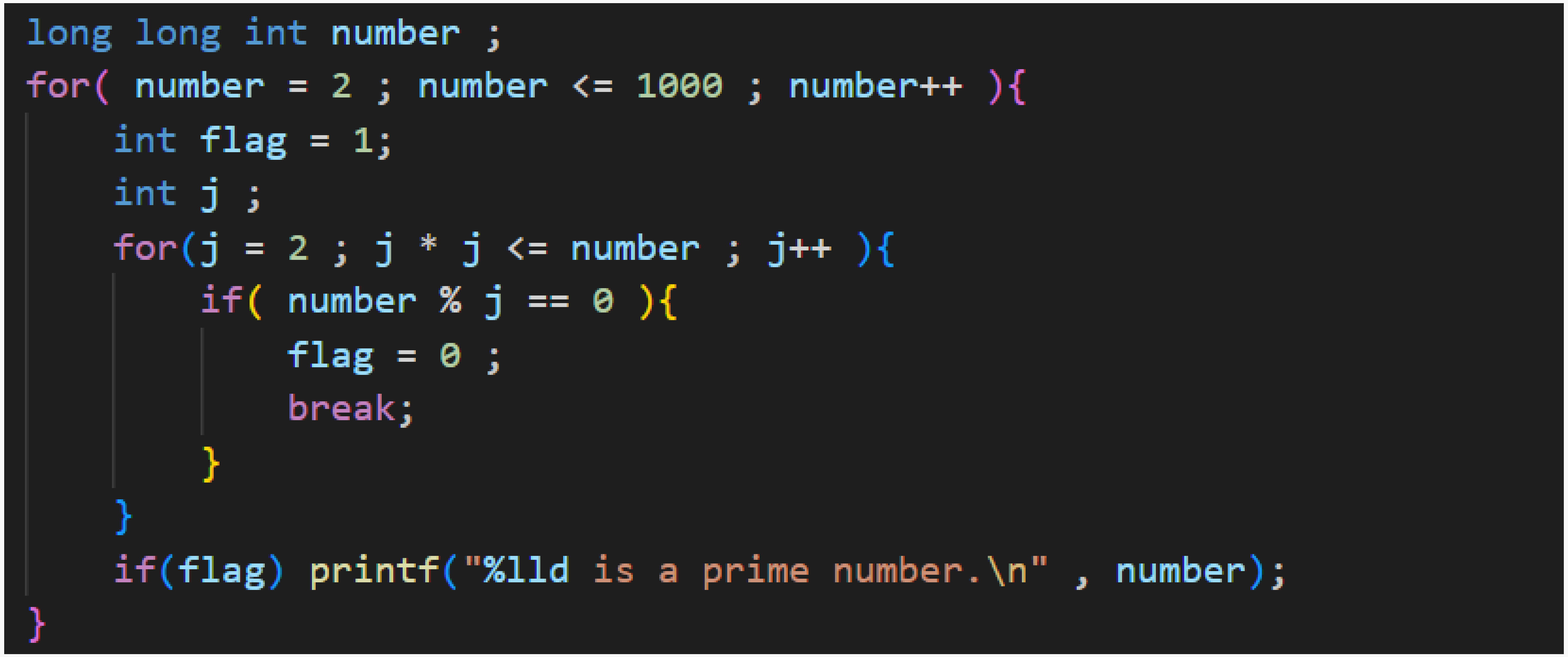


```c
long long int number1 = 6 ;
long long int number2 = 12 ;
long long int bound ;
if(number1 >= number2 ){
    bound = number2 ;
}
else{
    bound = number1 ;
}
long long i ;
long long int GCD = 0;
for( i = 1 ; i <= bound ; i++){
    if(number2 % i == 0 && number1 % i == 0 ){
        GCD = i ;
    }
}
printf("The GCD of %lld and %lld is %lld\n" , number1 , number2 , GCD );
printf("The LCM of %lld and %lld is %lld\n", number1 , number2 , number1 * number2 / GCD  );
```

3. Prime Number

Prime number

Since the factors of the number are composed of lots of pair.



```c
long long int number ;
for( number = 2 ; number <= 1000 ; number++ ){
    int flag = 1;
    int j ;
    for(j = 2 ; j * j <= number ; j++ ){
        if( number % j == 0 ){
            flag = 0 ;
            break;
        }
    }
    if(flag) printf("%lld is a prime number.\n" , number);
}
```

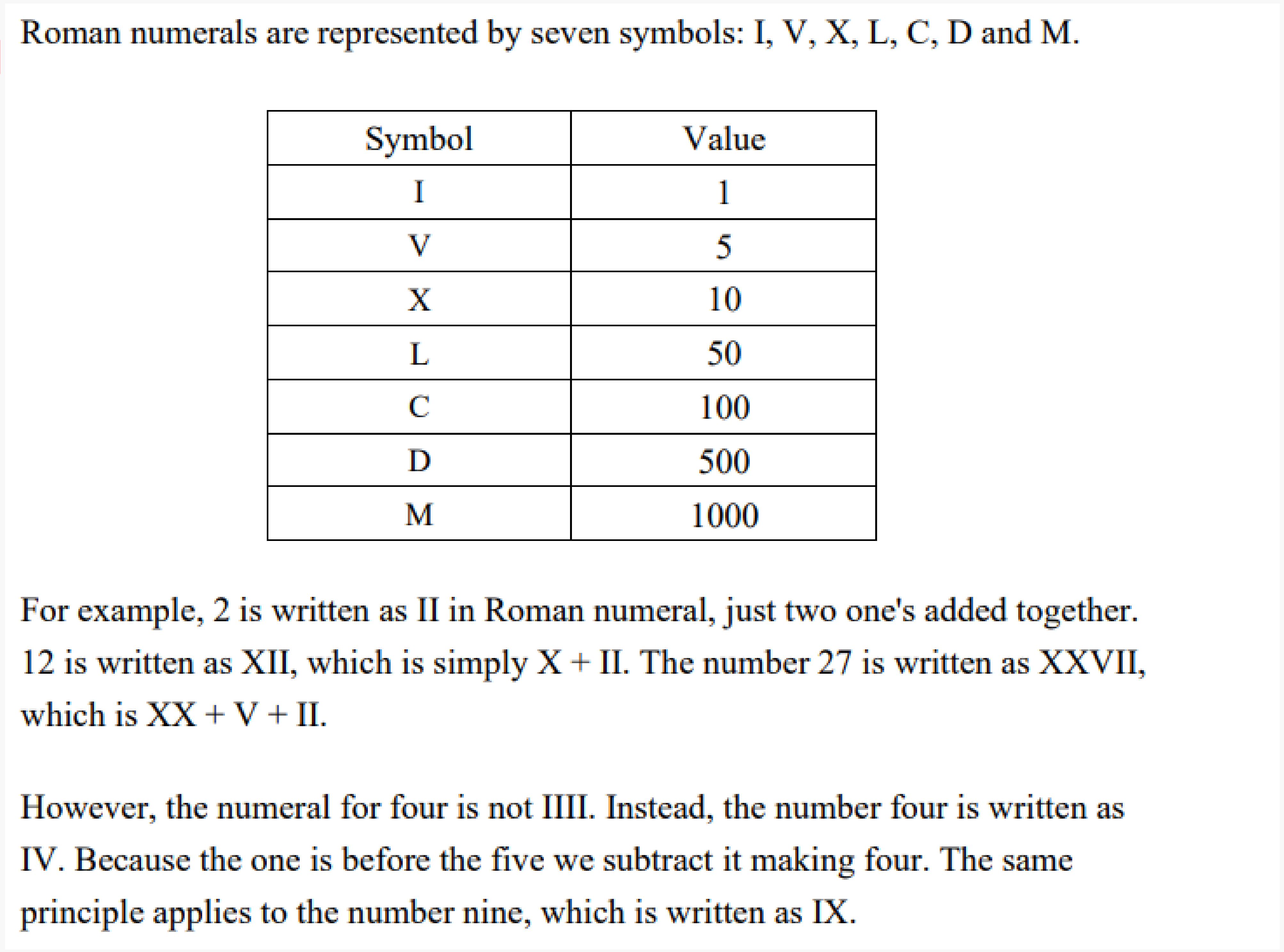
Topic : Some classic problem

1. Roman Numerals
2. Pascal's Triangle
3. Palindrome
4. Calendar
5. Print Special Pattern

1. Roman Numerals

Roman Numerals

112-a



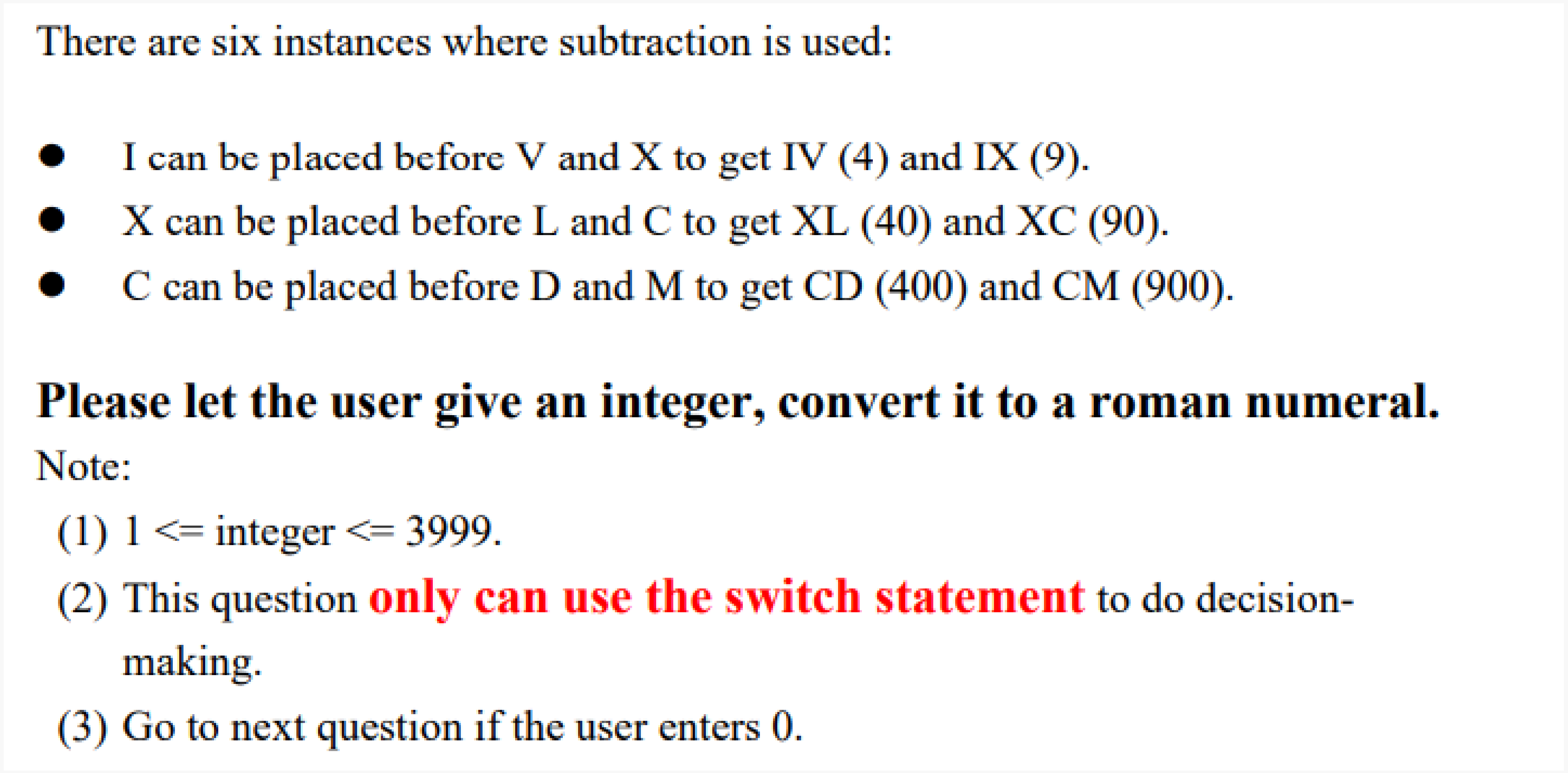
Roman numerals are represented by seven symbols: I, V, X, L, C, D and M.

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

For example, 2 is written as II in Roman numeral, just two one's added together. 12 is written as XII, which is simply X + II. The number 27 is written as XXVII, which is XX + V + II.

However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as IX.

Roman Numerals



There are six instances where subtraction is used:

- I can be placed before V and X to get IV (4) and IX (9).
- X can be placed before L and C to get XL (40) and XC (90).
- C can be placed before D and M to get CD (400) and CM (900).

Please let the user give an integer, convert it to a roman numeral.

Note:

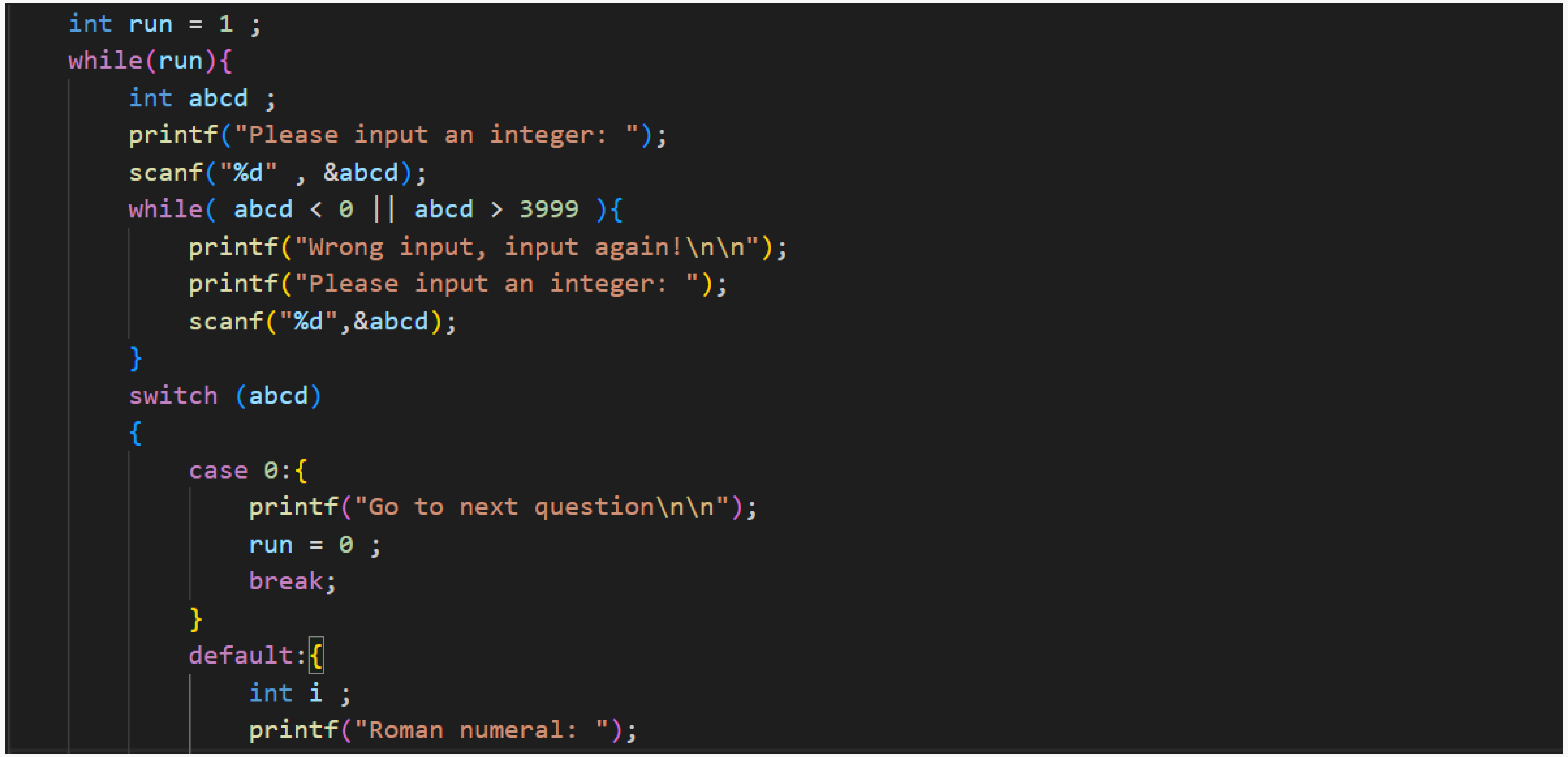
(1) 1 <= integer <= 3999.

(2) This question **only can use the switch statement** to do decision-making.

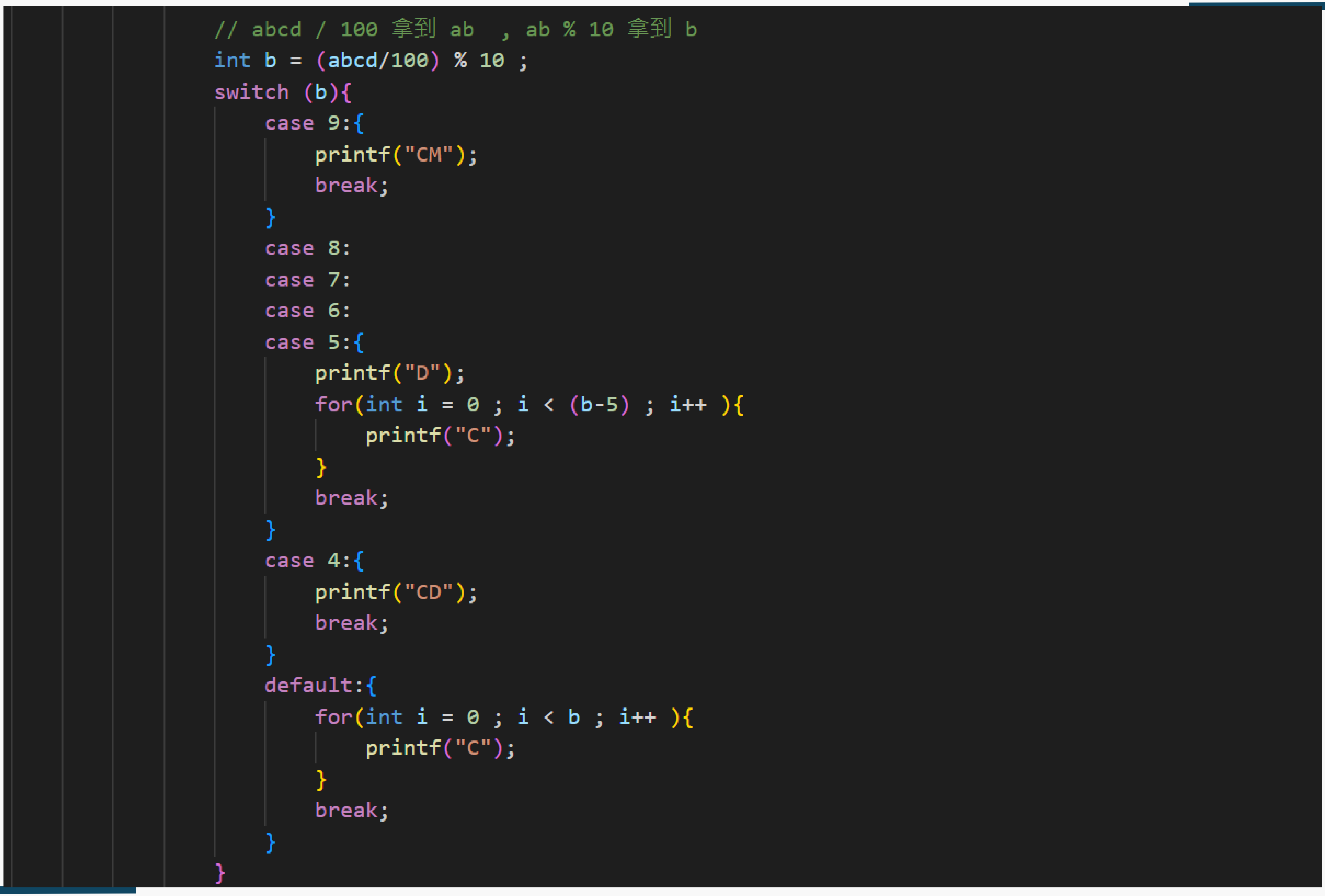
(3) Go to next question if the user enters 0.

Roman Numerals

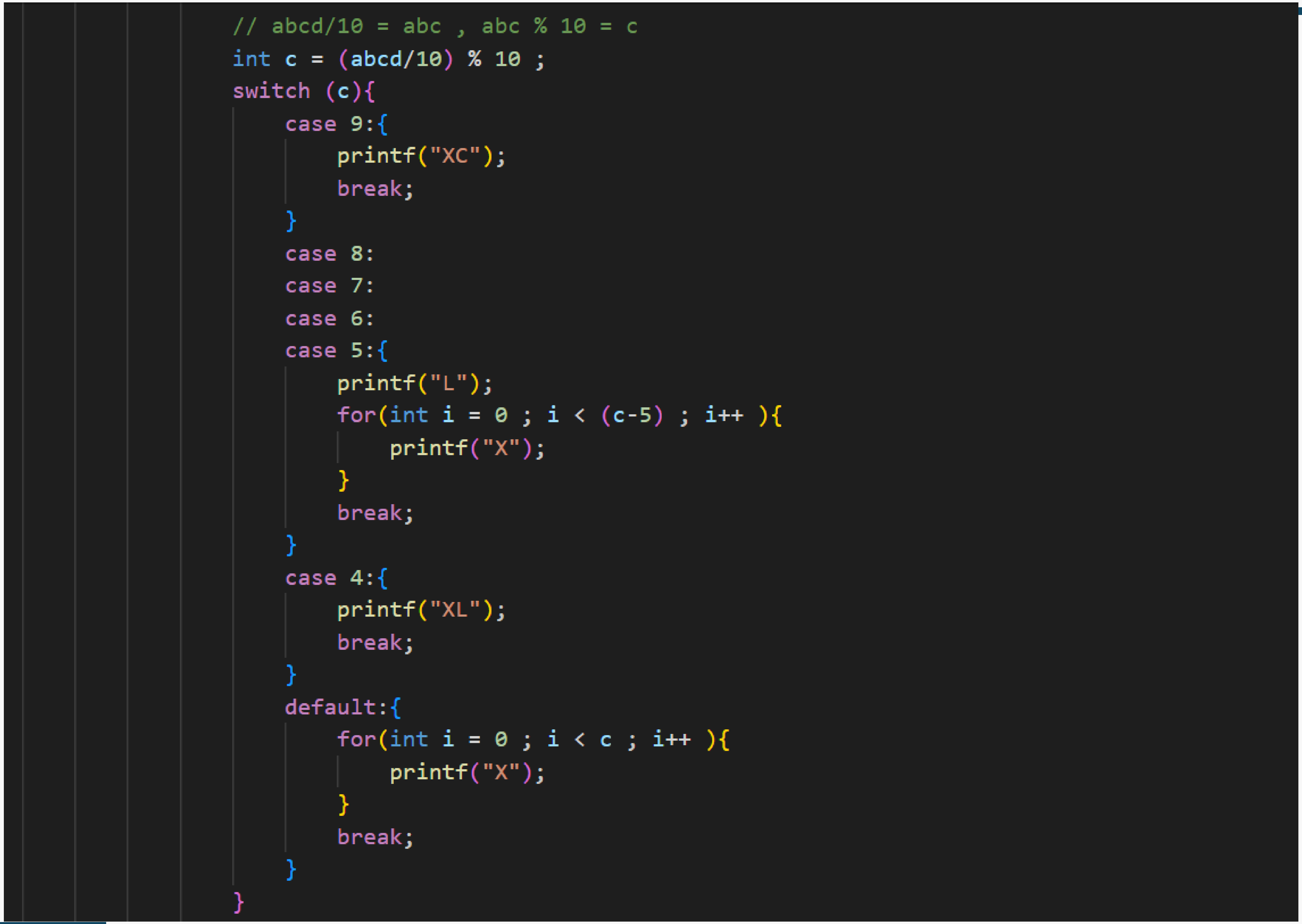
The method of solving this is convert it to a*1000+b*100+c*10+d and use if/else or switch to check output.



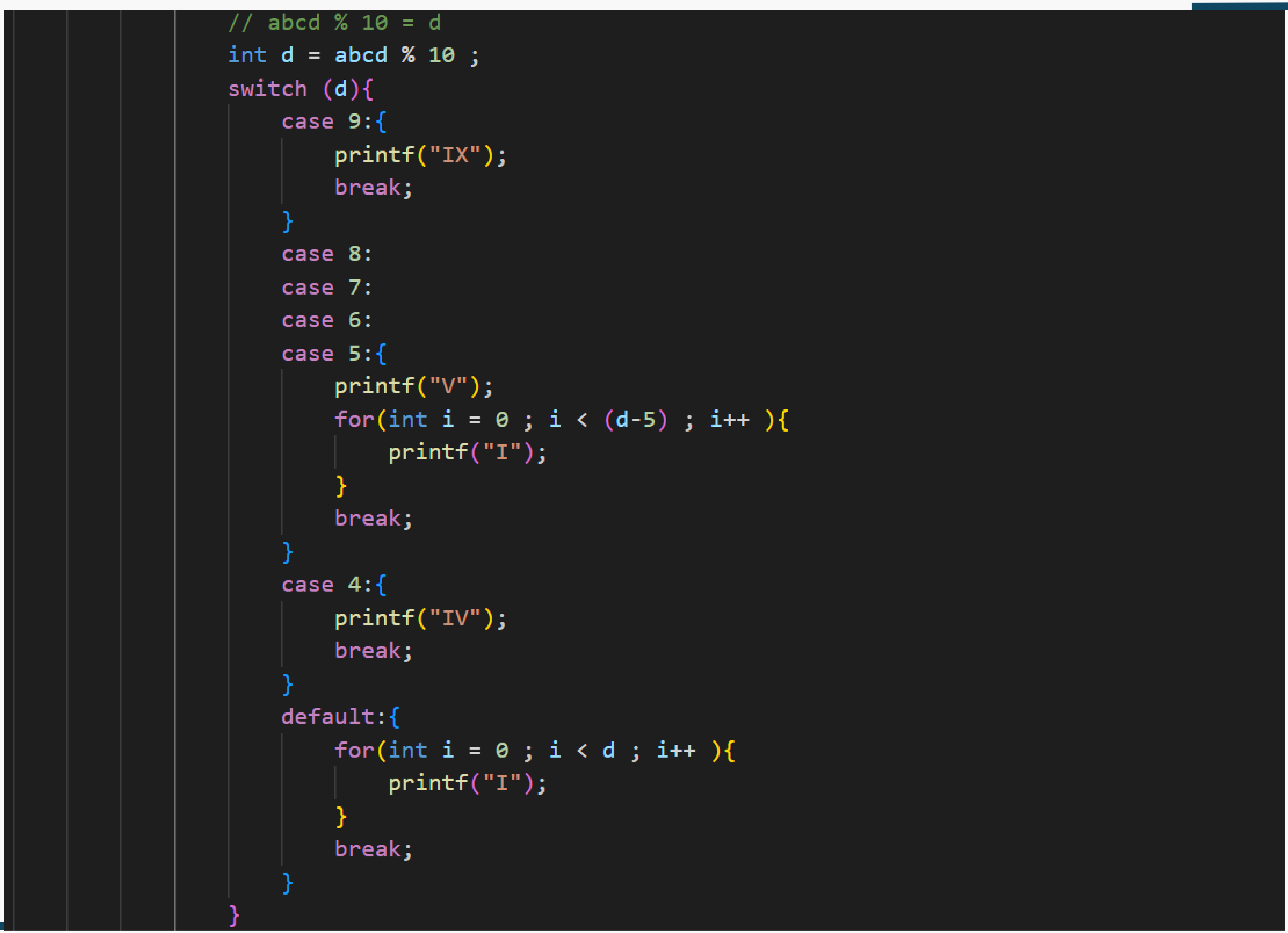
```
int run = 1 ;
while(run){
    int abcd ;
    printf("Please input an integer: ");
    scanf("%d" , &abcd);
    while( abcd < 0 || abcd > 3999 ){
        printf("Wrong input, input again!\n\n");
        printf("Please input an integer: ");
        scanf("%d",&abcd);
    }
    switch (abcd)
    {
        case 0:{
            printf("Go to next question\n\n");
            run = 0 ;
            break;
        }
        default:{
            int i ;
            printf("Roman numeral: ");
```



```c
// abcd / 100 拿到 ab  , ab % 10 拿到 b
int b = (abcd/100) % 10 ;
switch (b){
    case 9:{
        printf("CM");
        break;
    }
    case 8:
    case 7:
    case 6:
    case 5:{
        printf("D");
        for(int i = 0 ; i < (b-5) ; i++ ){
            printf("C");
        }
        break;
    }
    case 4:{
        printf("CD");
        break;
    }
    default:{
        for(int i = 0 ; i < b ; i++ ){
            printf("C");
        }
        break;
    }
}
```



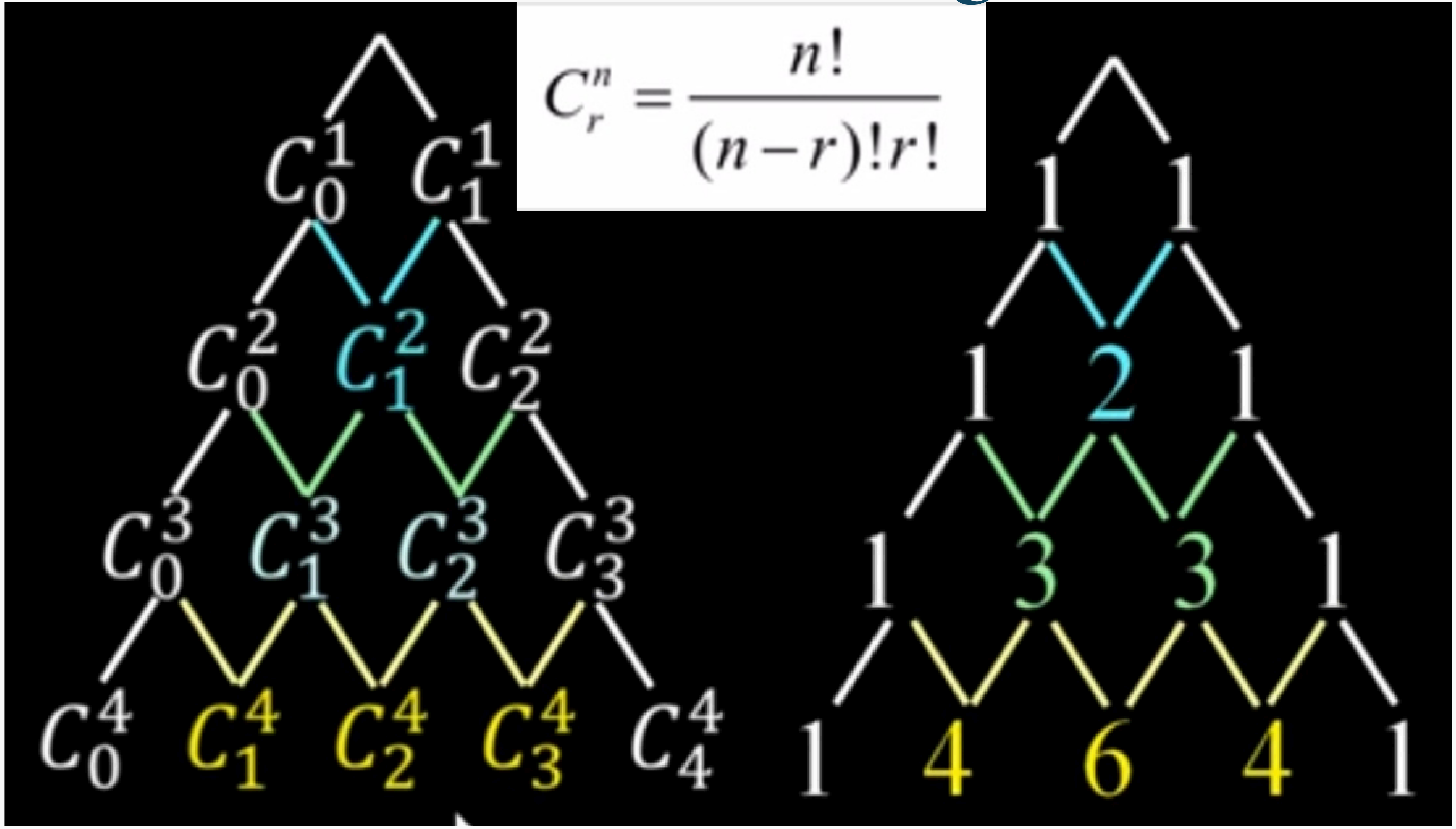
```
// abcd/10 = abc , abc % 10 = c
int c = (abcd/10) % 10 ;
switch (c){
    case 9:{
        printf("XC");
        break;
    }
    case 8:
    case 7:
    case 6:
    case 5:{
        printf("L");
        for(int i = 0 ; i < (c-5) ; i++ ){
            printf("X");
        }
        break;
    }
    case 4:{
        printf("XL");
        break;
    }
    default:{
        for(int i = 0 ; i < c ; i++ ){
            printf("X");
        }
        break;
    }
}
```



```c
// abcd % 10 = d
int d = abcd % 10 ;
switch (d){
    case 9:{
        printf("IX");
        break;
    }
    case 8:
    case 7:
    case 6:
    case 5:{
        printf("V");
        for(int i = 0 ; i < (d-5) ; i++ ){
            printf("I");
        }
        break;
    }
    case 4:{
        printf("IV");
        break;
    }
    default:{
        for(int i = 0 ; i < d ; i++ ){
            printf("I");
        }
        break;
    }
}
```

2. *Pascal's Triangle*

Pascal's Triangle

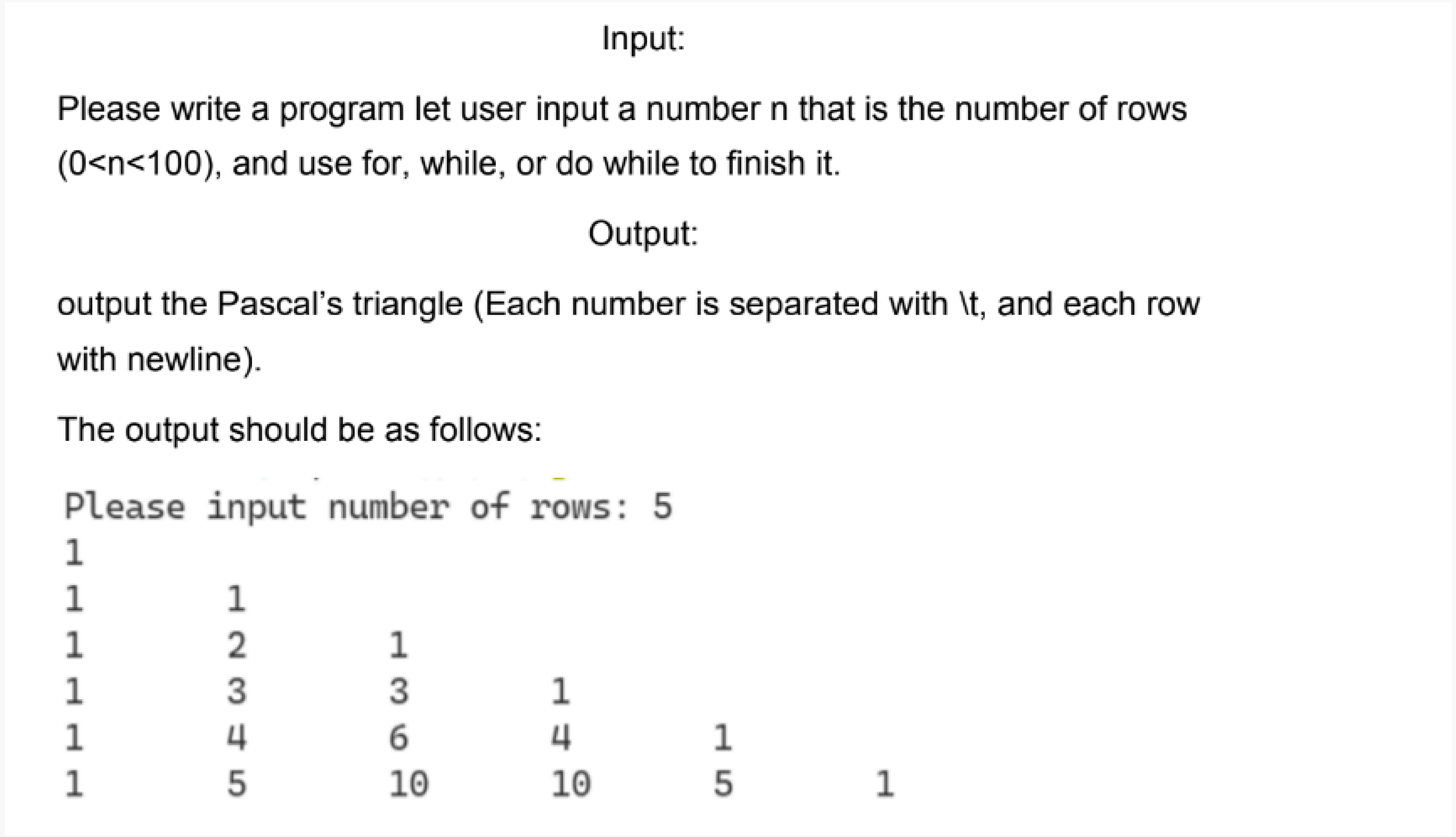


$C_0^1 \quad C_1^1$
$C_0^2 \quad C_1^2 \quad C_2^2$
$C_0^3 \quad C_1^3 \quad C_2^3 \quad C_3^3$
$C_0^4 \quad C_1^4 \quad C_2^4 \quad C_3^4 \quad C_4^4$
1 1
1 2 1
1 3 3 1
1 4 6 4 1

$$C_r^n = \frac{n!}{(n-r)!r!}$$

Pascal's Triangle

110-4



Input:

Please write a program let user input a number n that is the number of rows (0<n<100), and use for, while, or do while to finish it.

Output:

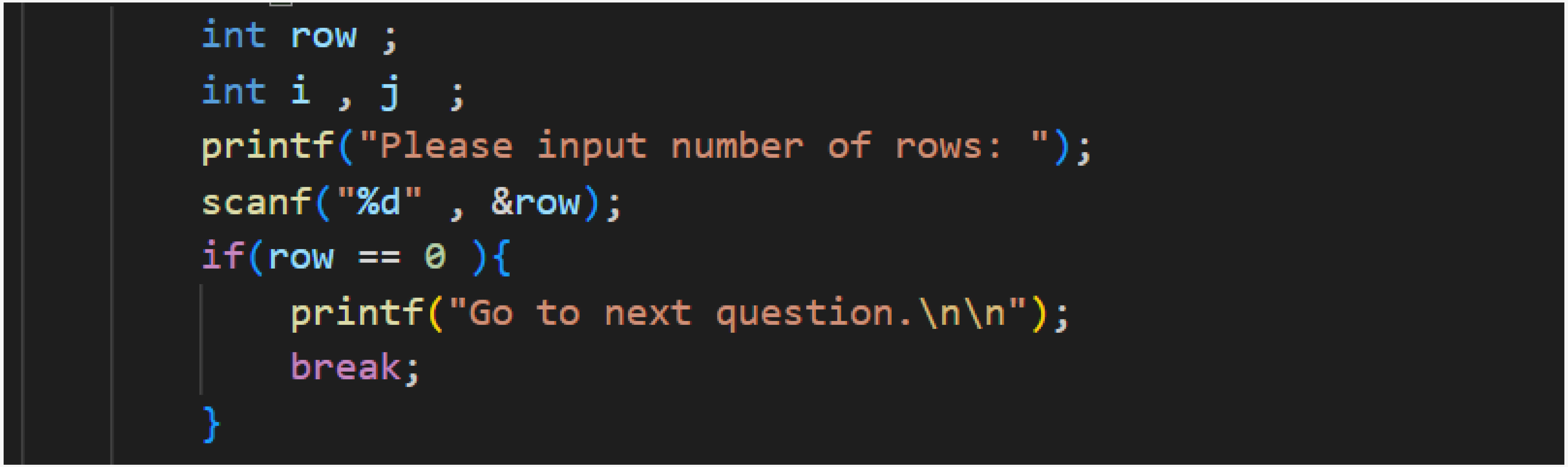
output the Pascal's triangle (Each number is separated with \t, and each row with newline).

The output should be as follows:

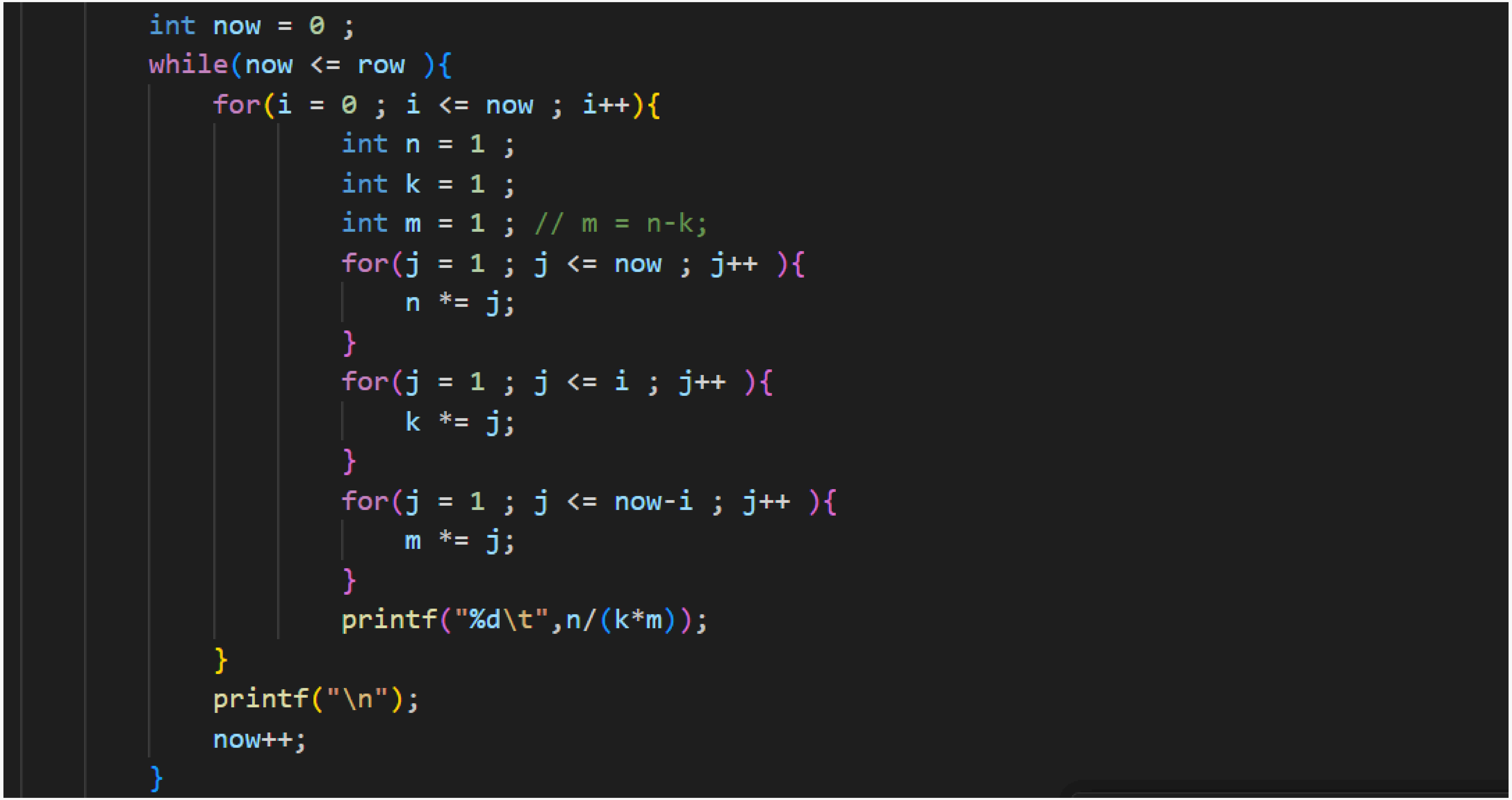
```
Please input number of rows: 5
1
1       1
1       2       1
1       3       3       1
1       4       6       4       1
1       5       10      10      5       1
```

Pascal's Triangle

Method: Use for/while loop to calculate each row



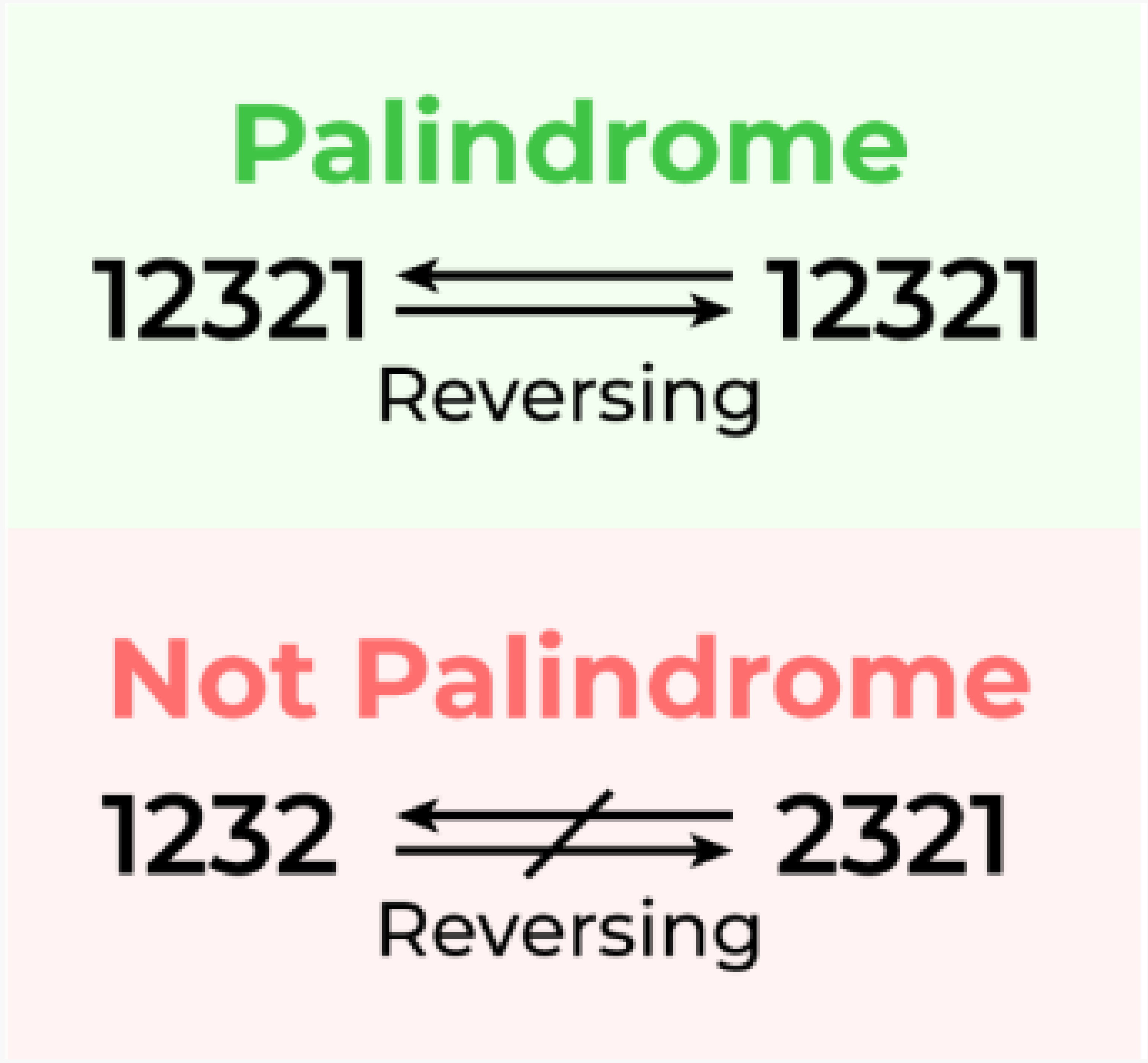
```
int row ;
int i , j  ;
printf("Please input number of rows: ");
scanf("%d" , &row);
if(row == 0 ){
    printf("Go to next question.\n\n");
    break;
}
```



```
int now = 0 ;
while(now <= row ){
    for(i = 0 ; i <= now ; i++){
            int n = 1 ;
            int k = 1 ;
            int m = 1 ; // m = n-k;
            for(j = 1 ; j <= now ; j++ ){
                n *= j;
            }
            for(j = 1 ; j <= i ; j++ ){
                k *= j;
            }
            for(j = 1 ; j <= now-i ; j++ ){
                m *= j;
            }
            printf("%d\t",n/(k*m));
    }
    printf("\n");
    now++;
}
```

3. Palindrome

Palindrome


Palindrome
12321
12321
Reversing
Not Palindrome
1232
2321
Reversing

Palindrome

Method: the method to reverse the number

123 = 1 * 100 + 2 * 10 + 3 * 1

123 % 10 = 3 --> now_reverse = 0 * 10 + 3

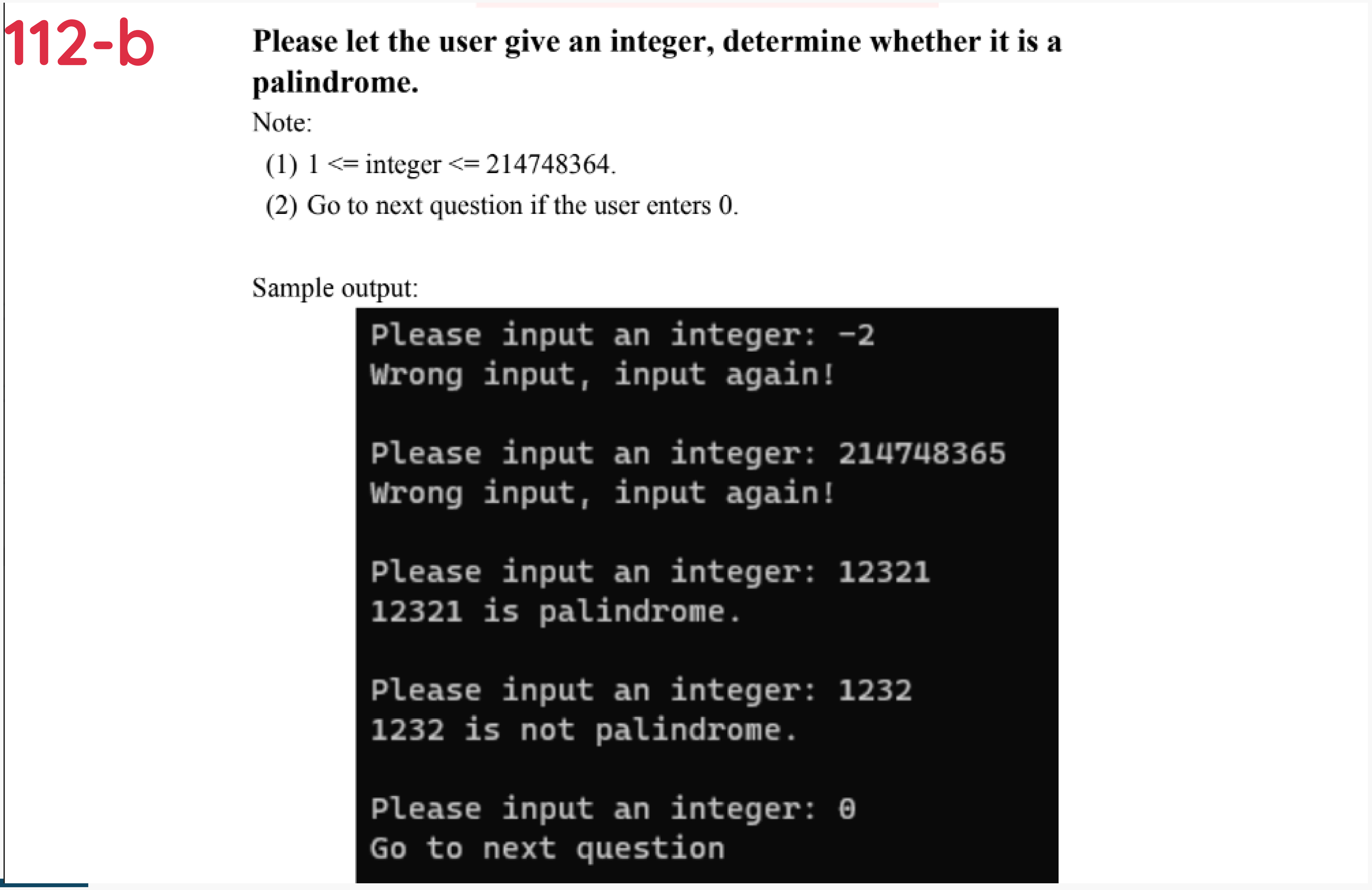
12 % 10 = 2 --> now_reverse = 3 * 10 + 2

1 % 10 = 1 --> now_reverse = 32 * 10 + 1

now_reverse = 321

Palindrome

112-b



Please let the user give an integer, determine whether it is a palindrome.

Note:

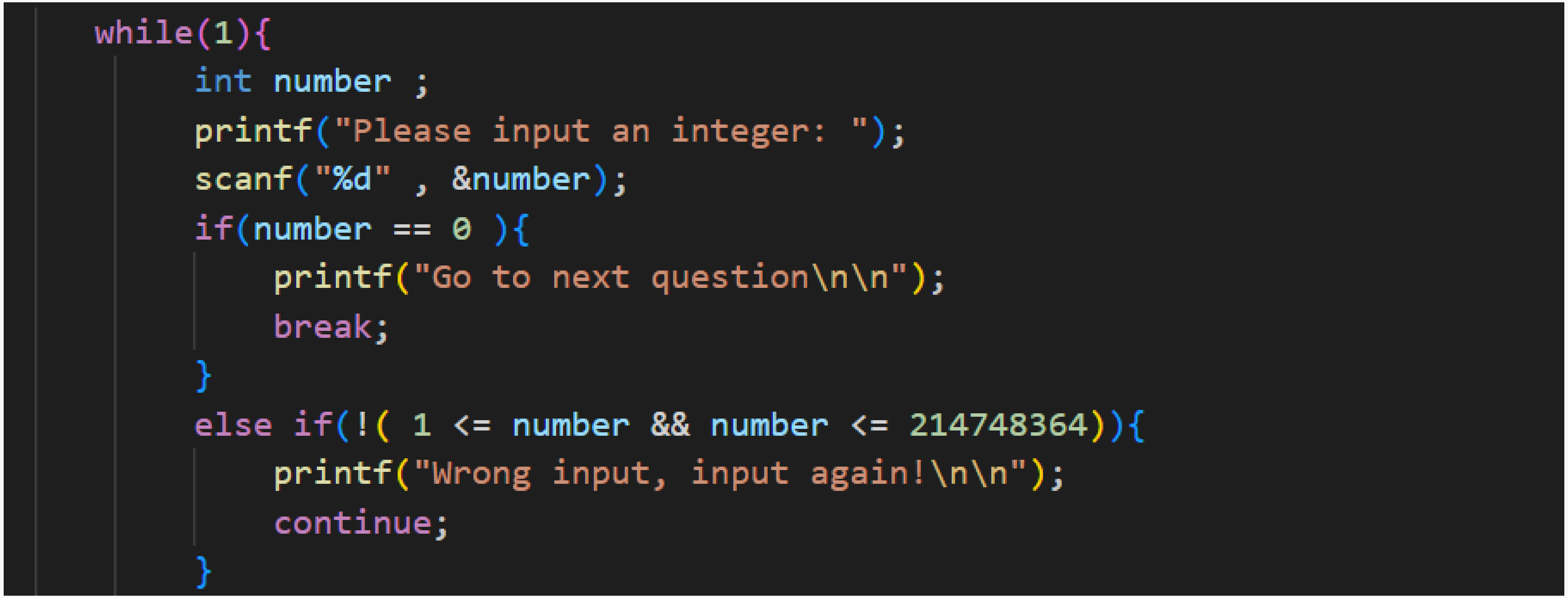
(1) 1 <= integer <= 214748364.

(2) Go to next question if the user enters 0.

Sample output:

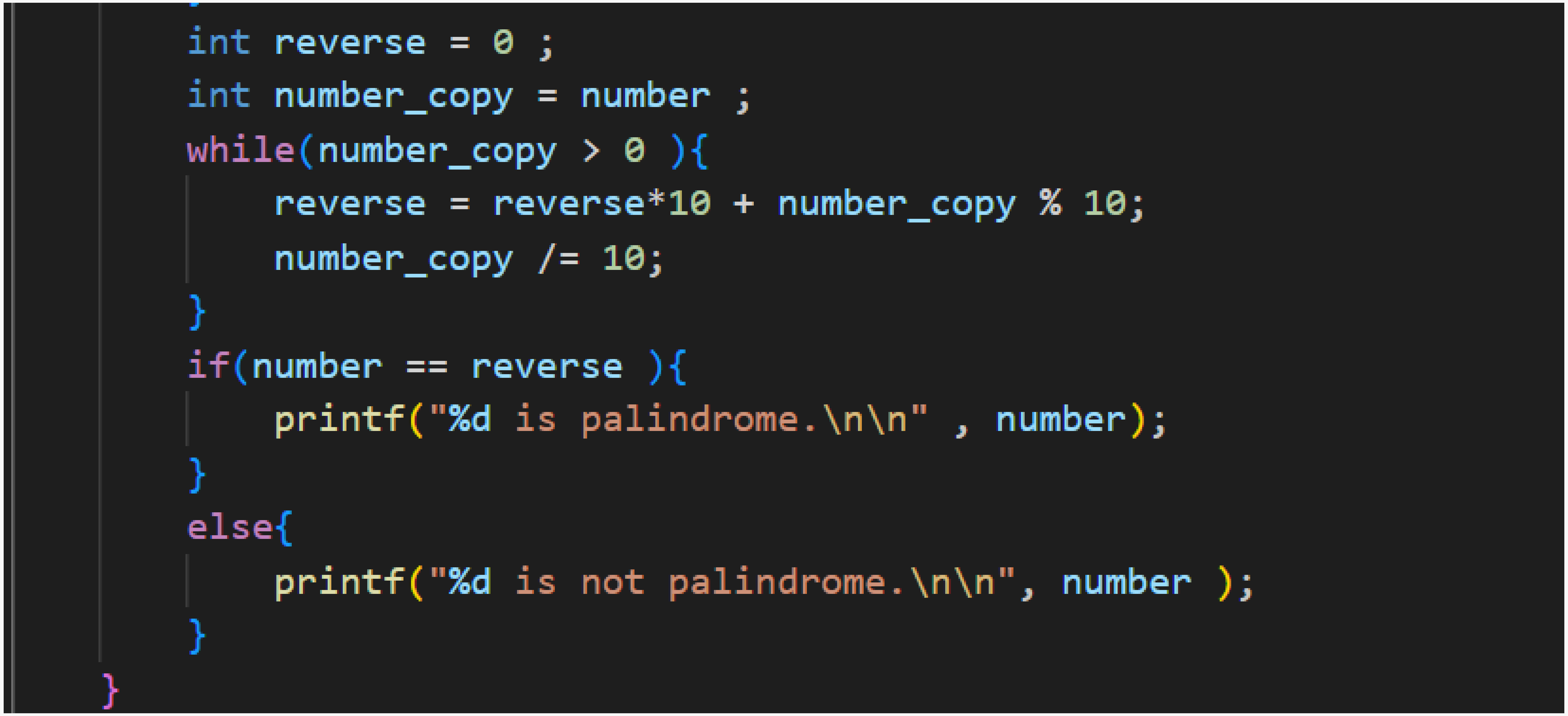
```
Please input an integer: -2
Wrong input, input again!

Please input an integer: 214748365
Wrong input, input again!

Please input an integer: 12321
12321 is palindrome.

Please input an integer: 1232
1232 is not palindrome.

Please input an integer: 0
Go to next question
```

Palindrome



```c
    while(1){
        int number ;
        printf("Please input an integer: ");
        scanf("%d" , &number);
        if(number == 0 ){
            printf("Go to next question\n\n");
            break;
        }
        else if(!( 1 <= number && number <= 214748364)){
            printf("Wrong input, input again!\n\n");
            continue;
        }
```

Palindrome



```
    int reverse = 0 ;
    int number_copy = number ;
    while(number_copy > 0 ){
        reverse = reverse*10 + number_copy % 10;
        number_copy /= 10;
    }
    if(number == reverse ){
        printf("%d is palindrome.\n\n" , number);
    }
    else{
        printf("%d is not palindrome.\n\n", number );
    }
}
```

4. Calendar

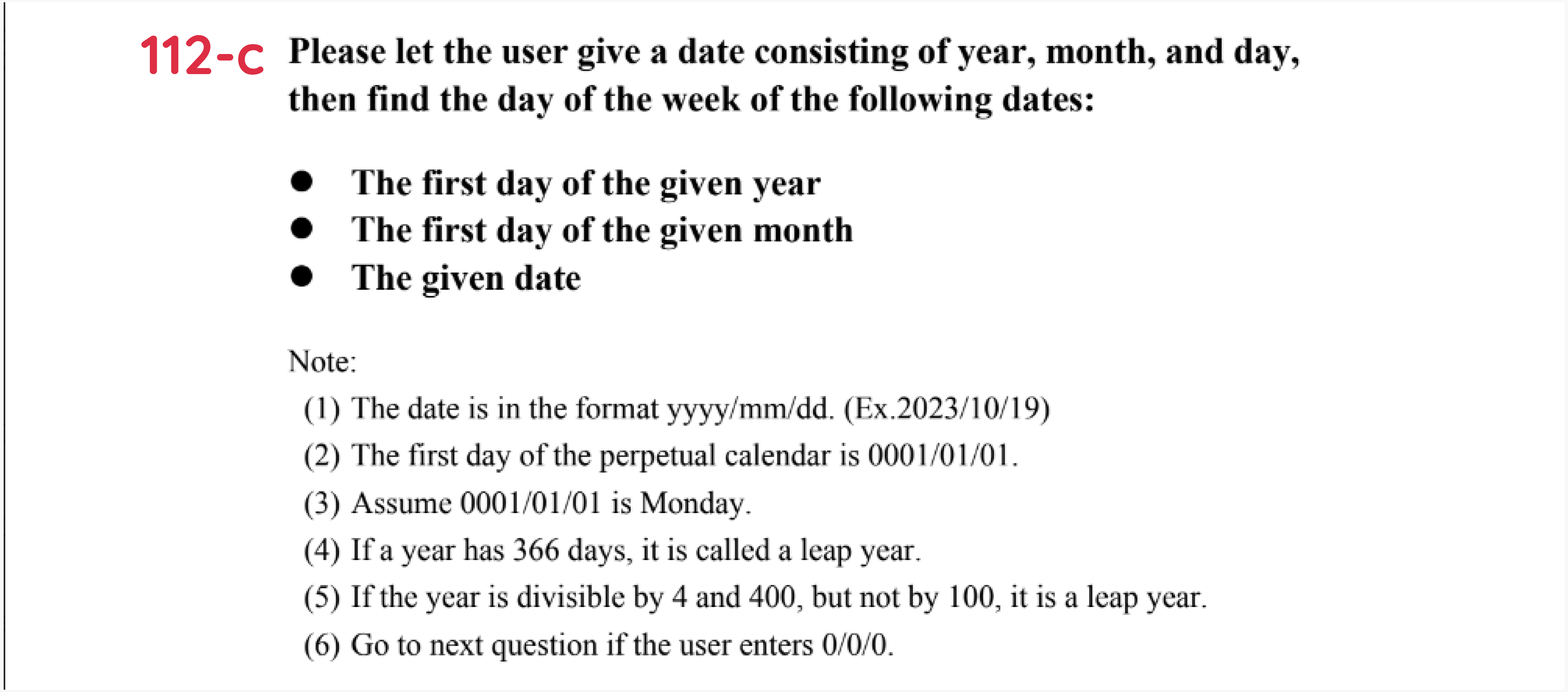
Calendar

How to calculate what day of the week yyyy/mm/dd is?

Ans : I only need to calculate the total day between initial day and the day I wanted.

if 2024/10/21 is Monday, then 2024/10/27 is Sunday
since 27-21 = 6 , 6+1 % 7 = 0 -> Sunday

Calendar



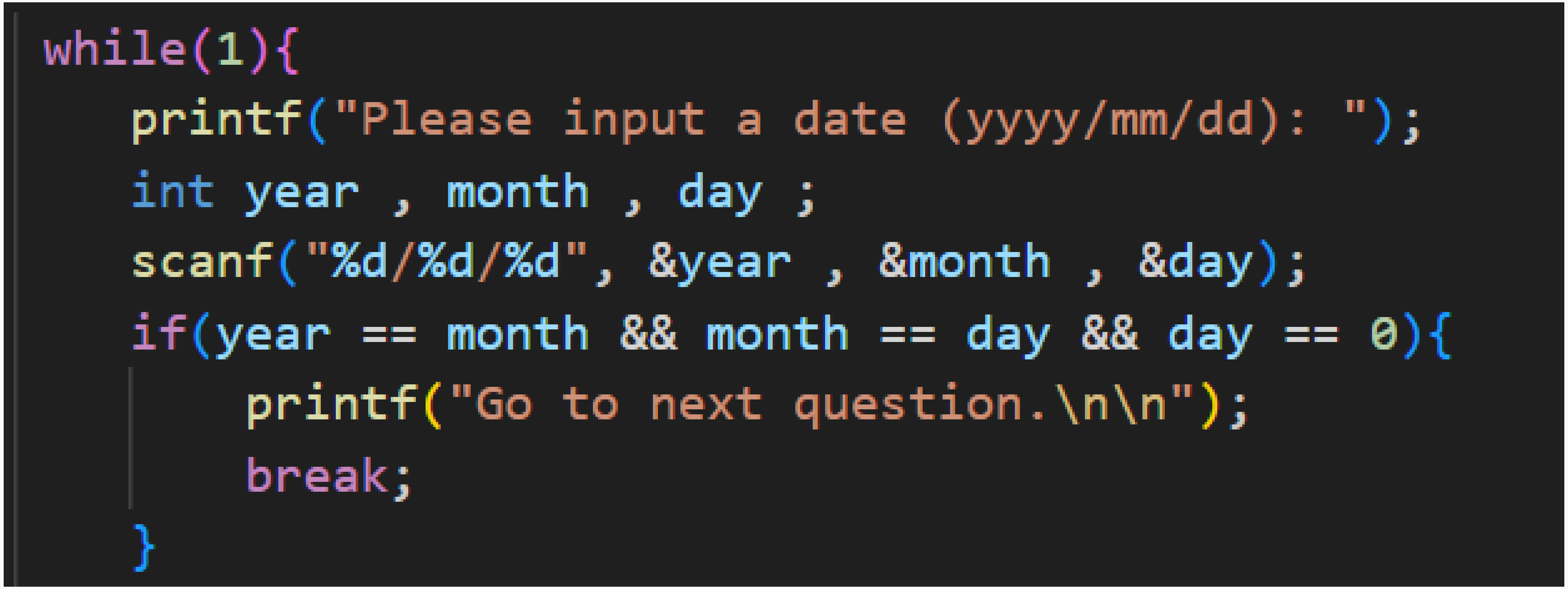
112-c **Please let the user give a date consisting of year, month, and day, then find the day of the week of the following dates:**

- **The first day of the given year**
- **The first day of the given month**
- **The given date**

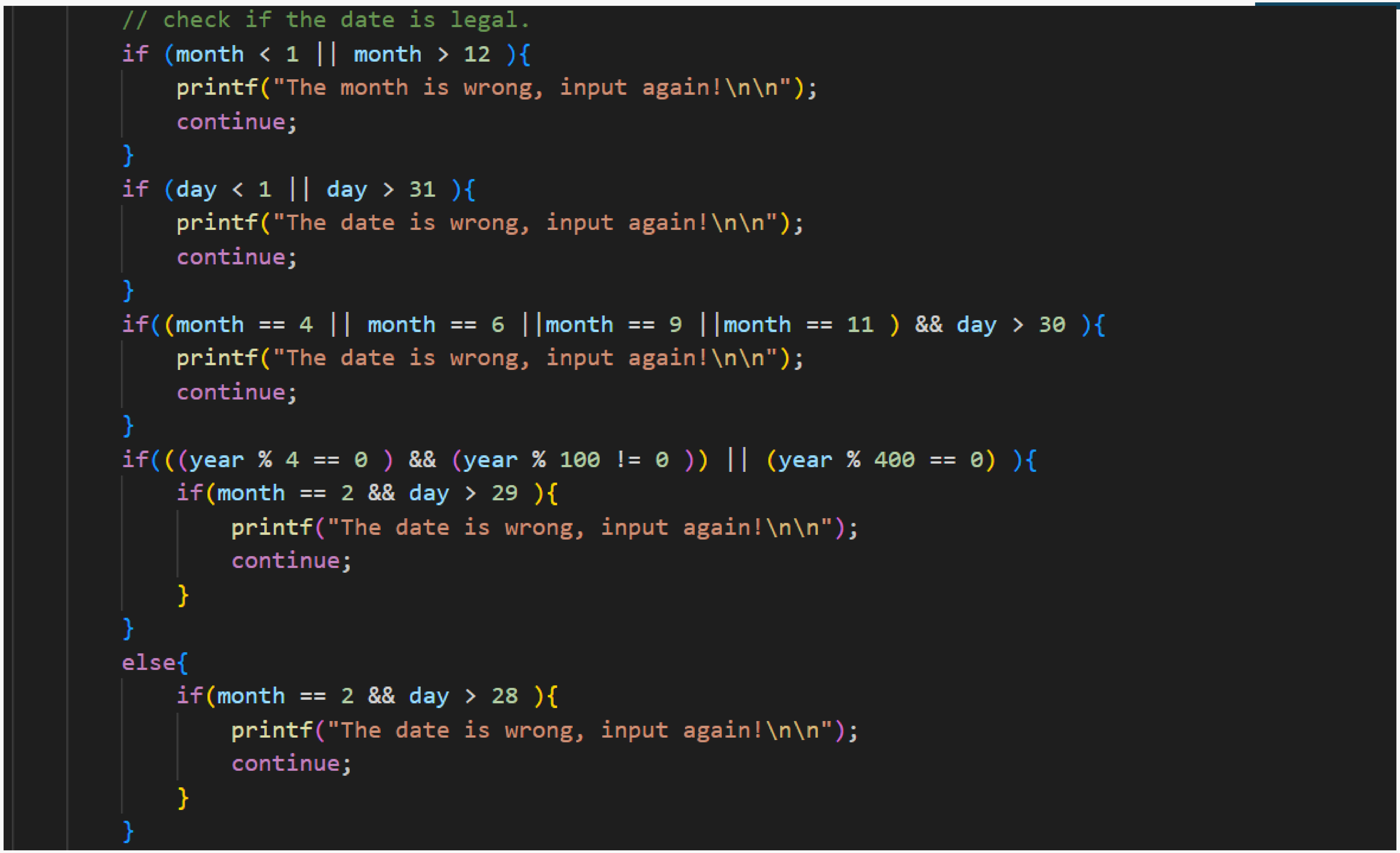
Note:

(1) The date is in the format yyyy/mm/dd. (Ex.2023/10/19)
(2) The first day of the perpetual calendar is 0001/01/01.
(3) Assume 0001/01/01 is Monday.
(4) If a year has 366 days, it is called a leap year.
(5) If the year is divisible by 4 and 400, but not by 100, it is a leap year.
(6) Go to next question if the user enters 0/0/0.

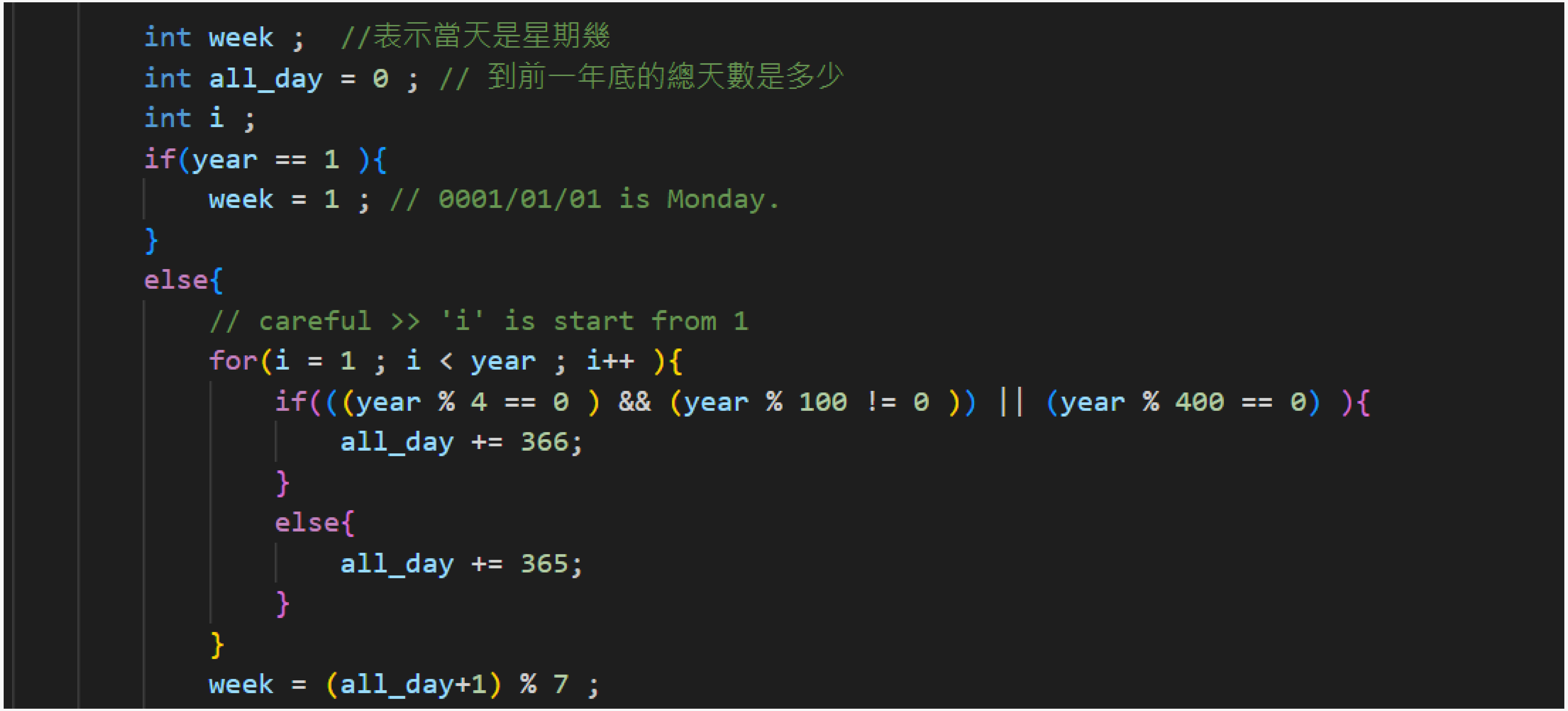
Calendar



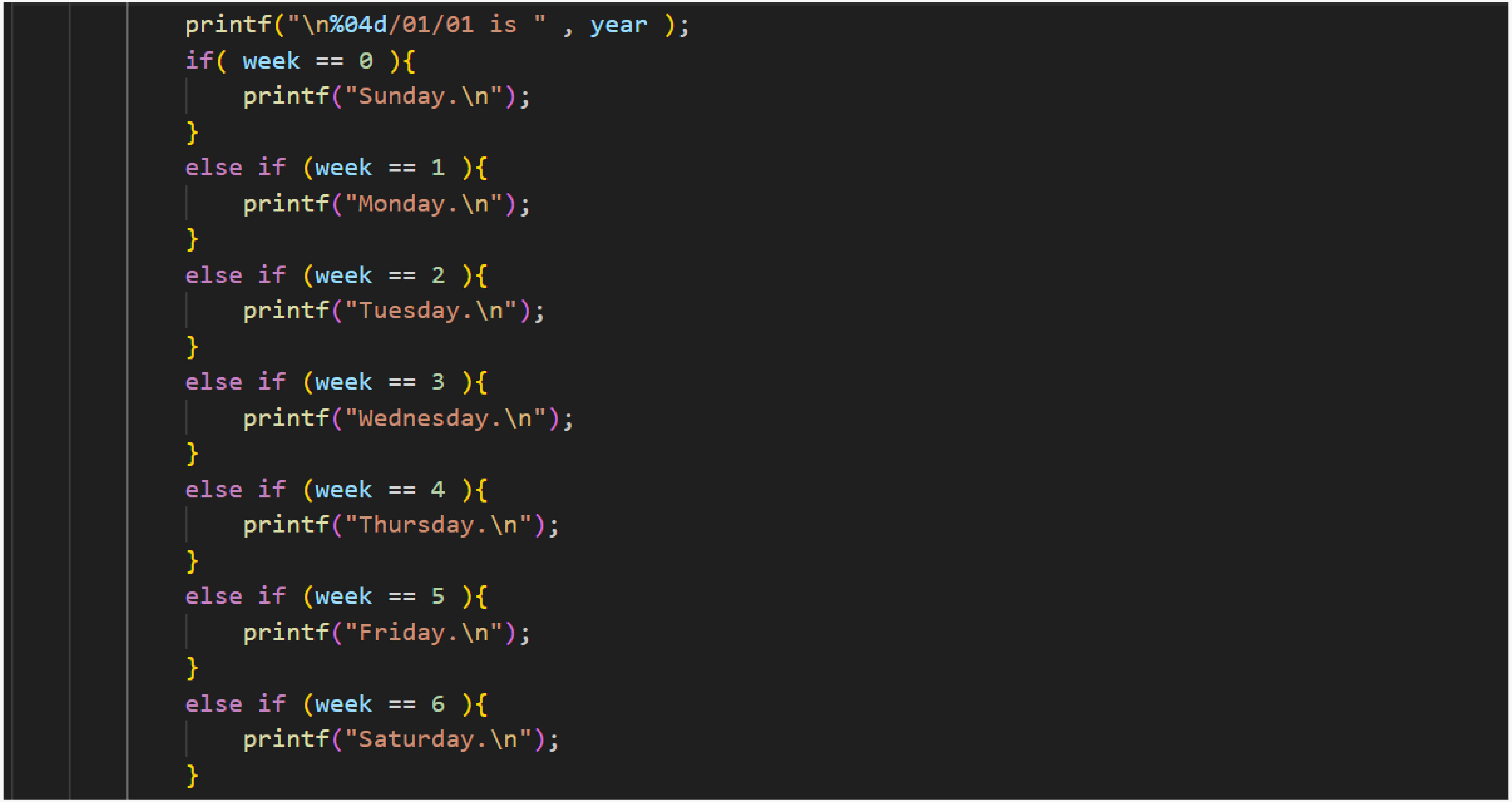
```c
while(1){
    printf("Please input a date (yyyy/mm/dd): ");
    int year , month , day ;
    scanf("%d/%d/%d", &year , &month , &day);
    if(year == month && month == day && day == 0){
        printf("Go to next question.\n\n");
        break;
    }
```



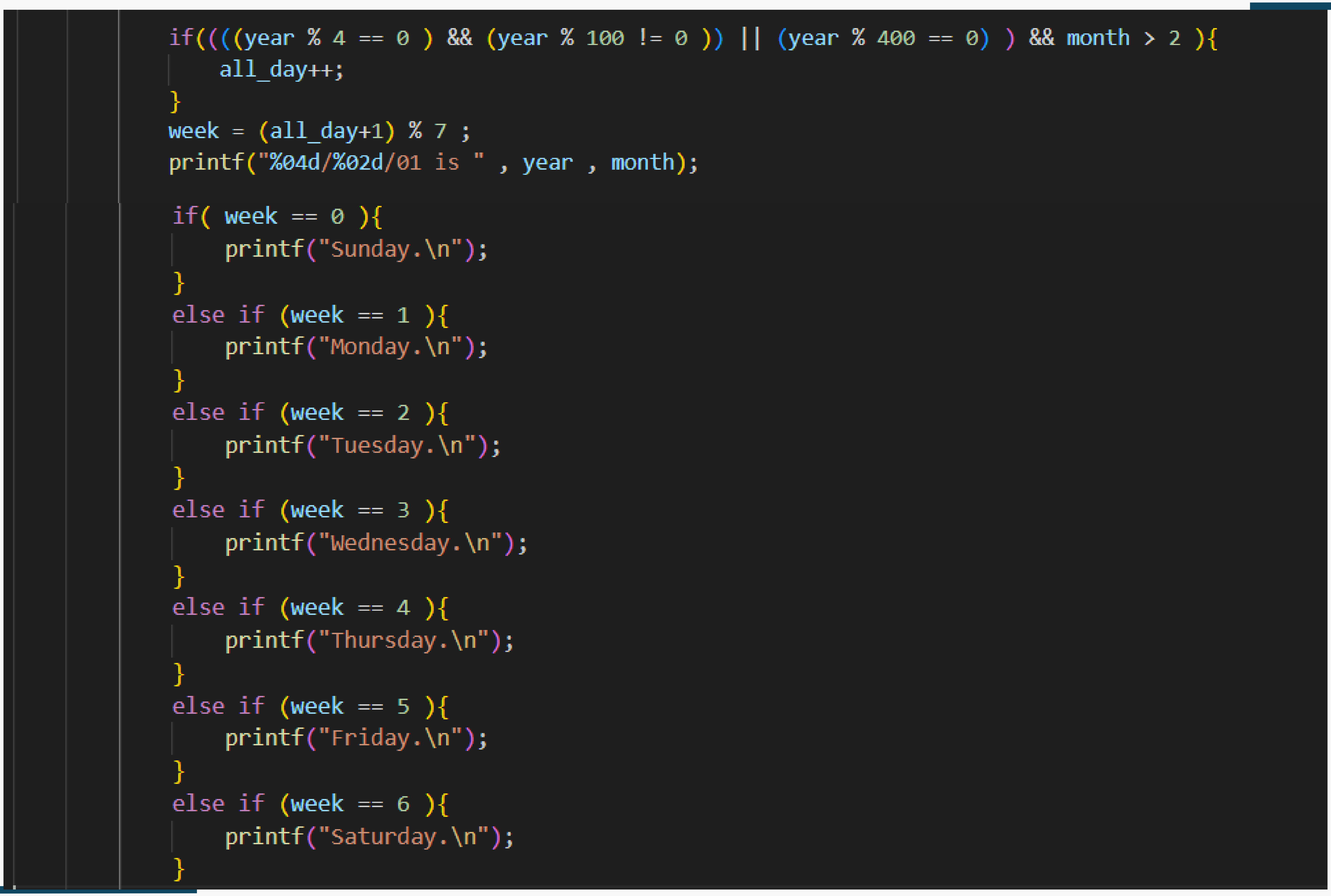
```
// check if the date is legal.
if (month < 1 || month > 12 ){
    printf("The month is wrong, input again!\n\n");
    continue;
}
if (day < 1 || day > 31 ){
    printf("The date is wrong, input again!\n\n");
    continue;
}
if((month == 4 || month == 6 ||month == 9 ||month == 11 ) && day > 30 ){
    printf("The date is wrong, input again!\n\n");
    continue;
}
if(((year % 4 == 0 ) && (year % 100 != 0 )) || (year % 400 == 0) ){
    if(month == 2 && day > 29 ){
        printf("The date is wrong, input again!\n\n");
        continue;
    }
}
else{
    if(month == 2 && day > 28 ){
        printf("The date is wrong, input again!\n\n");
        continue;
    }
}
```



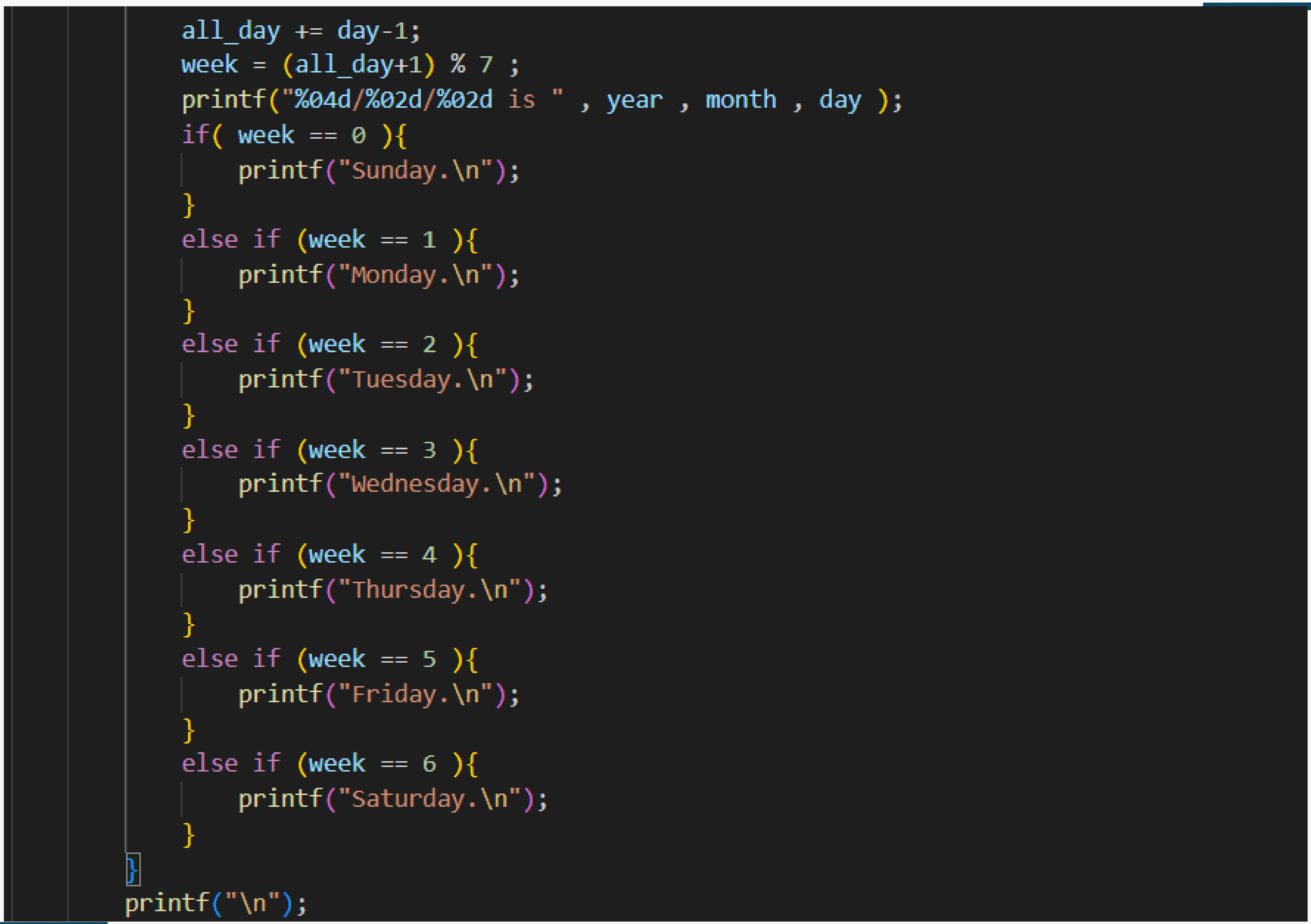
```
int week ;  //表示當天是星期幾
int all_day = 0 ; // 到前一年底的總天數是多少
int i ;
if(year == 1 ){
    week = 1 ; // 0001/01/01 is Monday.
}
else{
    // careful >> 'i' is start from 1
    for(i = 1 ; i < year ; i++ ){
        if(((year % 4 == 0 ) && (year % 100 != 0 )) || (year % 400 == 0) ){
            all_day += 366;
        }
        else{
            all_day += 365;
        }
    }
    week = (all_day+1) % 7 ;
```



```c
printf("\n%04d/01/01 is " , year );
if( week == 0 ){
    printf("Sunday.\n");
}
else if (week == 1 ){
    printf("Monday.\n");
}
else if (week == 2 ){
    printf("Tuesday.\n");
}
else if (week == 3 ){
    printf("Wednesday.\n");
}
else if (week == 4 ){
    printf("Thursday.\n");
}
else if (week == 5 ){
    printf("Friday.\n");
}
else if (week == 6 ){
    printf("Saturday.\n");
}
```



```c
if((((year % 4 == 0 ) && (year % 100 != 0 )) || (year % 400 == 0) ) && month > 2 ){
    all_day++;
}
week = (all_day+1) % 7 ;
printf("%04d/%02d/01 is " , year , month);

if( week == 0 ){
    printf("Sunday.\n");
}
else if (week == 1 ){
    printf("Monday.\n");
}
else if (week == 2 ){
    printf("Tuesday.\n");
}
else if (week == 3 ){
    printf("Wednesday.\n");
}
else if (week == 4 ){
    printf("Thursday.\n");
}
else if (week == 5 ){
    printf("Friday.\n");
}
else if (week == 6 ){
    printf("Saturday.\n");
}
```

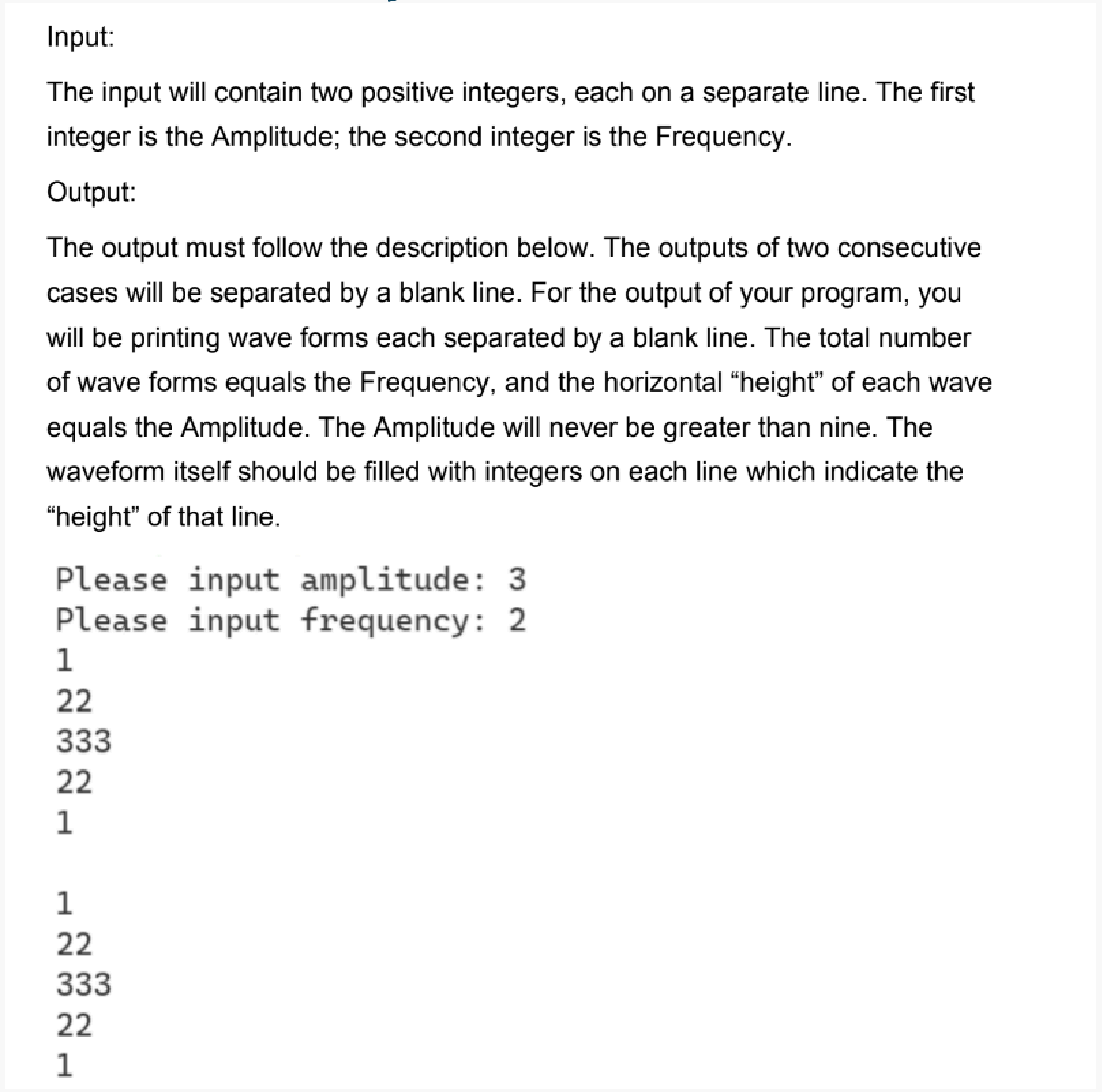


```c
        all_day += day-1;
        week = (all_day+1) % 7 ;
        printf("%04d/%02d/%02d is " , year , month , day );
        if( week == 0 ){
            printf("Sunday.\n");
        }
        else if (week == 1 ){
            printf("Monday.\n");
        }
        else if (week == 2 ){
            printf("Tuesday.\n");
        }
        else if (week == 3 ){
            printf("Wednesday.\n");
        }
        else if (week == 4 ){
            printf("Thursday.\n");
        }
        else if (week == 5 ){
            printf("Friday.\n");
        }
        else if (week == 6 ){
            printf("Saturday.\n");
        }
    }
    printf("\n");
```

5.Print Special Pattern

Print Special Pattern

110-5



Input:

The input will contain two positive integers, each on a separate line. The first integer is the Amplitude; the second integer is the Frequency.

Output:

The output must follow the description below. The outputs of two consecutive cases will be separated by a blank line. For the output of your program, you will be printing wave forms each separated by a blank line. The total number of wave forms equals the Frequency, and the horizontal "height" of each wave equals the Amplitude. The Amplitude will never be greater than nine. The waveform itself should be filled with integers on each line which indicate the "height" of that line.

```
Please input amplitude: 3
Please input frequency: 2
1
22
333
22
1

1
22
333
22
1
```

Print Special Pattern

112-exp-b

The hourglass is printed only when N is a positive odd number.

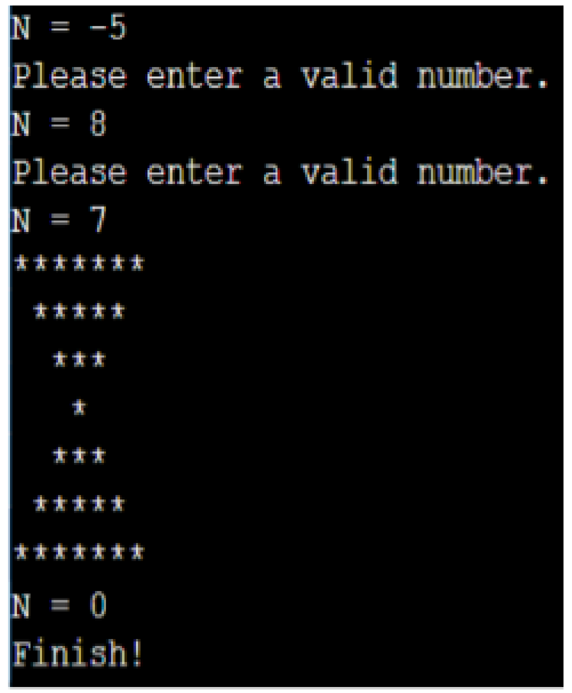
On the other hand, when N is an even number or a negative number, "Please enter a valid number" is printed.

Spaces only need to be printed before *.

Hint

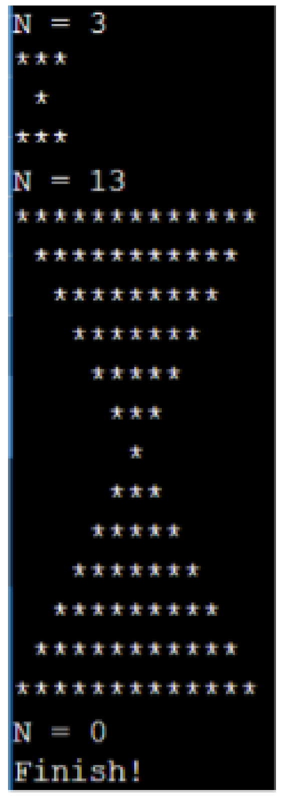
Please use **while(scanf(...) != EOF)**

example1:



```
N = -5
Please enter a valid number.
N = 8
Please enter a valid number.
N = 7
*******
 *****
  ***
   *
  ***
 *****
*******
N = 0
Finish!
```

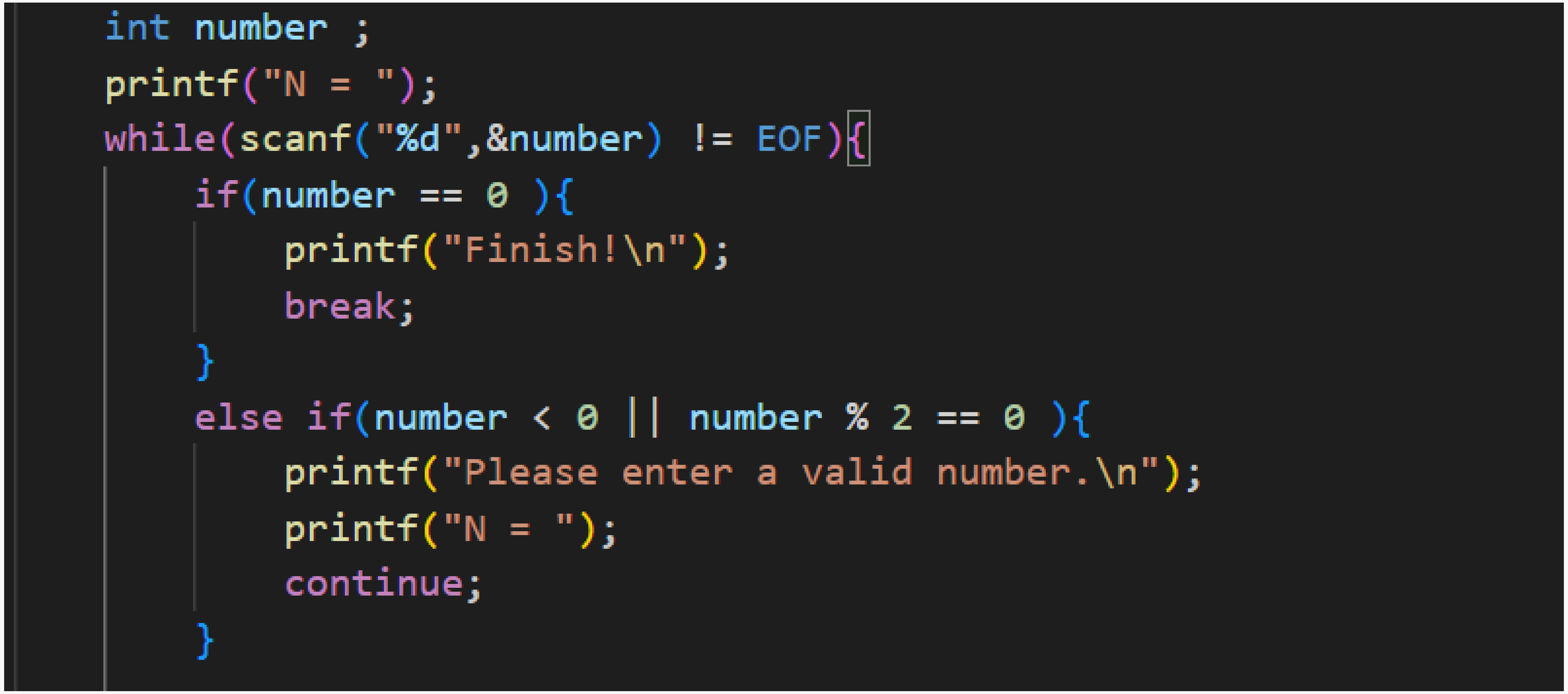
example2:



```
N = 3
***
 *
***
N = 13
*************
 ***********
  *********
   *******
    *****
     ***
      *
     ***
    *****
   *******
  *********
 ***********
*************
N = 0
Finish!
```

Print Special Pattern

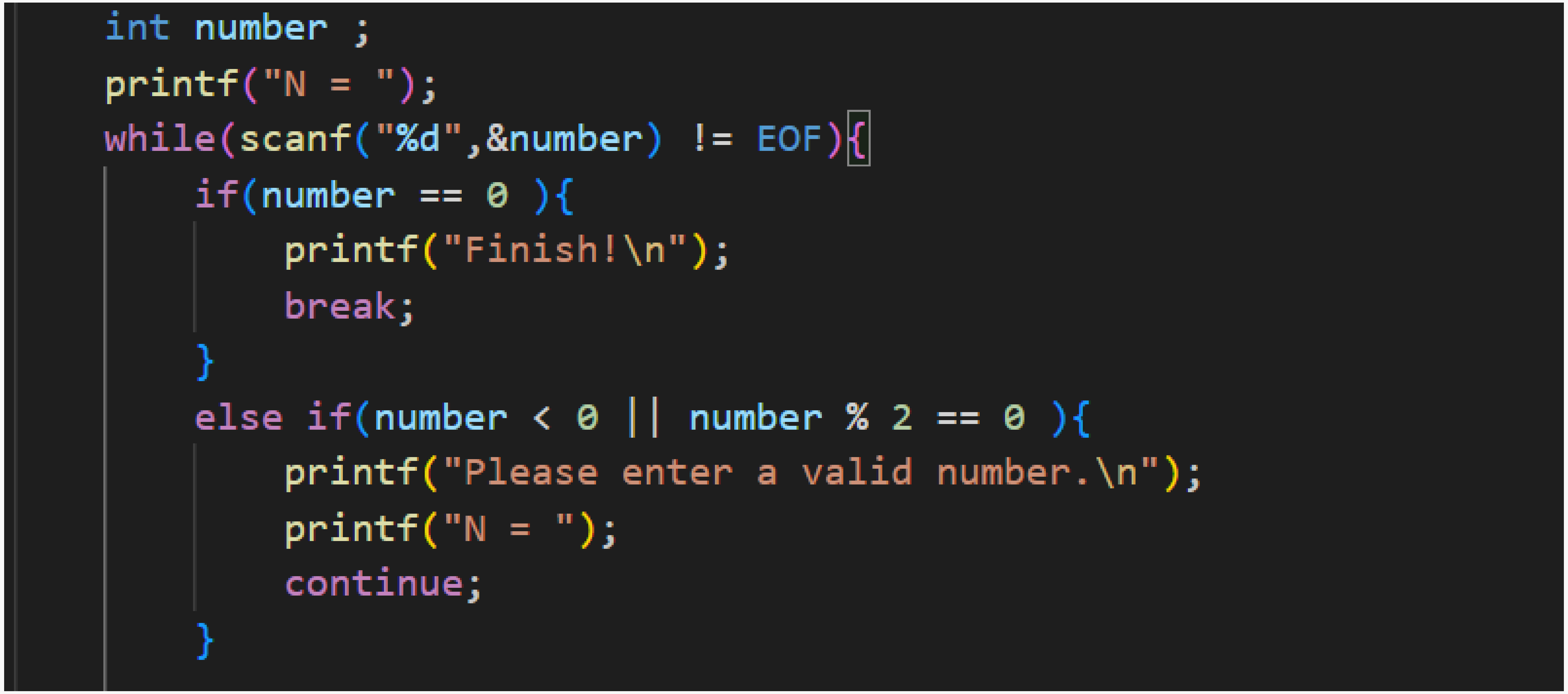
Method: Observing the output , and think....



```
    int number ;
    printf("N = ");
    while(scanf("%d",&number) != EOF){
        if(number == 0 ){
            printf("Finish!\n");
            break;
        }
        else if(number < 0 || number % 2 == 0 ){
            printf("Please enter a valid number.\n");
            printf("N = ");
            continue;
        }
```

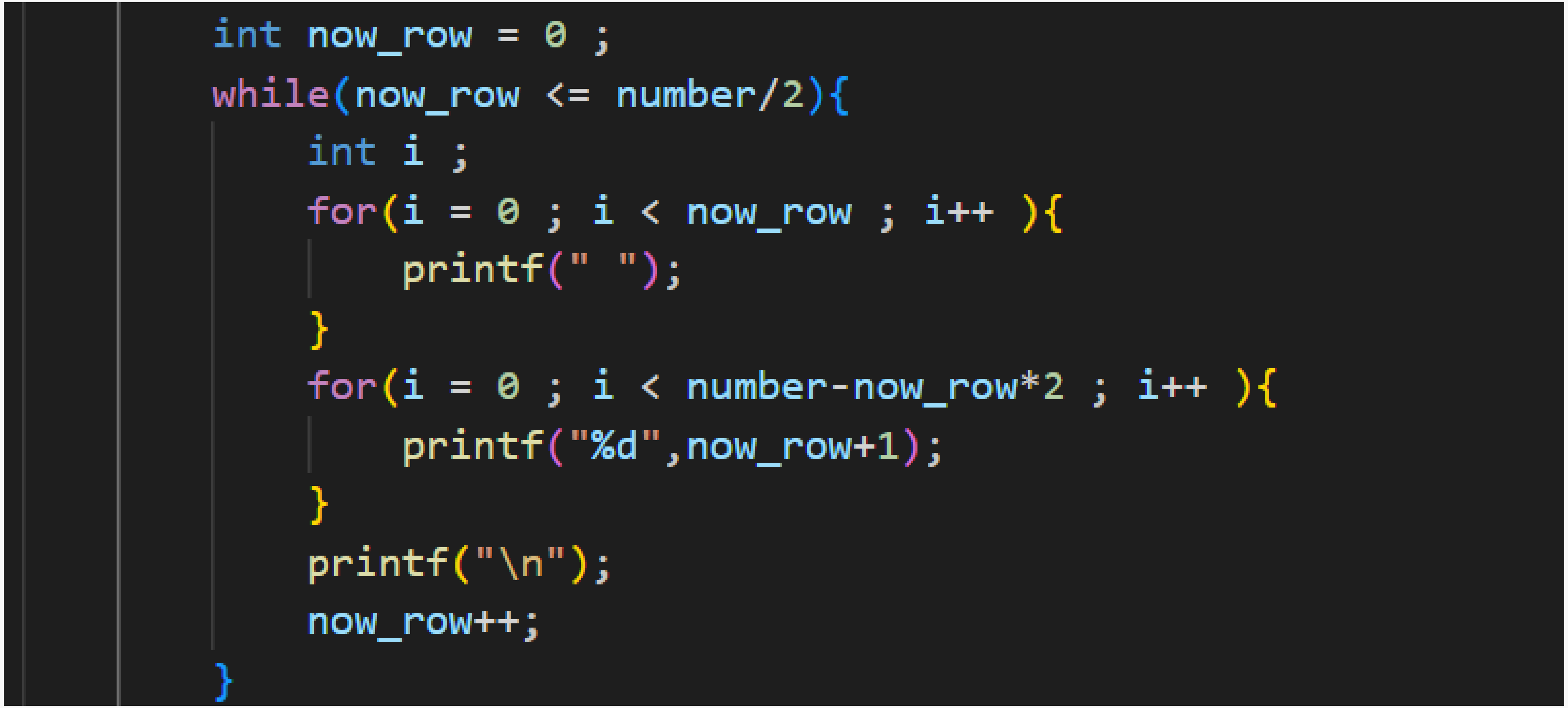
Print Special Pattern

Method: Observing the output , and think....



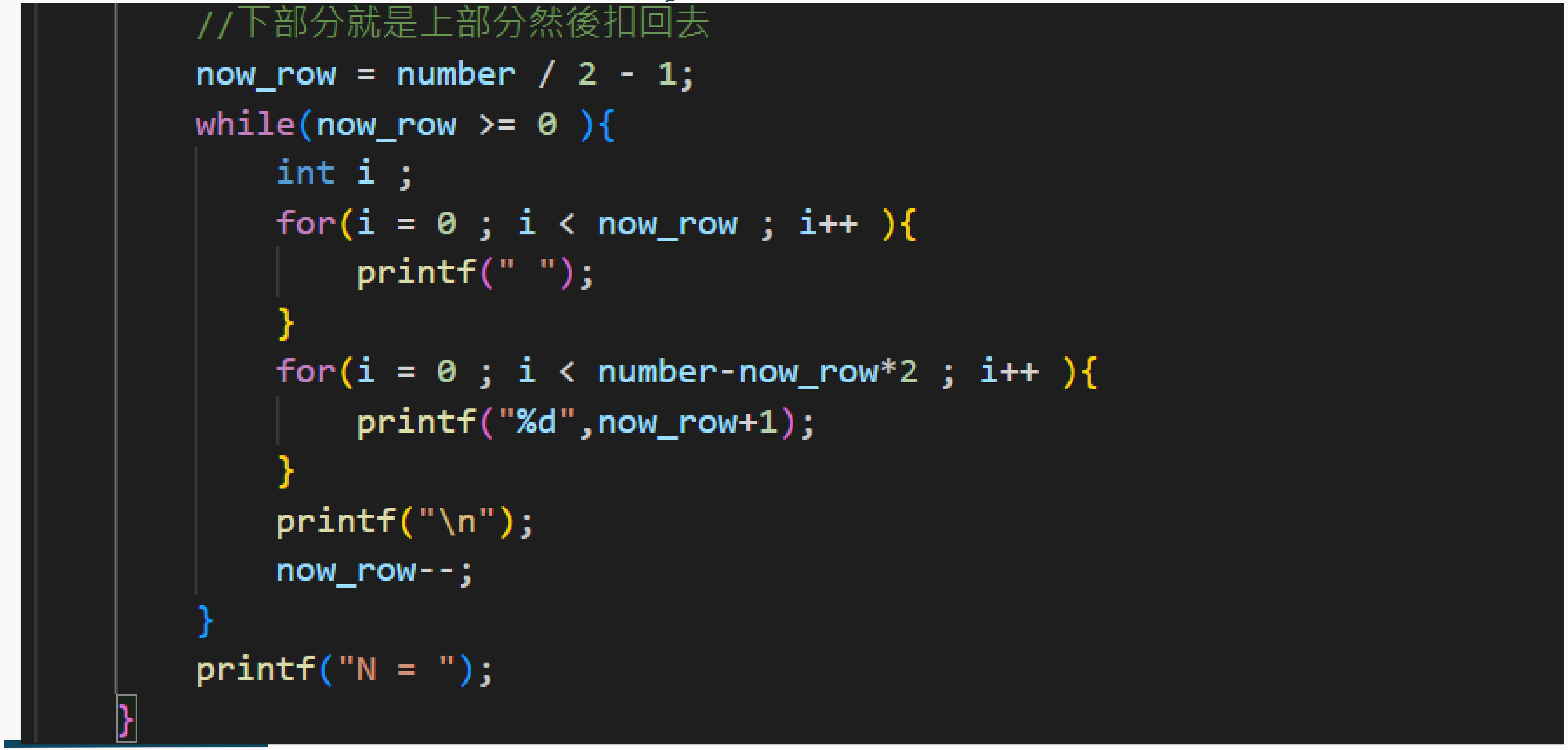
```
    int number ;
    printf("N = ");
    while(scanf("%d",&number) != EOF){
        if(number == 0 ){
            printf("Finish!\n");
            break;
        }
        else if(number < 0 || number % 2 == 0 ){
            printf("Please enter a valid number.\n");
            printf("N = ");
            continue;
        }
```

Print Special Pattern



```
int now_row = 0 ;
while(now_row <= number/2){
    int i ;
    for(i = 0 ; i < now_row ; i++ ){
        printf(" ");
    }
    for(i = 0 ; i < number-now_row*2 ; i++ ){
        printf("%d",now_row+1);
    }
    printf("\n");
    now_row++;
}
```

Print Special Pattern



```
        //下部分就是上部分然後扣回去
        now_row = number / 2 - 1;
        while(now_row >= 0 ){
            int i ;
            for(i = 0 ; i < now_row ; i++ ){
                printf(" ");
            }
            for(i = 0 ; i < number-now_row*2 ; i++ ){
                printf("%d",now_row+1);
            }
            printf("\n");
            now_row--;
        }
        printf("N = ");
    }
```

Next class topic preview....

1. Array
2. Function
3. Recursion
4. Pointer

•••••

Thank you