

C sizeof Operator (Question):

Sayak Haldar
IEST, Shibpur

1. What is the sizeof(char) in a 32-bit C compiler?

- a. 1 bit
- b. 2 bits
- c. 1 Byte
- d. 2 Bytes

2. What is the output of this C code?

- ```
1. #include <stdio.h>

2. printf("%d", sizeof('a'));
```

- a. 1
- b. 2
- c. 4
- d. None of the mentioned

**3. Size of an array can be evaluated by:**

(Assuming array declaration `int a[10];`)

- a. `sizeof(a);`
- b. `sizeof(*a);`
- c. `sizeof(a[10]);`
- d. `10 * sizeof(a);`

**4. What is the output of this C code?**

- ```
1.  #include <stdio.h>

2.  union temp

3.  {

4.      char a;

5.      char b;

6.      int c;

7.  }t;

8.  int main()

9.  {
```

```
10.    printf("%d", sizeof(t));  
11.    return 0;  
12. }
```

- a. 1
- b. 2
- c. 4
- d. 6

5. Which among the following has the highest precedence?

- a. &
- b. <<
- c. sizeof()
- d. &&

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

- a. ,
- b. sizeof()
- c. ~
- d. None of the mentioned

7. The sizeof(void) in a 32-bit C is_____.

- a. 0
- b. 1
- c. 2
- d. 4

8. What type of value does sizeof return?

- a. char
- b. short
- c. unsigned int
- d. Long

9. What will be the size of the following structure?

```
1.    #include <stdio.h>  
2.    struct temp  
3.    {  
4.        int a[10];
```

5. char p;

6. };

- a. 5
- b. 11
- c. 41
- d. 44

10. Comment on the output of following C program?

1. #include <stdio.h>

2. main()

3. {

4. int a = 1;

5. printf("size of a is %d, ", sizeof(++a));

6. printf("value of a is %d", a);

7. return 0;

8. };

- a. size of a is 4, value of a is 1
- b. size of a is 4, value of a is 2
- c. size of a is 2, value of a is 2
- d. size of a is 2, value of a is 2

11. Which among the following is right?

- a. sizeof(struct stemp*) > sizeof(union utemp*) > sizeof(char *)
- b. sizeof(struct stemp*) < sizeof(union utemp*) < sizeof(char *)
- c. sizeof(struct stemp*) = sizeof(union utemp*) = sizeof(char *)
- d. The order Depends on the compiler

12. Comment on the following C code?

1. #include <stdio.h>

2. printf("%d", sizeof(strlen("HELLOWORLD")));

- a. Output, 4
- b. Output, 10
- c. Output, 16

d. Error, sizeof cannot evaluate size of a function.

13. Which of the following cannot be used inside sizeof?

- a. pointers
- b. functions
- c. macro definition
- d. None of the mentioned

14. Comment on the following C code?

1. `#include <stdio.h>`
2. `//(sizeof double = 8, float = 4, void = 1)`
3. `#define PI 3.14`
4. `int main()`
5. `{`
6. `printf("%d", sizeof(PI));`
7. `return 0;`
8. `}`

- a. Output is 8
- b. Output is 4
- c. Output is 1
- d. Error, we can't use sizeof on macro-definitions