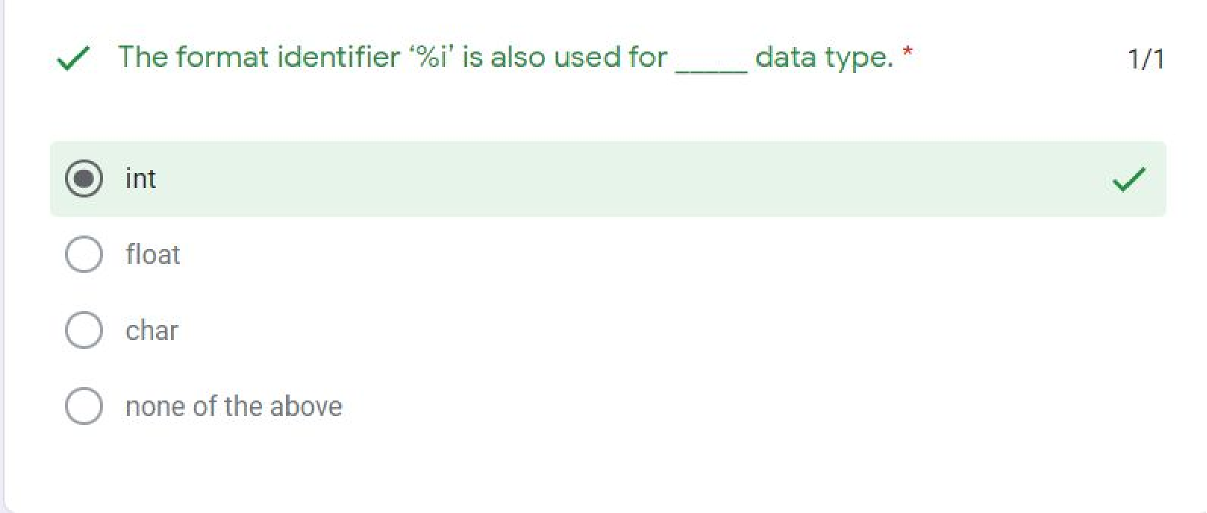


✓ The format identifier '%i' is also used for _____ data type. * 1/1

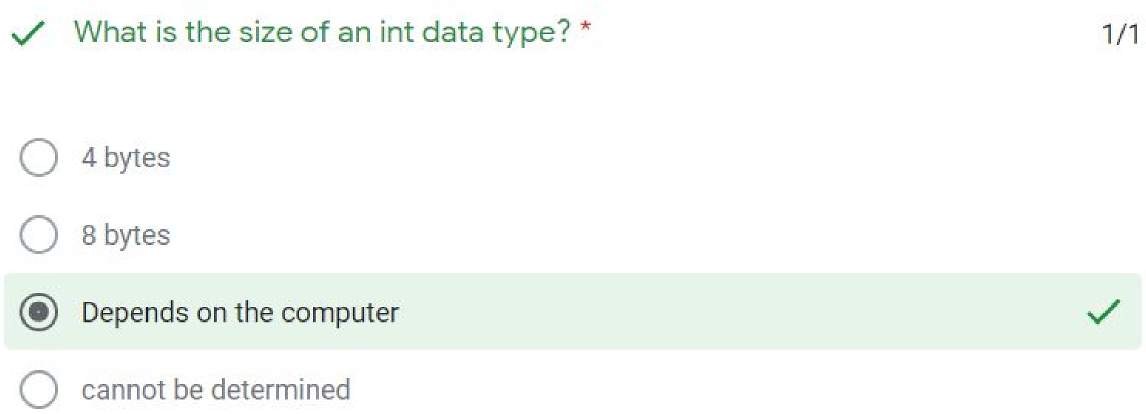
● int ✓

○ float

○ char

○ none of the above

Answer : Both %d and %i used for int datatype



✓ What is the size of an int data type? * 1/1

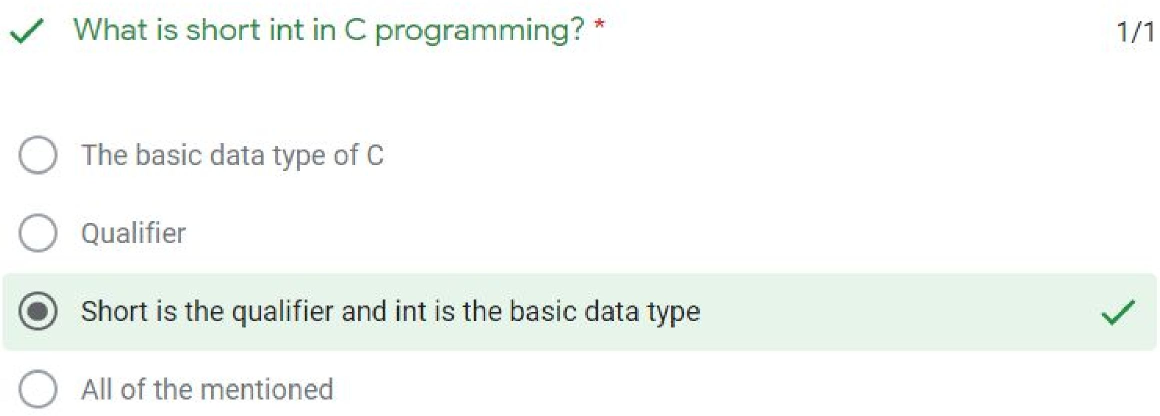
○ 4 bytes

○ 8 bytes

● Depends on the computer ✓

○ cannot be determined

Answer : The size of an `int` is really compiler dependent. Back in the day, when processors were 16 bit, an `int` was 2 bytes. Nowadays, it's most often 4 bytes on a 32-bit as well as 64-bit systems.



✓ What is short int in C programming? * 1/1

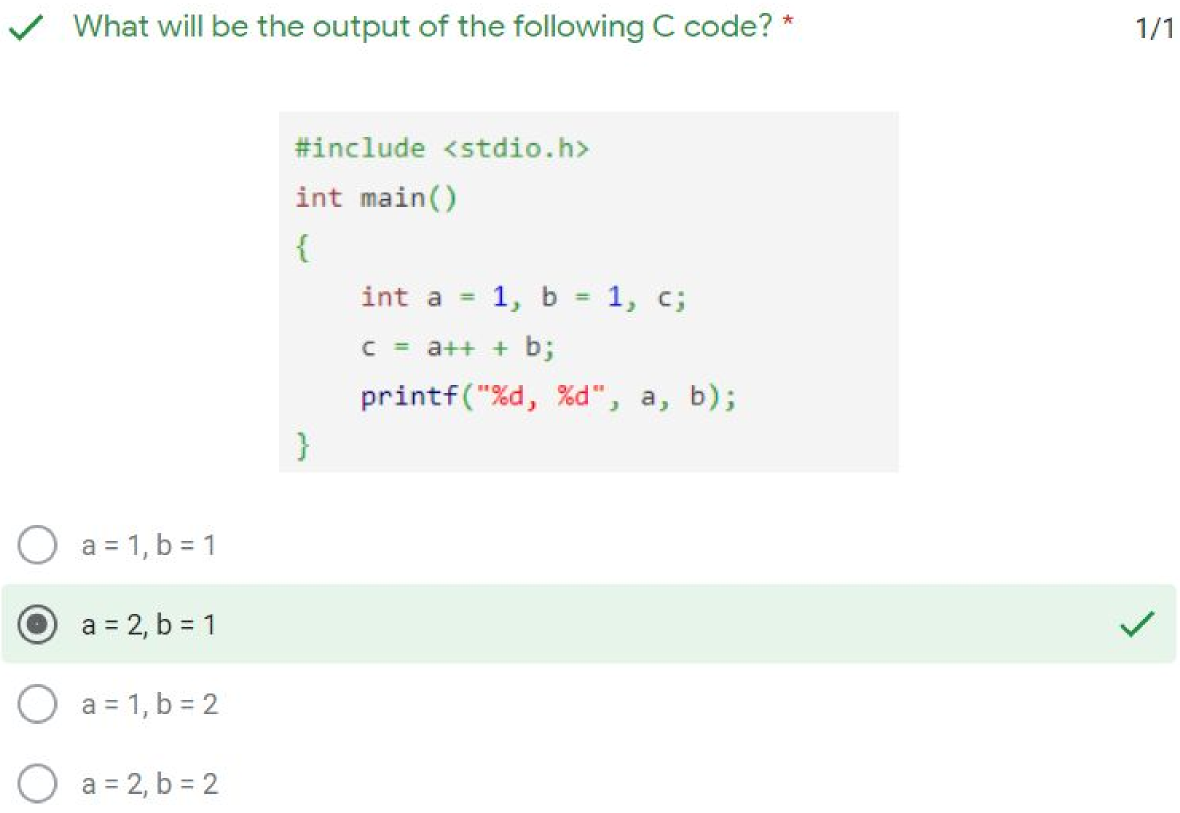
○ The basic data type of C

○ Qualifier

● Short is the qualifier and int is the basic data type ✓

○ All of the mentioned

Answer : int is the basic data type and short is the qualifier



✓ What will be the output of the following C code? * 1/1

```
#include <stdio.h>
int main()
{
    int a = 1, b = 1, c;
    c = a++ + b;
    printf("%d, %d", a, b);
}
```

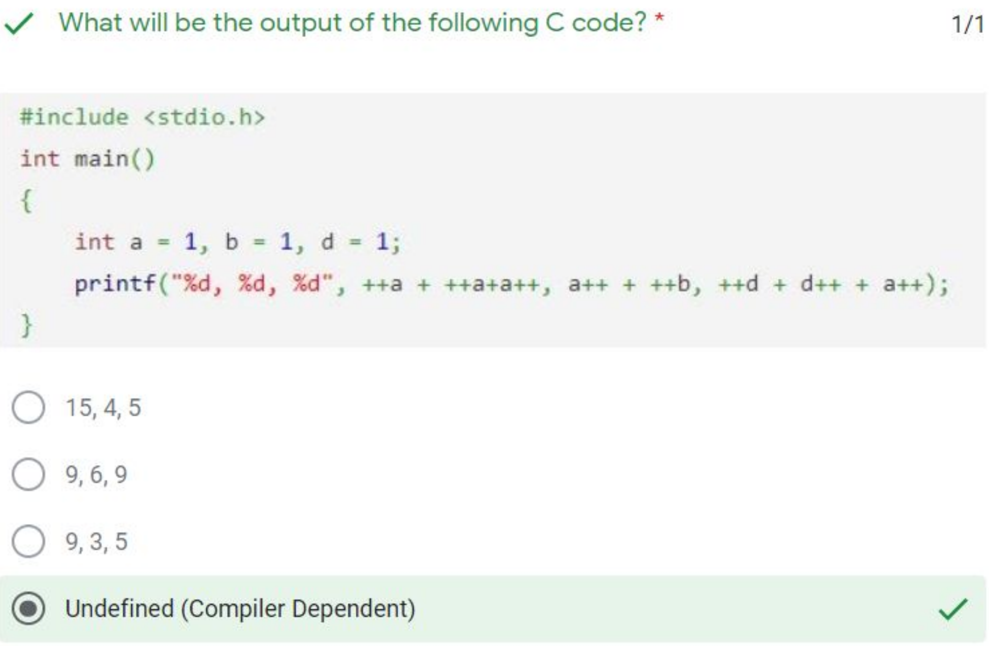
○ a = 1, b = 1

● a = 2, b = 1 ✓

○ a = 1, b = 2

○ a = 2, b = 2

Answer : => c = a++ + b
=> c = 1 + 1
=> c = 2
And then a is post incremented so a =2
Output is a=2 , b=1



✓ What will be the output of the following C code? * 1/1

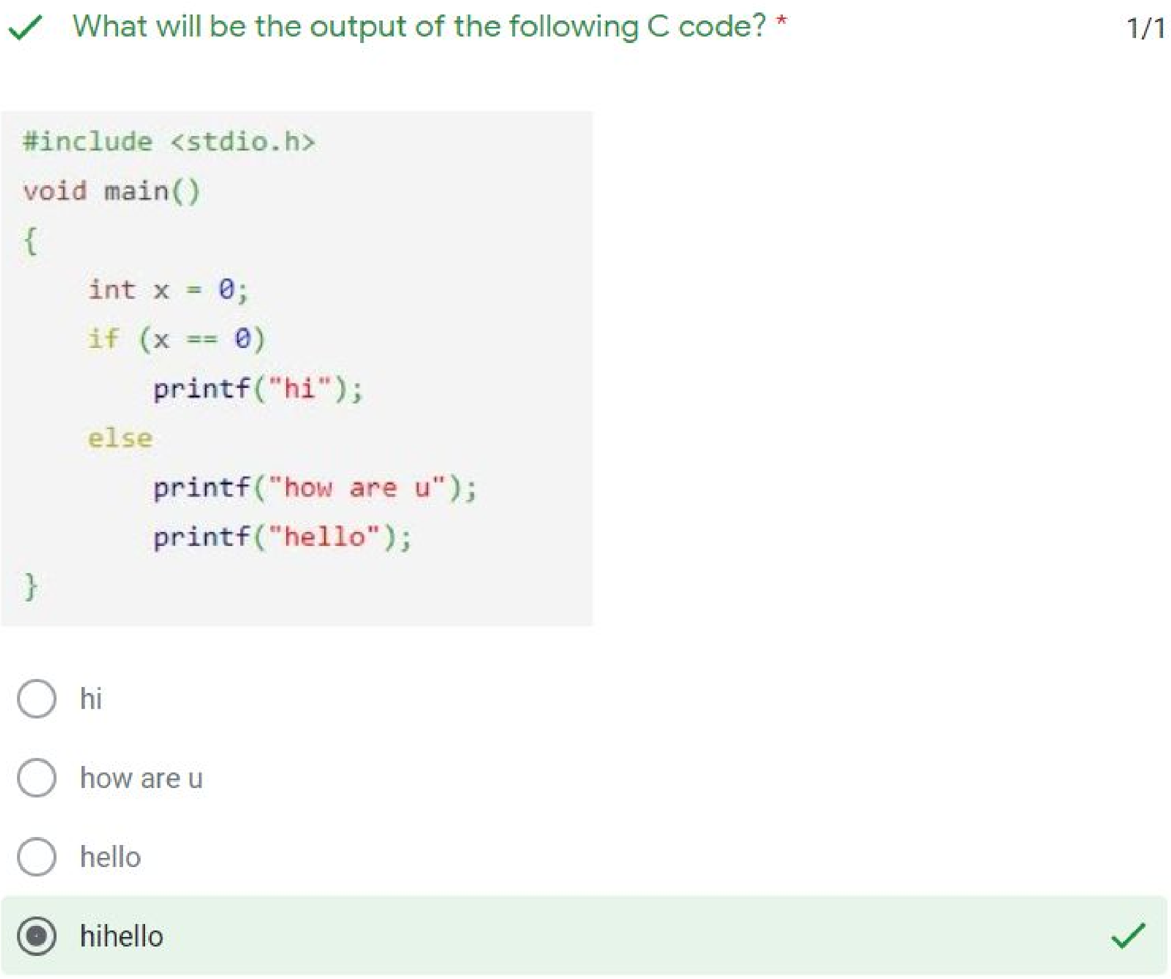
```
#include <stdio.h>
int main()
{
    int a = 1, b = 1, d = 1;
    printf("%d, %d, %d", ++a + ++a+a++, a++ + ++b, ++d + d++ + a++);
}
```

○ 15, 4, 5

○ 9, 6, 9

○ 9, 3, 5

● Undefined (Compiler Dependent) ✓



✓ What will be the output of the following C code? * 1/1

```
#include <stdio.h>
void main()
{
    int x = 0;
    if (x == 0)
        printf("hi");
    else
        printf("how are u");
        printf("hello");
}
```

○ hi

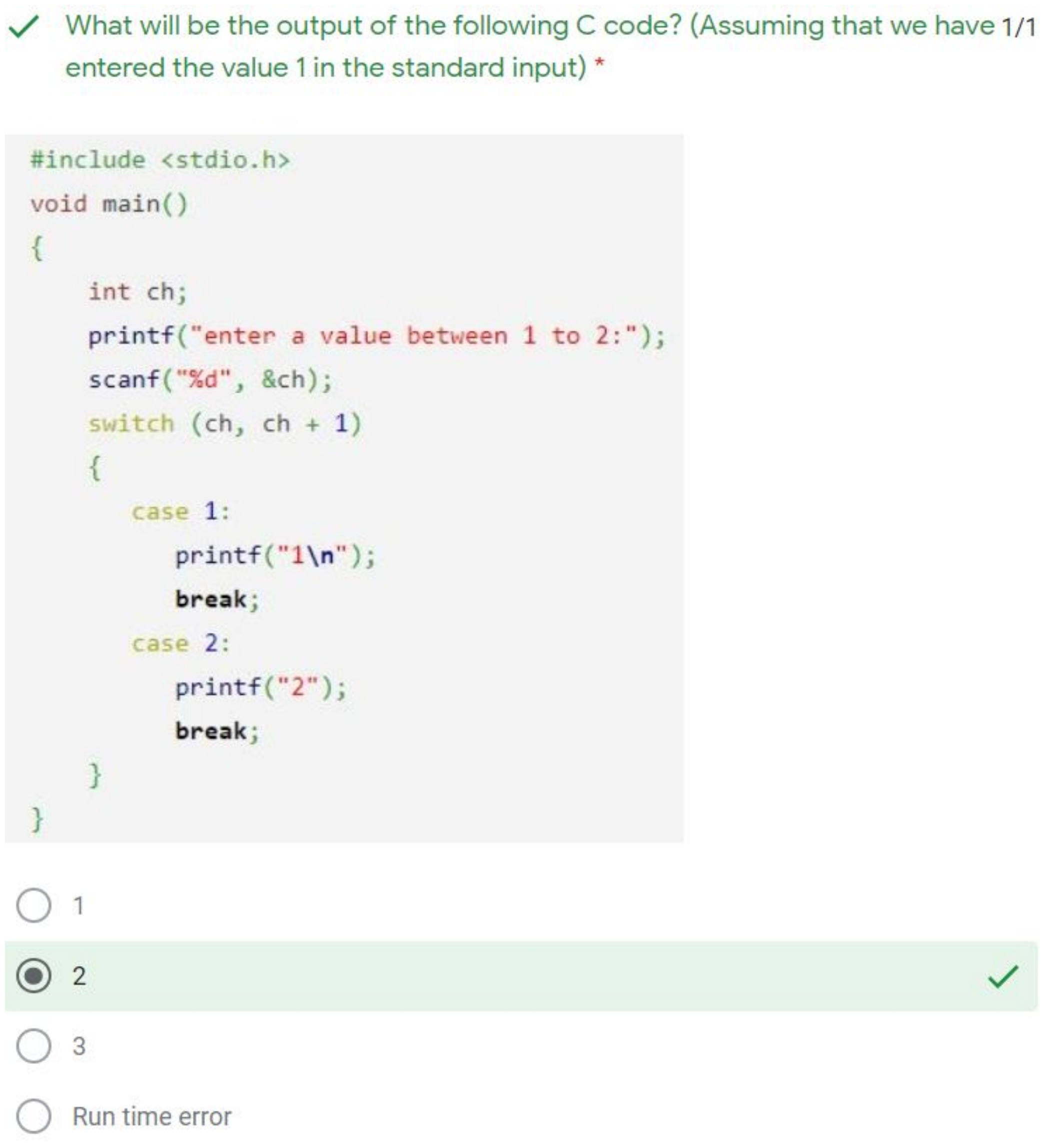
○ how are u

○ hello

● hihello ✓

Answer : hihello
Explanation : brackets are not included in the if and else part, hence one statement after if and else is executed based on the condition given.

printf(“hello”) - is not a part of the else statement.



✓ What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input) * 1/1

```
#include <stdio.h>
void main()
{
    int ch;
    printf("enter a value between 1 to 2:");
    scanf("%d", &ch);
    switch (ch, ch + 1)
    {
       case 1:
          printf("1\n");
          break;
       case 2:
          printf("2");
          break;
    }
}
```

○ 1

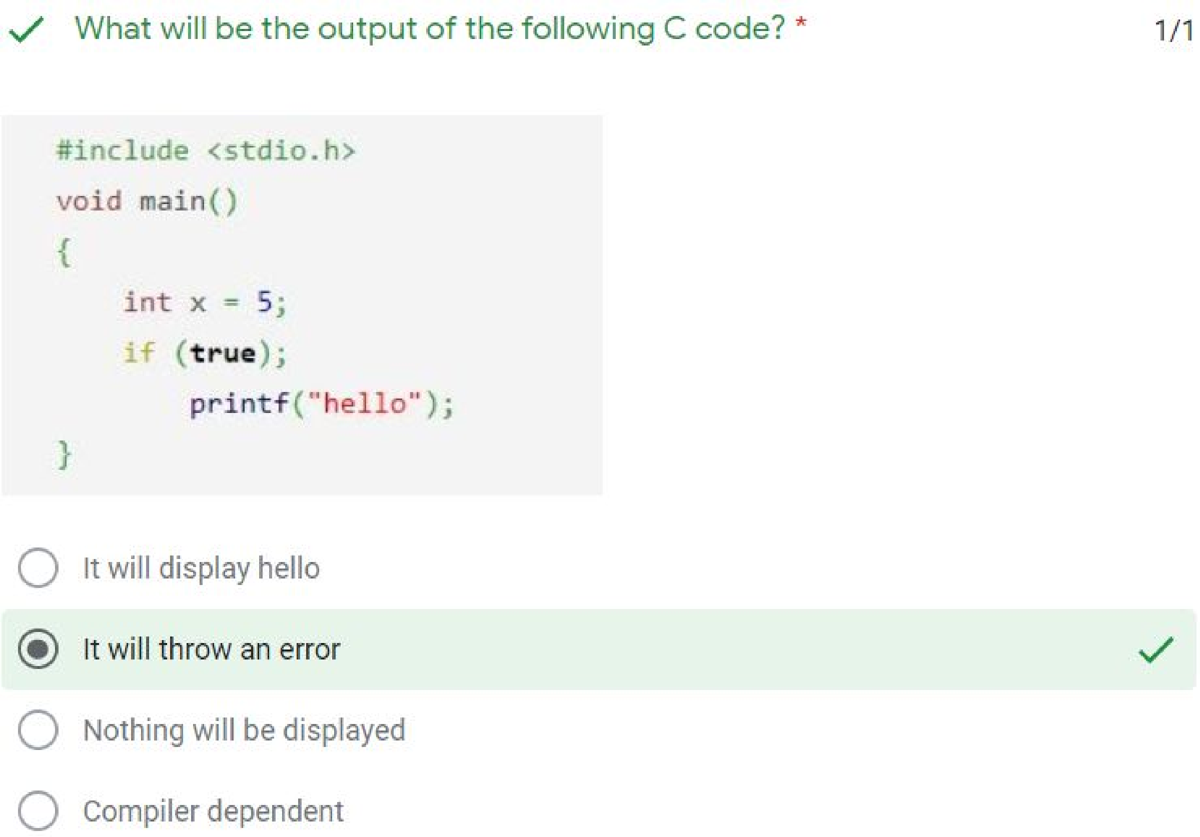
● 2 ✓

○ 3

○ Run time error

Answer:
The comma operator (represented by the token,) is a binary operator that evaluates its first operand and discards the result, it then evaluates the second operand and returns this value (and type). The comma operator has the lowest precedence of any C operator, and acts as a sequence point.

```
/* comma as an operator */
int i = (5, 10); /* 10 is assigned to i*/
int j = (f1(), f2()); /* f1() is called (evaluated) first followed by f2().
                                 The returned value of f2() is assigned to j */
```



✓ What will be the output of the following C code? * 1/1

```c
#include <stdio.h>
void main()
{
    int x = 5;
    if (true);
        printf("hello");
}
```

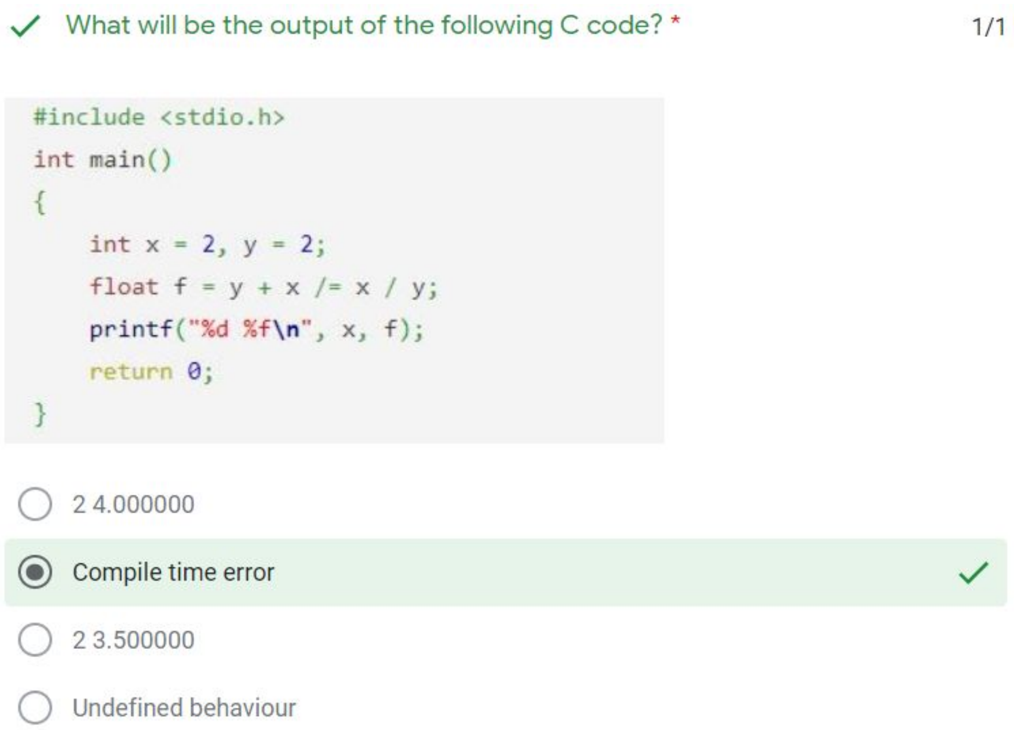
○ It will display hello

● It will throw an error ✓

○ Nothing will be displayed

○ Compiler dependent

Explanation: semicolon is included in the if statement.



✓ What will be the output of the following C code? * 1/1

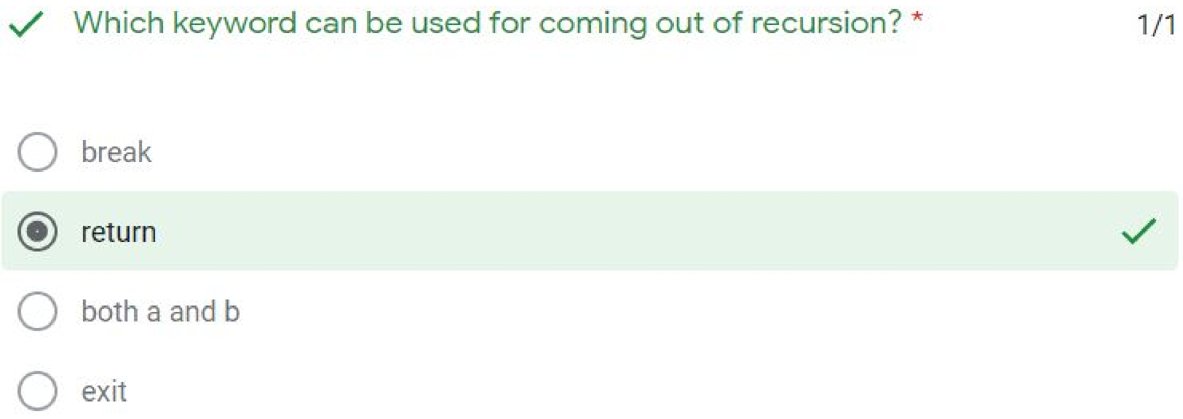
```c
#include <stdio.h>
int main()
{
    int x = 2, y = 2;
    float f = y + x /= x / y;
    printf("%d %f\n", x, f);
    return 0;
}
```

○ 2 4.000000

● Compile time error ✓

○ 2 3.500000

○ Undefined behaviour



✓ Which keyword can be used for coming out of recursion? * 1/1

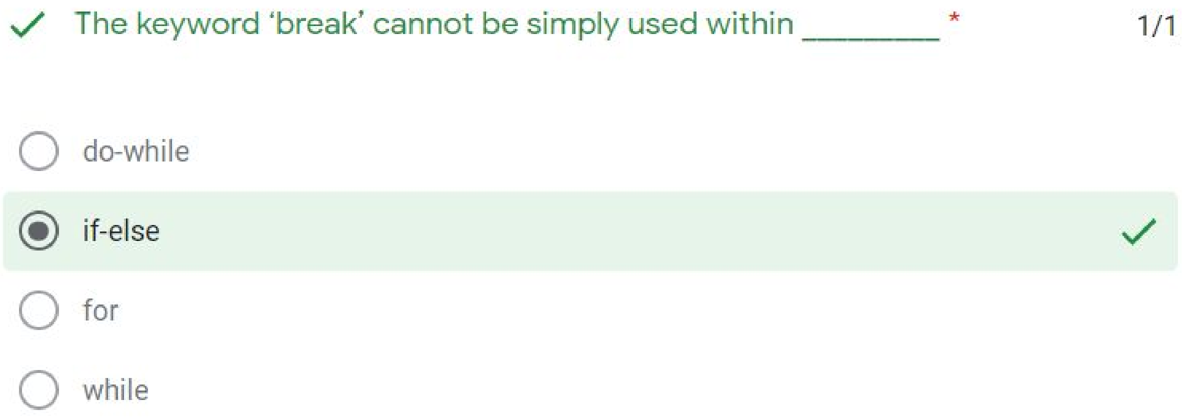
○ break

● return ✓

○ both a and b

○ exit

Answer : For coming out of a recursion, we should use return statement to get back to the line from where it is called.



✓ The keyword ‘break’ cannot be simply used within _________ * 1/1

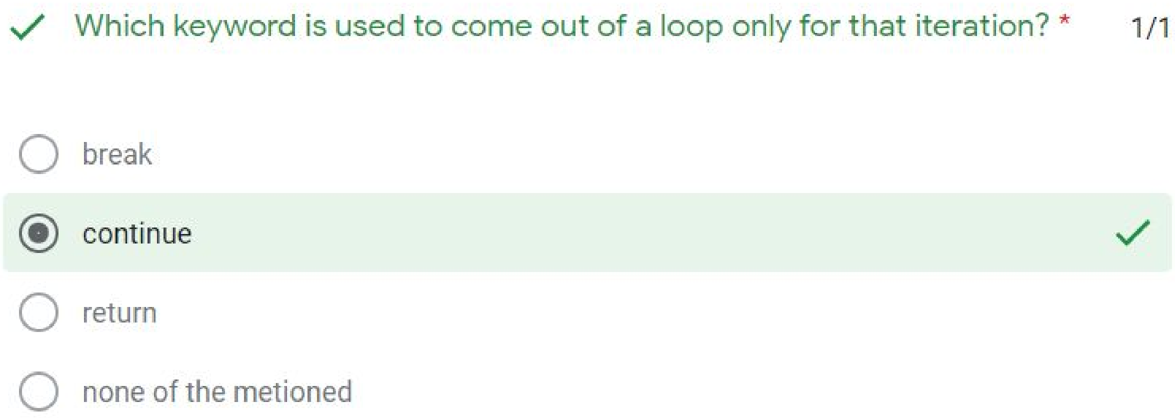
○ do-while

● if-else ✓

○ for

○ while

Answer: we can use break in a if statement and not in if-else statement.



✓ Which keyword is used to come out of a loop only for that iteration? * 1/1

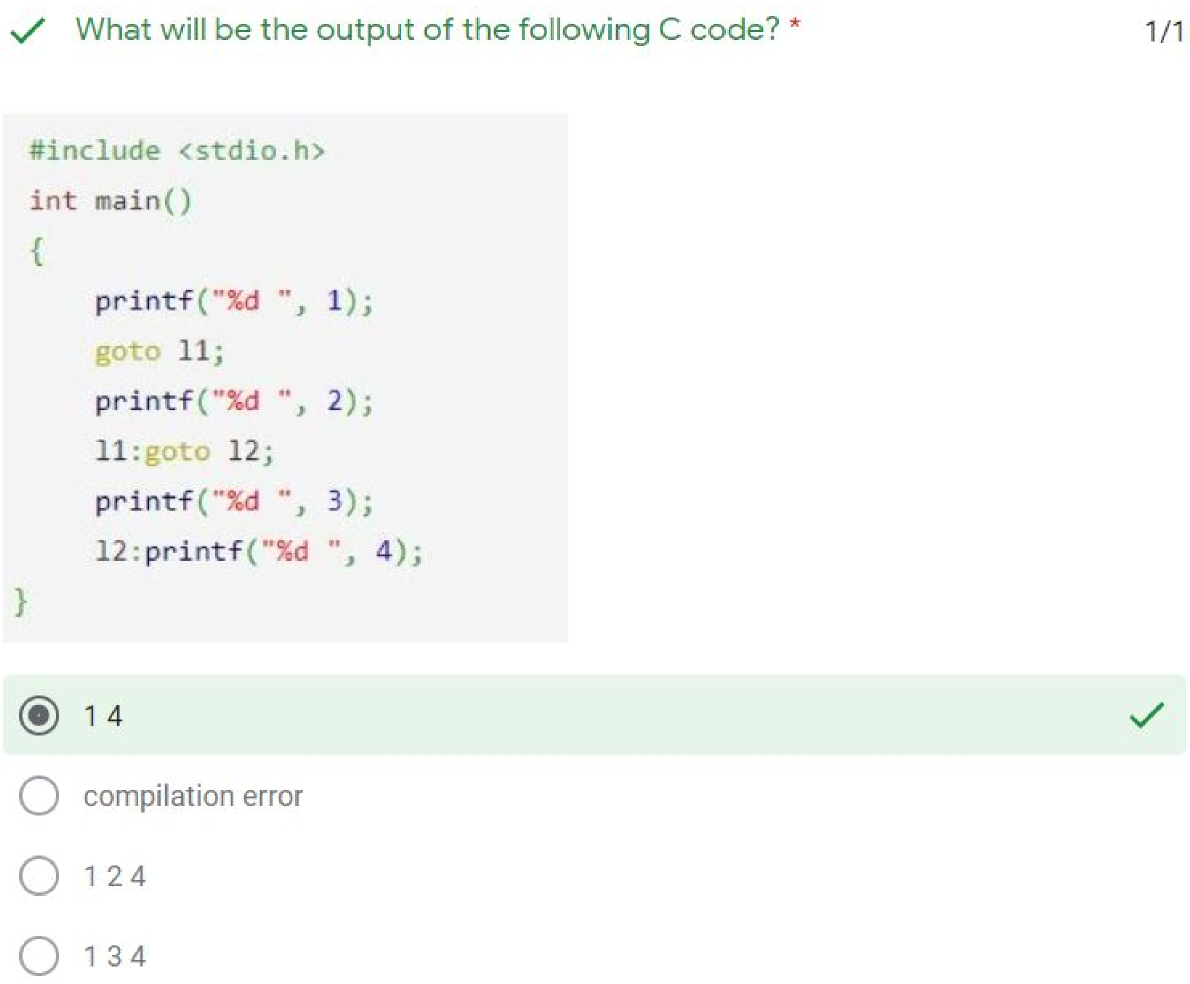
○ break

● continue ✓

○ return

○ none of the metioned

Answer: To come out for a specific iteration we use continue statement.



✓ What will be the output of the following C code? * 1/1

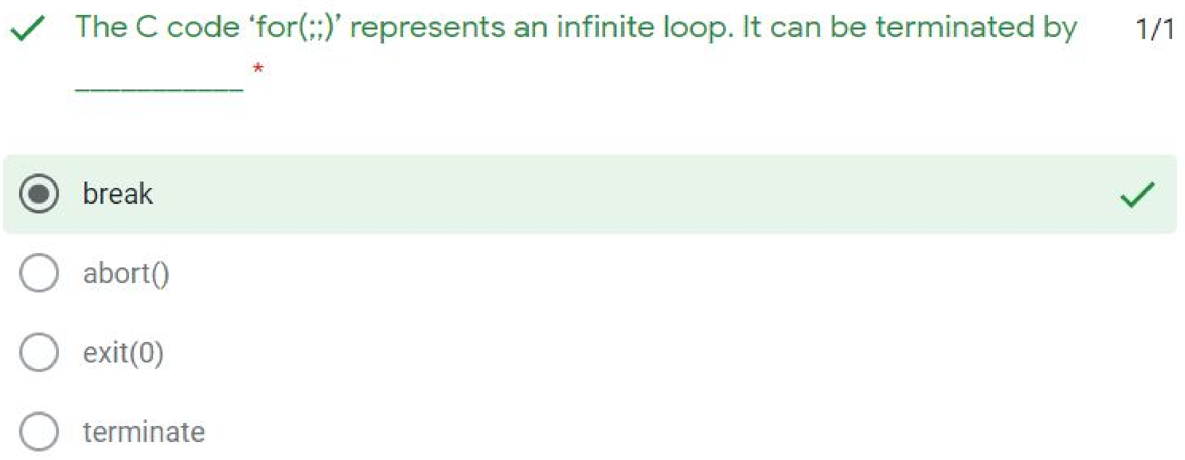
```
#include <stdio.h>
int main()
{
     printf("%d ", 1);
     goto l1;
     printf("%d ", 2);
     l1:goto l2;
     printf("%d ", 3);
     l2:printf("%d ", 4);
}
```

● 1 4 ✓

○ compilation error

○ 1 2 4

○ 1 3 4



✓ The C code 'for(;;)' represents an infinite loop. It can be terminated by ___________ * 1/1

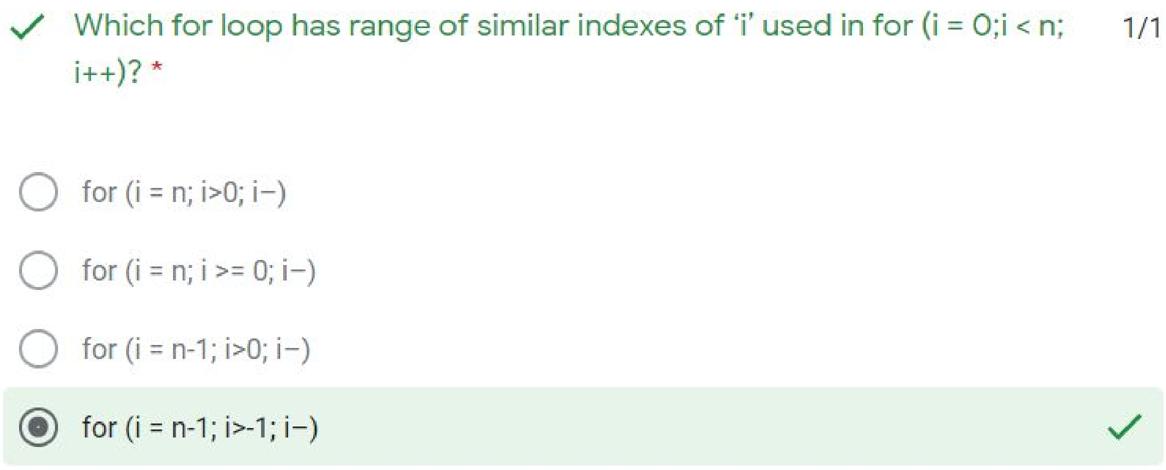
● break ✓

○ abort()

○ exit(0)

○ terminate

Answer: loop can be terminated by break statement.



✓ Which for loop has range of similar indexes of ‘i’ used in for (i = 0;i < n; i++)? * 1/1

○ for (i = n; i>0; i–)

○ for (i = n; i >= 0; i–)

○ for (i = n-1; i>0; i–)

● for (i = n-1; i>-1; i–) ✓