

3 Lecture - CS410

Important Mcqs

1. Question: In C, how do you declare an integer array named "numbers" with 5 elements?

- a) array numbers[5];
- b) int numbers[5];
- c) int[] numbers = {5};
- d) int numbers(5);

Solution: b) int numbers[5];

2. Question: What is the value of the expression "sizeof(numbers)" in C, where "numbers" is an integer array with 10 elements?

- a) 10
- b) 40
- c) 4
- d) 14

Solution: b) 40

3. Question: In C, how do you access the third element of an array named "data"?

- a) data(3);
- b) data[3];
- c) data{3};
- d) data.3;

Solution: b) data[3];

4. Question: What is the correct way to pass an array "arr" to a function in C?

- a) function(arr);
- b) function(arr[]);

c) `function(&arr);`

d) `function(*arr);`

Solution: b) `function(arr[]);`

5. Question: What is a pointer in C?

a) A variable that stores multiple values

b) A variable that stores the address of another variable

c) An array that points to another array

d) A function that points to another function

Solution: b) A variable that stores the address of another variable

6. Question: What does the "*" symbol represent when used with a pointer variable in C?

a) Multiplication

b) Exponentiation

c) Address of a variable

d) Dereferencing the pointer

Solution: d) Dereferencing the pointer

7. Question: How do you declare a pointer variable named "ptr" that points to an integer in C?

a) `int* ptr;`

b) `ptr* int;`

c) `pointer ptr = int;`

d) `ptr = int*;`

Solution: a) `int* ptr;`

8. Question: What is the value of "ptr" after the following code: `"int num = 10; int* ptr = #"`?

a) 10

b) The address of "num"

c) The address of "ptr"

d) Garbage value

Solution: b) The address of "num"

9. Question: What happens when you increment a pointer in C using "ptr++"?

- a) The pointer points to the previous element.
- b) The pointer points to the next element.
- c) The pointer becomes NULL.
- d) The pointer points to the first element.

Solution: b) The pointer points to the next element.

10. Question: How do you dynamically allocate memory for an integer array "arr" of size 5 in C?

- a) `int arr[5];`
- b) `int arr = (int*)malloc(5);`
- c) `int* arr = new int[5];`
- d) `int* arr = (int*)malloc(5 * sizeof(int));`

Solution: d) `int* arr = (int*)malloc(5 * sizeof(int));`