Tokens & Datatypes

MCQs (Part 01)

Question 1

1. Which of the following is not a token in C?

A) Keyword

B) Identifier

C) Comment ← Not a part of a program

D) Operator

Question 2

2. Which of the following keywords is not present in C?

- A) auto
- B) register
- C) friend
- D) extern

Question 3

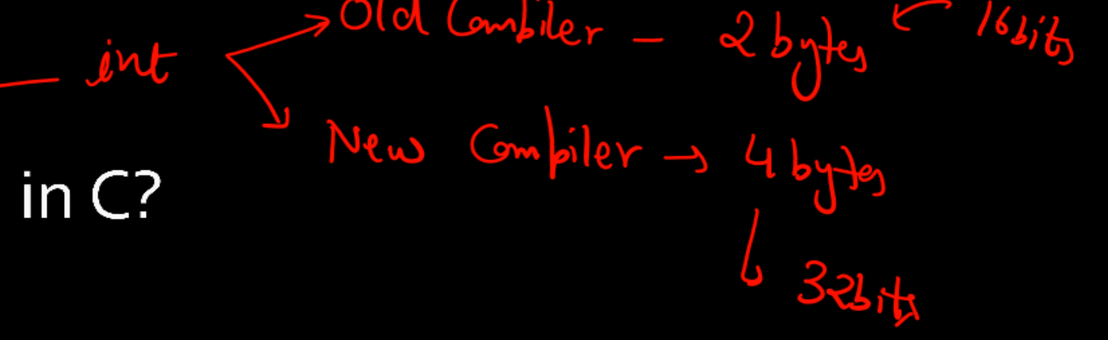
3. Which of the following is a valid identifier in C?

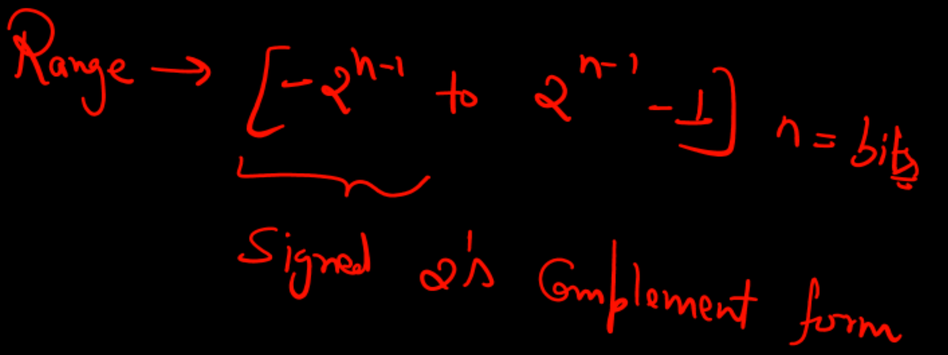
- A) 2var ← Not Start with Number
- B) _var2 ✓
- C) var-2 ← Special Symbol
- D) var 2 ← Space is NOT allowed

Question 4

What is the range of an integer literal in C?

- A) Depends on the compiler
- B) -32768 to 32767
- C) -2147483648 to 2147483647
- D) 0 to 65535





Question 5

- 5. Which of the following is a valid character literal in C?
- A) 'a' ← Single quote
- B) "a"
- C) 'abc'
- D) "a'"

Question 6

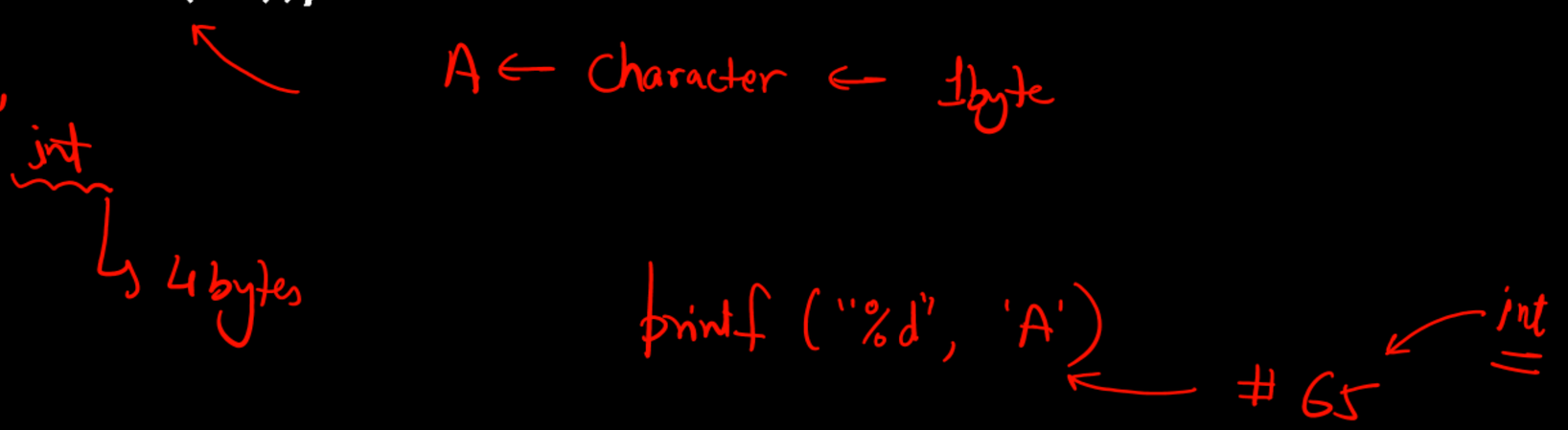
- 6. Which escape sequence represents a newline character in C?
- A) \n
- B) \t
- C) \b
- D) \\

Question 7

7. What will be the output of the following code snippet?

```
printf("%d", sizeof('A'));
```

- A) 1
- B) 2
- C) 4
- D) 8



Question 8

8. Which of the following is not a type of literal in C?

- A) Integer literal
- B) Floating-point literal
- C) Character literal
- D) Pointer literal

Question 9

9. Which operator is used for conditional execution in C?

- A) &&
- B) ||
- ~~C) ?:~~
- D) ==

↳ ternary operator

↳ ? :

Question 10

10. What is the significance of keywords in C?

A) They are user-defined names

B) They are pre-defined, reserved words used by the compiler

C) They can be used as variable names

D) They are only used in header files

Question 11

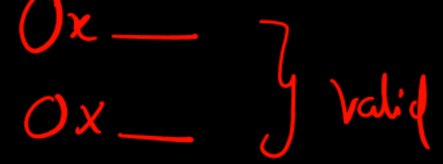
11. Which of the following is not a valid integer literal in C?

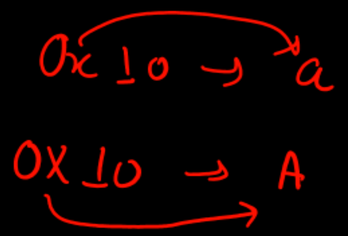
- A) 10 ← decimal integer
- B) 010 ← Octal integer
- C) 0x10 ← Hex Code
- D) 0b10

Question 12

12. Which is the correct representation of a hexadecimal integer literal in C?

- A) ox1F
- B) oX1F
- C) ox1g
- D) Both A and B



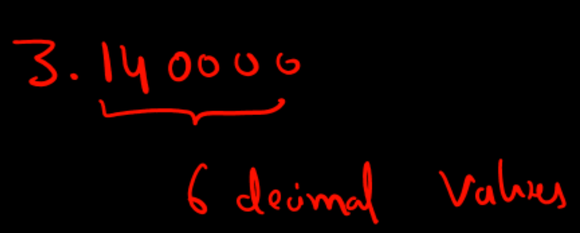


Question 13

13. What does the following code output?

```
printf("%f", 3.14);
```

- A) 3.140000
- B) 3.14
- C) 3
- D) 3.140



Question 14

14. What is a token in the context of C programming?

A) A single element of a program, such as a keyword or operator

B) A function name

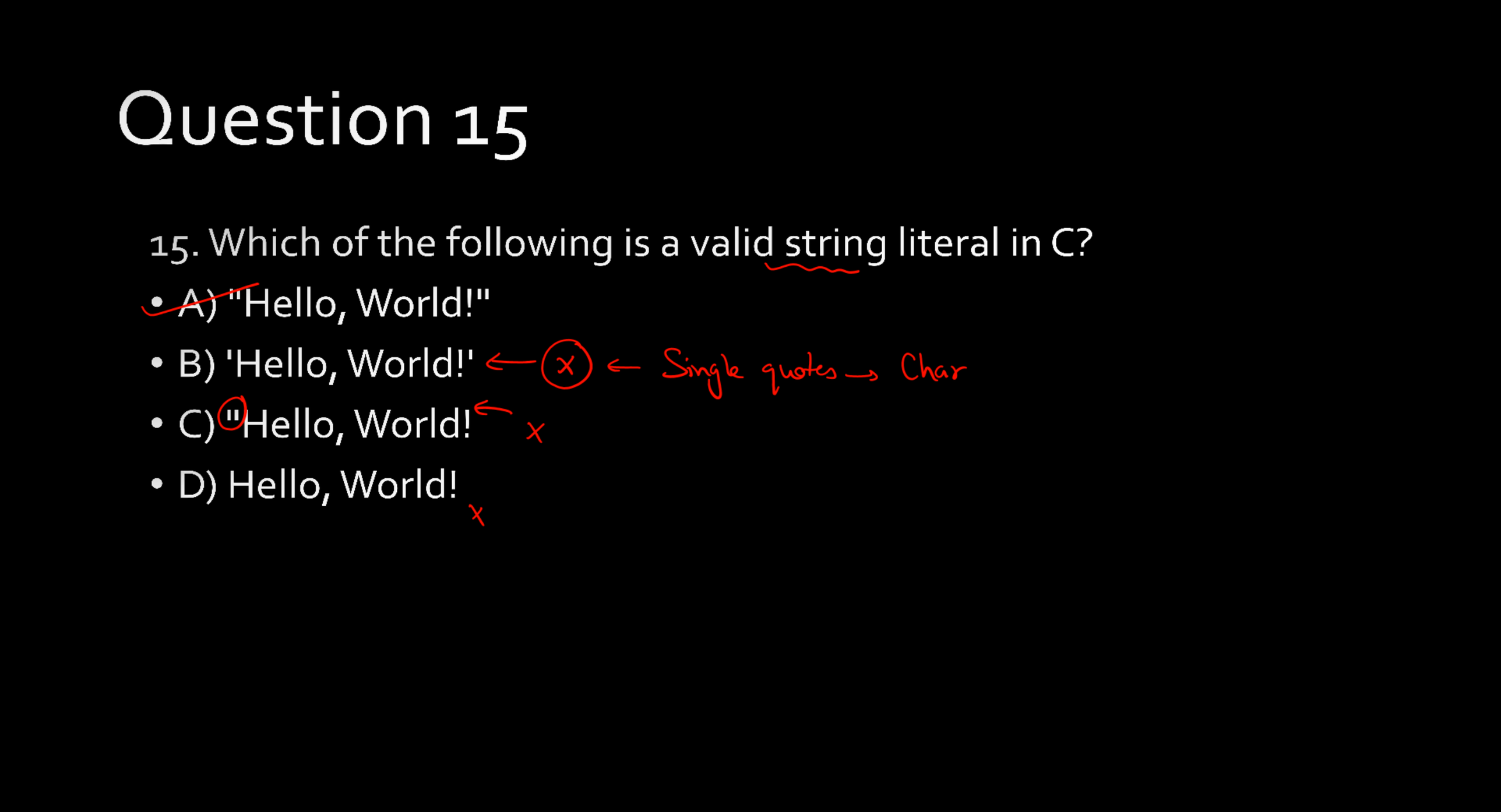
C) A variable that stores data

D) None of the above

Question 15

15. Which of the following is a valid string literal in C?

- A) "Hello, World!"
- B) 'Hello, World!'
- C) "Hello, World!
- D) Hello, World!



Question 16

16. Which of the following literals represents a double in C?

- A) 123.4
- B) 123.4f ← float
- C) 123 ← int
- D) '123' ← ✗

Double is default for any number having a point.

Question 17

17. Which of the following is a correct comment in C?

- A) /* This is a comment */
- B) // This is a comment
- C) Both A and B
- D) None of the above

/* —— */ → Multiline

// → Single line

Question 18

18. Which of the following represents an octal literal in C?

A) 10 ← int

B) 010 ← Octal

C) 0X10 ← Hex

D) 100 → int

* Any integer starts from 'zero' Considered as Octal.

Question 19

19. What is the value of oxFF in decimal?

A) 255

B) 256

C) 512

D) 1024

Other Base → decimal

oxFF → Hex

$F_{16^1} + F_{16^0} = F \times 16^1 + F \times 16^0$

$15 \times 16 + 15$

$240 + 15$

$= 255$

Question 20

20. Which of the following literals is not supported directly in C?

A) String literal ← array of character

B) Boolean literal

C) Integer literal

D) Floating-point literal

Question 21

The Containers whose value doesnot Change.

21. Which of the following correctly declares a constant integer in C?

A) int const a = 10;

B) const int a = 10;

C) #define a 10

D) All of the above

int x = 10;

x++;

variable

x => 11

'const' Keyword is used to make a Value Constant

← Preprocessing

include
define
} Macros

#define x 10

fix the value of x = 10 for whole program.

Const int x = 10; or int Const x = 10;

x++ ← Error

Question 22

22. What is the value of the expression sizeof(123.45F)?

A) 4

B) 8

C) 2

D) 1

float
↳ 4 bytes

Question 23

23. Which keyword is used to declare a variable that cannot be modified?

Read only

A) static

B) volatile

C) const

D) register

Question 24

24. What is the ASCII value of the character literal 'A' in C?

A) 65

B) 66

C) 97

D) 98

A → 65

a → 97

Question 25

25. Which of the following is a valid floating-point literal in C?

A) 3.14

B) 3.

C) 3.0

D) All of the above

float m = 3.; ← allowed

Question 26

26. Which is not a valid identifier in C?

A) int1

B) _var

C) break ← Keyword

D) Var_1

Question 27

27. Which of the following statements is true regarding string literals in C?

A) They are enclosed in single quotes.

B) They are enclosed in double quotes.

C) They can contain newline characters without escape sequences.

D) None of the above.

Question 28

28. What does the term "whitespace" refer to in the context of C tokens?

A) Characters like spaces, tabs, and newlines that are ignored by the compiler.

B) Characters used in string literals.

C) Comments in the code.

D) Unused memory.

Question 29

29. Which of the following escape sequences is incorrect in C?

A) \a

B) \b

C) \v

D) \x

All Correct

Question 30

30. Which of the following is a valid representation of a float literal in C?

A) 3.14e2 ← 3.14×10^2 $e = 10$

B) 3.14E-2 ← 3.14×10^{-2}

C) 3.14f → float

D) All of the above

Question 31

31. What type of literal is the following in C: 012?

A) Decimal integer

B) Octal integer

C) Hexadecimal integer

D) Floating-point number

Question 32

33. Which of the following is a valid hexadecimal floating-point literal in C?

A) 0x1.1p2 → 1.1×16^{2}

B) 0x1.2p-2 ← 1.2×16^{-2}

C) 0x.1p1 → 0.1×16^{1}

D) All of the above

p = 16

Question 33

34. Which of the following is not considered an operator in C?

A) +

B) sizeof

C) &

D) and

Question 34

35. Which of the following identifiers is valid according to C naming conventions?

A) int*var ← invalid

B) var_name ✓

C) var-name ✗

D) 3var ← Number

Question 35

36. What will be the output of the following C code?

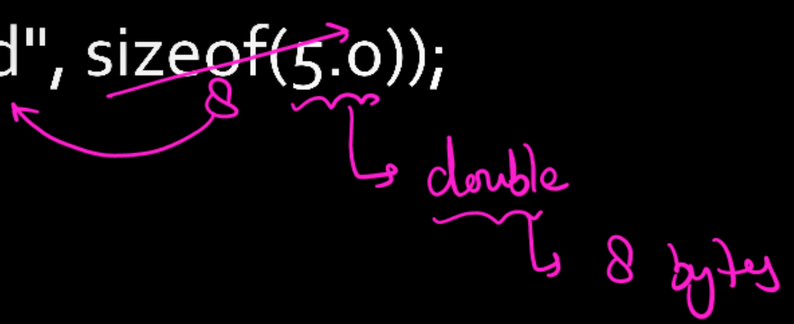
```
printf("%d", sizeof(5.0));
```

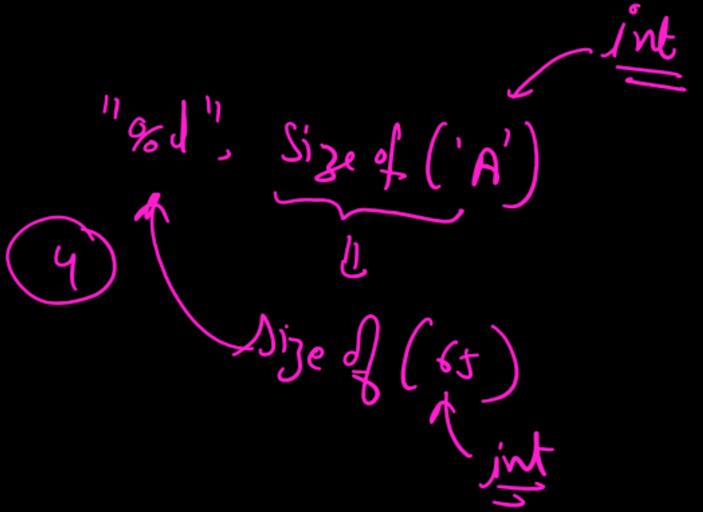
A) 2

B) 4

C) 8

D) Undefined





Question 36

Change back to its previous state
अचानक से बदल जाना

37. What does the keyword volatile signify in C?

A) The variable is stored in CPU registers.

B) The value of the variable can be changed unexpectedly.

C) The variable can be modified only once.

D) The variable is used in multithreading.

Question 37

38. Which of the following is an example of an integer constant in C?

A) 5.0 ← float

B) 5L ← long int

C) 5.0f ← float

D) '5' ← char

Question 38

39. Which of the following is not an escape sequence in C?

A) \r ← Carriage return

B) \t

C) \s

D) \n

Question 39

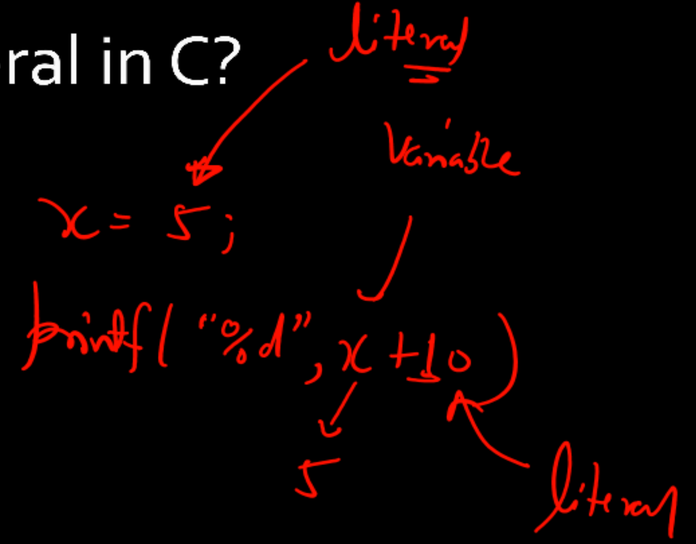
40. Which of the following correctly describes a literal in C?

A) A variable that holds a value

B) A function that returns a constant

C) A fixed value used directly in the code

D) A keyword used to define a constant



Question

Which of the following is not a basic data type in C?

Options:

a) int

b) float

c) double

d) string

Question

Which of the following data types can store a single character in C?

Options:

a) char

b) int

c) float

d) double

Question

What is the output of the following code snippet?

```
int a = 5;
float b = 5.5;
printf('%d', a + b);
```

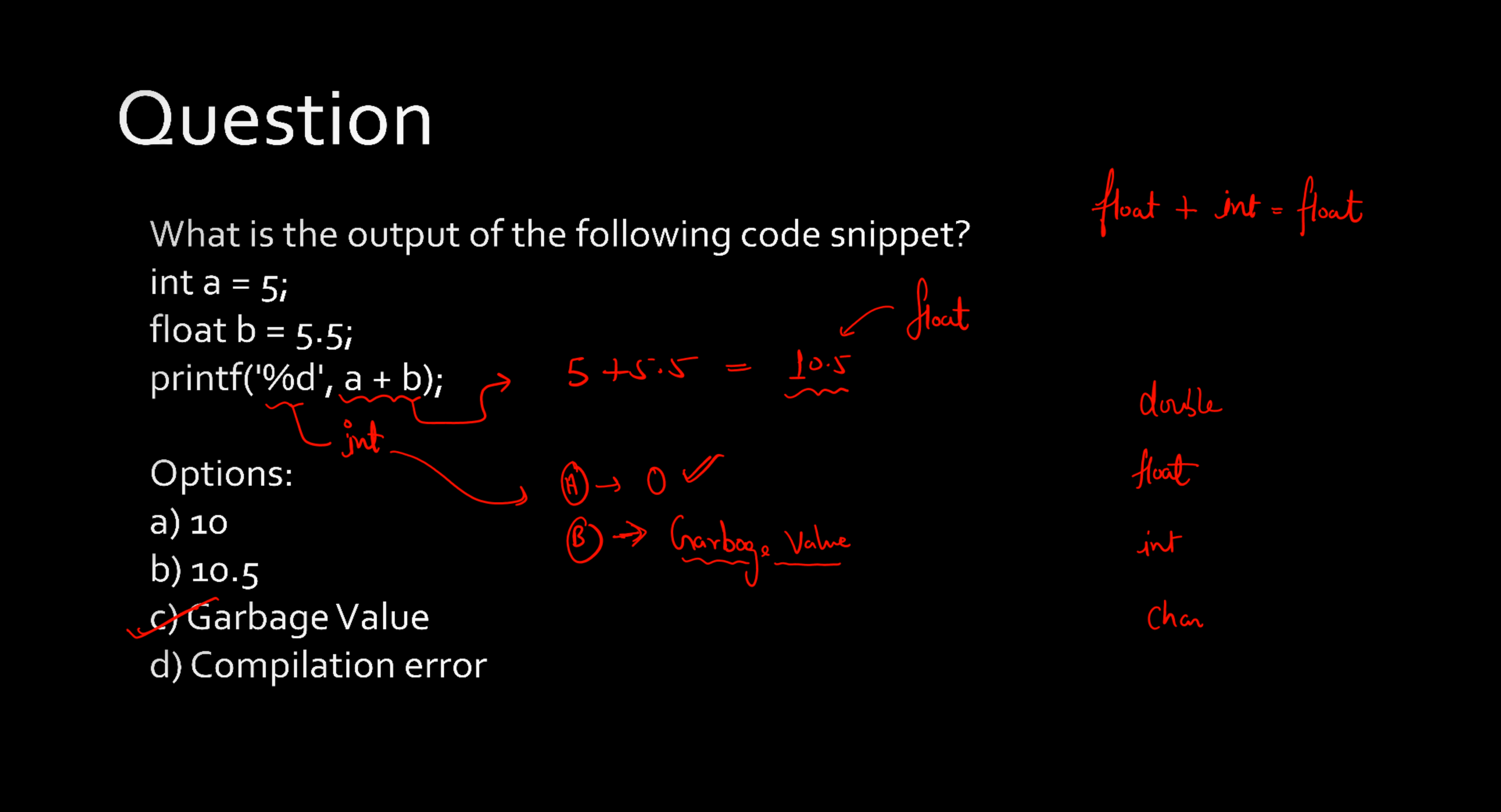
Options:

a) 10

b) 10.5

c) Garbage Value

d) Compilation error



Question

What is the range of a signed char data type in C?

2's Complement

1 byte = 8 bits

Range = -2^{n-1} to $2^{n-1}-1$

= -2^7 to 2^7-1

= -128 to 127

Options:

a) 0 to 255

b) -128 to 127

c) -32768 to 32767

d) 0 to 65535

Question

Which of the following statements is correct about the float data type in C?

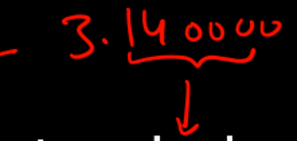
Options:

a) It can store integers only.

b) It can store fractional numbers up to 6 decimal places.

c) It is used to store large integers.

d) It occupies 8 bytes of memory.



Question

What is the difference between int and unsigned int data types in C?

Options:

a) int cannot be negative, but unsigned int can.

b) unsigned int cannot be negative, but int can.

c) Both are the same.

d) int occupies more memory than unsigned int.

Question

Which data type should be used to store a very large integer value in C?

Options:

a) int

b) char

c) long long int

d) float

(type specifiers / type quantifier)

Question

What is the size of the long double data type in C on most 64-bit systems?

double = 8 bytes

long double = 16 bytes

Options:

a) 4 bytes

b) 8 bytes

c) 10 bytes

d) 16 bytes

Question

What is the output of the following code snippet?

```
int a = 10;
printf('%u', sizeof(a));
```

Options:

a) 2

b) 4

c) 8

d) 10

Question

Which of the following data types can be used to store a boolean value in C?

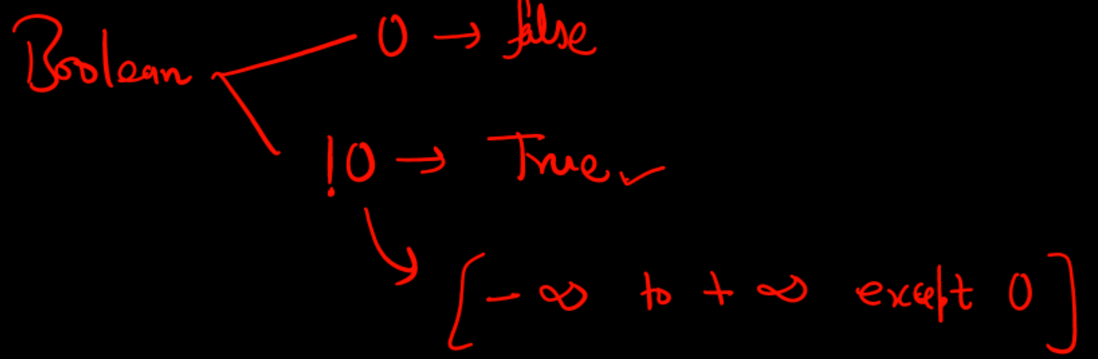
Options:

a) int

b) bool

c) _Bool

d) double



Question

Which format specifier is used to print a short int in C?

2 bytes

Options:

a) %d

b) %hd

c) %ld

d) %f

Question

What will be the output of the following code snippet?

```
int a = 32768;
short b = a;
printf('%d', b);
```

Options:

a) 32768

b) -32768

c) 0

d) 65536

max = 32767

min = -32768

a = 32768

b = 32768

Short = 2 bytes → 16 bits

Range = -2^{16-1} to $2^{16-1}-1$

$= [-2^{15}$ to $2^{15}-1]$

32768 to 32768-1

[-32768 to 32767]

Question

Implicit Conversion → automatic Conversion
Eg 3 + 5.0f = 8.0f
int + float = float ← Implicit

Explicit Conversion
↳ manually

Which type conversion is automatically performed by the C compiler?

3 + (int) 5.0 = 8 ← int ← Explicit
(int) 5.0 ← manual

3 + 5.0f = 8.000000 f ← implicit

Options:

a) Implicit conversion

b) Explicit conversion

c) User-defined conversion

d) None of the above

Question

Which of the following statements is true about the sizeof operator in C?

Options:

a) sizeof is a function.

b) sizeof is a macro.

c) sizeof is an operator.

d) sizeof is a keyword.

Question

What is the output of the following code?

```
int x = 0;
printf('%d', sizeof(x++));
```

x = 0

int

int = 4 byte

Options:

a) 0

b) 1

c) 4

d) 8

Question

Which of the following is a valid type specifier in C?

Options:

a) unsigned float

b) signed int

c) unsigned double

d) signed char

Question

What will happen if you try to store a value greater than the maximum limit of unsigned char in C?

Options:

a) Compilation error

b) The value wraps around to the minimum limit

c) The value remains unchanged

d) Runtime error

Question

Which data type is typically used to represent an object that holds memory addresses?

Options:

a) int

b) float

c) pointer

d) char

Question

What is the maximum value of unsigned int on a 32-bit system?

Options:

a) 32767

b) 65535

c) 2147483647

d) 4294967295

4 bytes = 32 bits

Range = (0 to $2^n - 1$)

$2^{32} - 1$

Question

What will be the output of the following code snippet?

```
float f = 3.14;
printf('%d', (int)f);
```

float = 4bytes

3 ← int

Explicit Conversion

Options:

a) 3

b) 3.14

c) 314

d) Compilation error