

C - Array of pointers

Before we understand the concept of arrays of pointers, let us consider the following example, which uses an array of 3 integers –

```
#include <stdio.h>

const int MAX = 3;

int main () {

    int var[] = {10, 100, 200};
    int i;

    for (i = 0; i < MAX; i++) {
        printf("Value of var[%d] = %d\n", i, var[i] );
    }

    return 0;
}
```

[Live Demo](#)

When the above code is compiled and executed, it produces the following result –

```
Value of var[0] = 10
Value of var[1] = 100
Value of var[2] = 200
```

There may be a situation when we want to maintain an array, which can store pointers to an int or char or any other data type available. Following is the declaration of an array of pointers to an integer –

```
int *ptr[MAX];
```

It declares **ptr** as an array of MAX integer pointers. Thus, each element in ptr, holds a pointer to an int value. The following example uses three integers, which are stored in an array of pointers, as follows –

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

[Live Demo](#)

```
const int MAX = 3;
```

```
int main () {
```

```
    int var[] = {10, 100, 200};
```

```
    int i, *ptr[MAX];
```

```
    for ( i = 0; i < MAX; i++) {
```

```
        ptr[i] = &var[i]; /* assign the address of integer. */
```

```
    }
```

```
    for ( i = 0; i < MAX; i++) {
```

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

```
    }
```

```
    return 0;
```

```
}
```

When the above code is compiled and executed, it produces the following result –

```
Value of var[0] = 10
```

```
Value of var[1] = 100
```

```
Value of var[2] = 200
```

You can also use an array of pointers to character to store a list of strings as follows –

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

[Live Demo](#)

```
const int MAX = 4;
```

```
int main () {
```

```
char *names[] = {  
    "Zara Ali",  
    "Hina Ali",  
    "Nuha Ali",  
    "Sara Ali"  
};  
  
int i = 0;  
  
for ( i = 0; i < MAX; i++) {  
    printf("Value of names[%d] = %s\n", i, names[i] );  
}  
  
return 0;  
}
```

When the above code is compiled and executed, it produces the following result –

```
Value of names[0] = Zara Ali  
Value of names[1] = Hina Ali  
Value of names[2] = Nuha Ali  
Value of names[3] = Sara Ali
```