Difference between pointer to an array and array of pointers

[Pointer to an array](https://www.geeksforgeeks.org/pointer-array-array-pointer/):

[Pointer to an array](https://www.geeksforgeeks.org/pointer-array-array-pointer/) is also known as [array pointer](https://www.geeksforgeeks.org/pointer-array-array-pointer/). We are using the pointer to access the components of the array.

int a[3] = {3, 4, 5 };

int \*ptr = a;

We have a pointer ptr that focuses to the 0th component of the array. We can likewise declare a pointer that can point to whole array rather than just a single component of the array.

**Syntax**:

data type (\*var name)[size of array];

Declaration of the pointer to an array:

// pointer to an array of five numbers

int (\* ptr)[5] = NULL;

The above declaration is the pointer to an array of five integers. We use parenthesis to pronounce pointer to an array. Since subscript has higher priority than indirection, it is crucial to encase the indirection operator and pointer name inside brackets.

**Example:**

// C program to demonstrate

// pointer to an array.

#include <stdio.h>

int main()

{

// Pointer to an array of five numbers

int(\*a)[5];

int b[5] = { 1, 2, 3, 4, 5 };

int i = 0;

// Points to the whole array b

a = &b;

for (i = 0; i < 5; i++)

printf("%d\n", \* (\*a + i));

return 0;

}

**Output:**

1

2

3

4

5

[Array of pointers](https://www.geeksforgeeks.org/pointers-in-c-and-c-set-1-introduction-arithmetic-and-array/):

“[Array of pointers](https://www.geeksforgeeks.org/pointers-in-c-and-c-set-1-introduction-arithmetic-and-array/)” is an array of the [pointer variables](https://www.geeksforgeeks.org/pointers-c-examples/). It is also known as pointer arrays.

**Syntax**:

int \*var\_name[array\_size];

Declaration of an array of pointers:

int \*ptr[3];

We can make separate pointer variables which can point to the different values or we can make one integer array of pointers that can point to all the values.

**Example:**

// C program to demonstrate

// example of array of pointers.

#include <stdio.h>

const int SIZE = 3;

void main()

{

// creating an array

int arr[] = { 1, 2, 3 };

// we can make an integer pointer array to

// storing the address of array elements

int i, \*ptr[SIZE];

for (i = 0; i < SIZE; i++) {

// assigning the address of integer.

ptr[i] = &arr[i];

}

// printing values using pointer

for (i = 0; i < SIZE; i++) {

printf("Value of arr[%d] = %d\n", i, \*ptr[i]);

}

}

**Output:**

Value of arr[0] = 1

Value of arr[1] = 2

Value of arr[2] = 3

**Example:** We can likewise make an array of pointers to the character to store a list of strings.

**#include <stdio.h>**

**const int size = 4;**

**void main()**

**{**

**// array of pointers to a character**

**// to store a list of strings**

**char\* names[] = {**

**"amit",**

**"amar",**

**"ankit",**

**"akhil"**

**};**

**int i = 0;**

**for (i = 0; i < size; i++) {**

**printf("%s\n", names[i]);**

**}**

**}**

**Output:**

amit

amar

ankit

akhil