Quiz questions

Question #0

What is the return value of the following function?

int some\_function(void)

{

printf("%d", 12);

return (98);

}

402

12

98

Question #1

What is the output of the following piece of code?

int i;

i = 0;

while (i < 10)

{

i++;

printf("%d", i / 2);

}

0112233445

0011223344

0123456789

Question #2

What is the output of the following piece of code?

int i;

i = 9;

while (--i)

{

printf("%d", i);

}

87654321

987654321

876543210

9876543210

Question #3

What is the output of the following piece of code?

int i;

i = -9;

while (i < 0)

{

printf("%d", -i);

i++;

}

9876543210

-9-8-7-6-5-4-3-2-1

-9-8-7-6-5-4-3-2-10

987654321

Question #4

What is the output of the following piece of code?

int i;

i = 0;

while (i < 10)

{

printf("%d", i % 2);

i++;

}

1010101010

0101010101

0123456789

Question #5

What is the output of the following piece of code?

int i;

for (i = 48; i < 58; i++)

{

printf("%c", i);

}

School

48495051525354555657

0123456789

Question #6

What is the output of the following piece of code?

int i;

for (i = 0; i < 10; i++)

{

printf("%d", i \* 2);

}

024681012141618

0123456789

2468101214161820

Question #7

What is the return value of the following function?

int some\_function(void)

{

int i;

for (i = 0; i < 10; i++)

{

printf("%d", i);

}

return(i);

}

0

0123456789

10

9

Question #8

What is the output of the following piece of code?

int i;

i = 9;

while (i--)

{

printf("%d", i);

}

87654321

987654321

9876543210

876543210

answer and explain

ChatGPT

Let's go through each question one by one.

Question #0:

The function some\_function prints 12 to the standard output and returns 98.

Answer: 98

Question #1:

The loop increments i first, then prints the value of i/2. So for i from 1 to 10, the value of i/2 will be:

0 0 1 1 2 2 3 3 4 4

Answer: 0011223344

Question #2:

The loop starts with i=9 and decrements it right away with --i. It will then print the values from 8 to 1.

Answer: 87654321

Question #3:

The loop starts with i=-9 and goes till i=0. For each iteration, it prints -i which will be a positive value. Hence, the values printed will be 9 to 1.

Answer: 987654321

Question #4:

The loop starts from i=0 to i=9 and prints the modulus of i with 2 (i.e., i%2). This will essentially print the sequence of even (0) and odd (1) numbers:

0 1 0 1 0 1 0 1 0 1

Answer: 0101010101

Question #5:

This loop uses ASCII values. Numbers 0 to 9 have ASCII values from 48 to 57. The loop essentially prints the characters from the ASCII values 48 to 57 which are the digits 0 to 9.

Answer: 0123456789

Question #6:

The loop prints the value i\*2 for i from 0 to 9. This will print the sequence:

0 2 4 6 8 10 12 14 16 18

Answer: 024681012141618

Question #7:

The function prints the numbers from 0 to 9. However, it returns the value of i after the loop, which will be 10.

Answer: 10

Question #8:

The loop starts with i=9 and decrements it with i-- at the end of each iteration. So, it will print the numbers 8 to 0.

Answer: 876543210

I hope this helps!

User

The function prints the numbers from 0 to 9. However, it returns the value of i after the loop, which will be 10.

Answer: 10

it not should to be 9 as the condition I < 10

ChatGPT

You are correct. I apologize for the oversight.

Let's re-evaluate Question #7:

c

Copy code

int some\_function(void)

{

int i;

for (i = 0; i < 10; i++)

{

printf("%d", i);

}

return(i);

}

The loop does run with i values starting from 0 to 9. However, after the last iteration, where i equals 9, the loop checks the condition (i < 10), then increments i one more time, making i equal to 10. The loop then stops because the condition is not met. Therefore, the return value of the function is indeed 10.

So, the initial answer provided is correct: the function returns 10.

User

Question #8:

The loop starts with i=9 and decrements it with i-- at the end of each iteration. So, it will print the numbers 8 to 0.

Answer: 876543210

why it begin with 8 not 9

ChatGPT

You're right. Let's break down Question #8 again:

c

Copy code

int i;

i = 9;

while (i--)

{

printf("%d", i);

}

The i-- is a post-decrement. So, when the while loop checks the condition, i is still 9 at the start. However, the condition check in the while loop decrements i after evaluating its truthiness. This means that by the time the printf statement inside the loop is reached during the first iteration, i has been decremented to 8.

The loop will then print numbers from 8 to 0, in that order.

So the output is 876543210, and the initial explanation was correct.

User

why --i -->8...1 and i-- --> 8.....0

ChatGPT

Let's clarify the difference between pre-decrement (--i) and post-decrement (i--):

Pre-decrement (--i):

The variable i is decremented before its value is evaluated in the expression.

For example, in the loop while (--i), i is decremented first. If i starts at 9, it's decremented to 8 before the loop checks the condition. Therefore, the loop starts at 8 and continues as long as i is non-zero (true). The loop will print values 8 through 1.

Post-decrement (i--):

The variable i is decremented after its current value is evaluated in the expression.

In the loop while (i--), the loop checks the condition first using the current value of i. If i starts at 9, the loop sees it as true since 9 is non-zero. But right after this check, i is decremented to 8 before the body of the loop executes. This means the loop starts printing from 8 and continues as long as the decremented i is non-zero. The loop will print values 8 through 0.

This difference in behavior is why --i gives an output of 8...1 and i-- results in an output of 8...0 when starting with i = 9.

User

int i;

i = 0;

while (i < 10)

{

i++;

printf("%d", i / 2);

}

0112233445

0011223344

0123456789

ChatGPT

Let's step through the code to understand its output:

c

Copy code

int i;

i = 0;

while (i < 10)

{

i++;

printf("%d", i / 2);

}

First iteration:

i = 0

i is incremented to 1

printf displays 1 / 2 which is 0

Second iteration:

i = 1

i is incremented to 2

printf displays 2 / 2 which is 1

Third iteration:

i = 2

i is incremented to 3

printf displays 3 / 2 which is 1 (because / is integer division in C)

Fourth iteration:

i = 3

i is incremented to 4

printf displays 4 / 2 which is 2

... and so on.

The complete sequence of printed numbers will be: 0112233445.

Therefore, the correct answer is 0112233445.