C++ Array of Pointers

tutorialspoint.com/cplusplus/cpp_array_of_pointers.htm

Advertisements

Previous Page
Next Page

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

Live Demo

```
#include <iostream>

using namespace std;
const int MAX = 3;

int main () {
   int  var[MAX] = {10, 100, 200};

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

      cout << "Value of var[" << i << "] = ";
      cout << var[i] << endl;
   }

   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
```

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];
```

This declares **ptr** as an array of MAX integer pointers. Thus, each element in ptr, now holds a pointer to an int value. Following example makes use of three integers which will be stored in an array of pointers as follows –

Live Demo

```
#include <iostream>

using namespace std;
const int MAX = 3;

int main () {
   int  var[MAX] = {10, 100, 200};
   int *ptr[MAX];

   for (int i = 0; i < MAX; i++) {
      ptr[i] = &var[i]; // assign the address of integer.
   }

   for (int i = 0; i < MAX; i++) {
      cout << "Value of var[" << i << "] = ";
      cout << *ptr[i] << endl;
   }

   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 –

Live Demo

```
#include <iostream>

using namespace std;
const int MAX = 4;

int main () {
const char *names[MAX] = { "Zara Ali", "Hina Ali", "Nuha Ali", "Sara Ali" };

   for (int i = 0; i < MAX; i++) {
      cout << "Value of names[" << i << "] = ";
      cout << (names + i) << endl;
   }

   return 0;
}
```

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

```
Value of names[0] = 0x7ffd256683c0
Value of names[1] = 0x7ffd256683c8
Value of names[2] = 0x7ffd256683d0
Value of names[3] = 0x7ffd256683d8
```

Advertisements