

# Arrays and function calls Solutions

- How are arrays usually passed as function arguments?
  - Built-in arrays are usually passed by address
- Explain how this mechanism works
  - The name of the array decays into a pointer to the first element

- Give an example prototype for a function which takes an array as argument

```
void somefunc(int *pi);           // Function that takes pointer to int
```

```
void somefunc(int[] pi);         // Alternative syntax
```

- How can the array's elements be accessed inside the function?
  - The array's elements can be accessed using array notation or pointer offsets

- Why can passing built-in arrays to functions be unsafe?
  - The function cannot find out how many elements the array has
- What can be done to avoid this problem?
  - Use `std::array`
  - If built-in arrays must be used, pass the number of elements as a separate argument
- Write a simple program which creates an array and passes it as an argument to a function. The function should print out the elements of the array

- How can built-in arrays be passed by reference to a function?
  - The number of elements must be known at compile time
  - The syntax looks like this

```
void func3(int (&pi)[4]);    // Reference to array of 4 ints
```

- Alter your program so that the array is passed by reference