ARRAYS

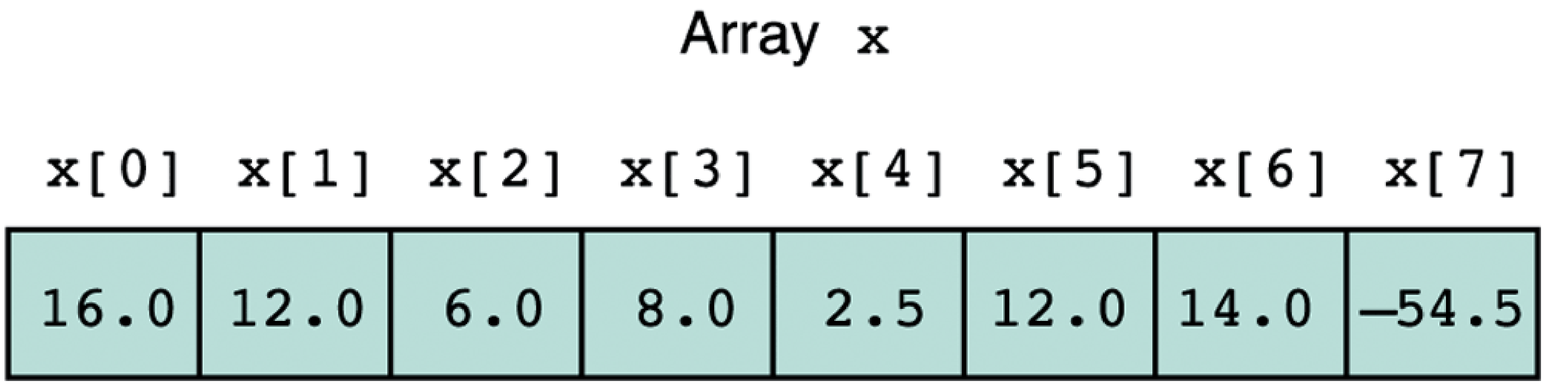
BEESHANGA ABEWARDANA JAYAWICKRAMA

UTS:
ENGINEERING AND
INFORMATION
TECHNOLOGY

OVERVIEW OF ARRAYS

Arrays allocate a block of contiguous memory to store a <u>fixed</u> number of data elements of the same type.

```
double x[8];
```



STATEMENTS THAT MANIPULATE ARRAY X

```
printf("%.1f", x[0]);

x[3] = 25.0;

sum = x[0] + x[1];

sum += x[2];

x[3] += 1.0;

x[2] = x[0] + x[1];
```

Array index starts from 0 and goes up to size -1.

ARRAYS AND FUNCTIONS

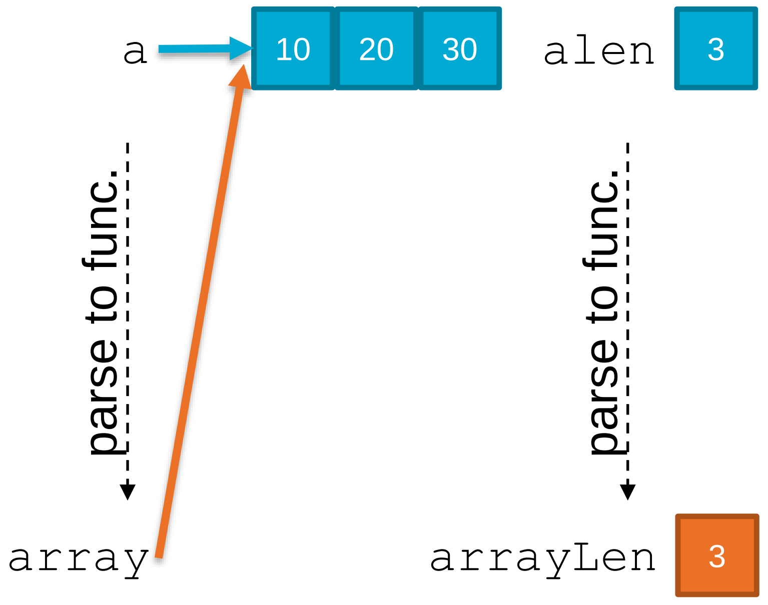
```
int main(void)
{
    int alen = 3;
    int a [] = {10, 20, 30};

    somefun(a, alen);

}

void somefun(int array[],int arrayLen)
{
    ...


}
```



Functions do NOT make local copies of arrays.
Any change made to the array inside the function will modify the array in main.

ARRAYS AND FUNCTIONS

When passing arrays to functions

- > Together with the arrays, always pass the size of the array as an int variable. Function has no direct way to find out the size of the array.
- > A local copy is not made, instead any change you make to an array inside a function will be reflected in the original copy that was passed to the function.
  - > However if only an element of the array is passed to a function, a local copy is made.
  - > Use the const keyword when passing an array to a function that should remain unchanged.
  - > Can a function return an array?

AN ARRAY HAS A FIXED SIZE

An array has a fixed size. What if we don't know the exact size we need?

One possible solution

> At least we must know the maximum possible size.

> Allocate a large array of maximum size, but use only the currently required portion.

> Use an int to keep track of the used portion.

A better solution: wait until you learn about pointers ☺

STRINGS

BEESHANGA ABEWARDANA JAYAWICKRAMA

UTS:
ENGINEERING AND
INFORMATION
TECHNOLOGY

STRINGS IN C

String is a word/sentence/paragraph/etc. C has NO data type called string.

But effectively a string is a block of characters

> A string can be stored as an array of char

Example:

```
char str[] = "Bee J";
printf("%s\n", str);
```

Food for thought: Previously we said when passing arrays to functions we should pass the size of the array as an int. How does `printf` know the size of the `str` array?

NULL TERMINATION

Strings in C must be terminated with a null character i.e. '\0'

```
char str[] = "Bee J";
```

This is stored in memory as:

'B'	'e'	'e'	' '	'J'	'\0'

The size of the array str is 6, NOT 5. The contrary is, if you want to store a string that is n chars long, you need an array of size n+1 (to append a NULL character).

Printf displays all characters in memory until '\0' is found.

STRING.H

Food for thought: Can we do the following?

```
char str[6];

str = "Bee J";
```

NO. str is an array, so have to assign each char separately, including the NULL character.

```
str[0] = 'B'; str[1]='e'; str[2]='e'; str[3]=' ';
str[4]='J'; str[5] = '\0';
```

-or- use string.h `strcpy (str, "Bee J")`

Other useful functions from string.h – `strcmp, strcat`

Read about these functions on http://www.cplusplus.com

ALTERNATIVE WAY TO DEFINE STRINGS

Strings can be defined in two ways

`char str[ /*size*/ ];` - allows changing the string.

`char* str;` - cannot change the string after it has been assigned once.

The second way uses pointers, you will learn more about pointers later ☺