Function With Array

In C, there are many situations where passing multiple variables of the same type to a function is necessary. For instance, consider a function that sorts 10 elements in ascending order. To achieve this, 10 numbers need to be passed as actual parameters from the main function. Instead of declaring 10 separate variables and passing them individually, we can declare and initialize an array, then pass it to the function. This simplifies the process since the function can handle any number of values.

Since the array name represents the address of its first element, it's important to note that we only need to pass the array name to the function that expects an array. The formal parameter defined as an array will automatically reference the array passed as the actual parameter.

Consider the following syntax to pass an array to the function.

```
functionname(arrayname);//passing array
```

Methods to declare a function that receives an array as an argument

There are 3 ways to declare the function which is intended to receive an array as an argument.

First Ways

```
return_type function(type arrayname[])
```

Second Ways

```
return_type function(type arrayname[])
```

Third Ways

```
return_type function(type *arrayname)
```

Example:

```
#include <stdio.h>
void display(int age1, int age2) {
  printf("%d\n", age1);
  printf("%d\n", age2);
}
int main() {
  int ageArray[] = {2, 8, 4, 12};
  // pass second and third elements to display()
  display(ageArray[1], ageArray[2]);
  return 0;
}
```