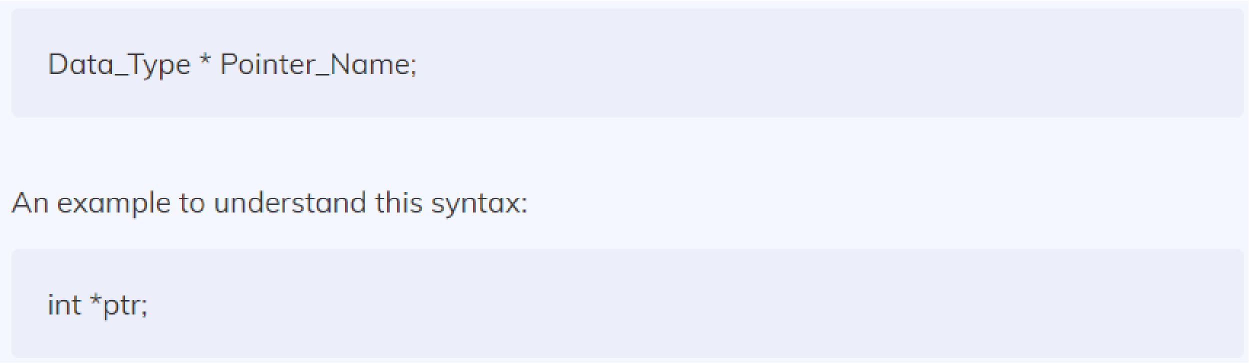
Pointers Questions

What is a pointer?

This is said to be a very basic pointer interview question. Pointer is a variable that stores the address of another variable. The syntax of a pointer is represented as



```
Data_Type * Pointer_Name;
```

An example to understand this syntax:

```
int *ptr;
```

Describe the usage of pointers in C.

Some of the areas where pointers are used are:

- **To access array elements**
- **Used to return multiple values.**
- **Used in Dynamic memory Allocation.**
- **To pass arguments by reference**
- **Reduces the execution time of the program.**
- **Pointers help us to build complex data structures like a linked list, stack, queues, trees, and graphs.**

What is a Null Pointer?

A null reference or null pointer is a value saved for indicating that the reference or pointer does not refer to a valid object. They can be used to stop indirection in a recursive Data Structure, as an error value or as a sentinel value.

What is the size of a generic pointer?

For a system of 16-bit, the size of a generic pointer is 2 bytes. If the system is 32-bit, the size of a generic pointer is 4 bytes. If the system is 64-bit, the size of a generic pointer is 8 bytes.

What is a dangling pointer in C?

A pointer pointing to a non-existing memory location is a dangling pointer. A dangling pointer is a pointer that has a value (not NULL) which refers to some memory that is not valid or does not exist.

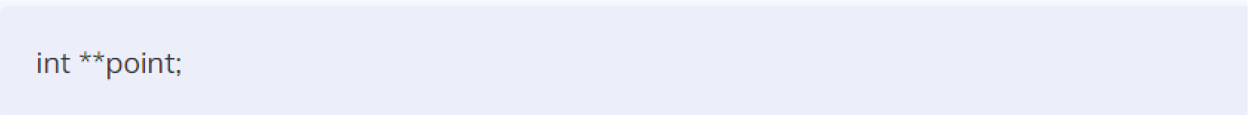
What is a generic pointer? When can we use a generic pointer?

When a variable is declared void type, it is known as a generic pointer. It is a pointer that can point to any data. They are used when we want to return such a pointer which applies to all types of pointers. They can also be used to increase the re-usability of the pointer.

Explain the term double-pointer.

If a pointer holds another pointer's address, then such pointer is known as double-pointer or pointer-to-pointer.

For example,



```
int **point;
```

What is a wild pointer?

A pointer that is not initialized properly before its first use is known as the wild pointer. They are called so because the uninitialized pointer’s behavior is undefined as it may point to some arbitrary location that can cause the program to crash. Generally, compilers warn about the wild pointer.

Explain near, far, and huge pointers?

- **A far pointer is size 32 bit, which includes a segment selector, making it possible to point the addresses outside of the default segment.**

- **Near pointer is utilized for storing 16-bit addresses within the current segment on a 16-bit machine. The drawback is that we can only access 64kb of data each time.**

- **The huge pointer is also 32-bit and can access outside segments. In the far pointer, Huge can be changed but the segment part cannot be changed.**

Explain the meaning.

1. const int ptr;

2. const int *ptr;

3. int * const ptr;

4. int const * a const;

- The "ptr" is a constant integer.
- Here "ptr" is a pointer to a const integer, the value of the integer is not modifiable, but the pointer is not modifiable.
- Here "ptr" is a constant pointer to an integer which means that the value of the pointed integer is changeable, but the pointer is not modifiable.
- Here "a" is a const pointer to a const integer which means the value of the pointed integer and pointer both cannot be changed.

What is Dereference or Indirection Operator (*)?

A dereference operator is a unary operator that is used in the declaration of the pointer and accesses a value indirectly, through a pointer. The operand of such operator should be a pointer and the result of the operation is value addressed by the operand (pointer).

For example,

- int *iPointer; // Use of indirection operator in the declaration of pointer
- a = *iPointer; //Use of indirection operator to read the value of the address pointed by the pointer
- *iPointer = a; //Use of indirection operator to write the value to the address pointed by pointer.

How many pointers can point to the same address?

Multiple pointers can point to the same memory address.

Where is the pointer variable stored in memory?

Pointer variables are stored in stack memory.

What is an array of pointers?

It is an array of pointer variables.

Declaration:

data_type *name_of_array[array_size];

Example:

int *a[5];

Here “a” is an array of 5 integer pointers.

What is the difference between arrays and pointers?

Arrays are used to store elements of the same type whereas Pointers are address variables that store the address of a variable.

Is the pointer faster than the array?

Arrays will always be faster as memory allocation of an array is continuous. So, accessing an array is much faster compared to a pointer where memory allocation might or might not be continuous.