CSD101: Introduction to computing and programming (ICP)

Altering execution within a loop: break, continue

break

- The break statement is used inside the body of a loop to immediately exit the loop.
- Control passes to the statement immediately after the loop.

continue

- The continue statement is used in the body of a loop when the statements following the continue statement must not be executed.
- So, control passes straight to the end of the loop. In a for this means control will pass to the increment expression in the for loop. In a while loop control will immediately pass to the <condition>.

Arguments in functions I

When a function is defined it can have 0 or more formal arguments. A formal argument is a type and a variable.

Example 4

```
void printArray(int arr[], int start, int len){
/*prints array arr starting at position start
  surrounded by square brackets [...]*/
  printf("[");
  for(int i=start;i<len;i+=1)
    printf("%d,",arr[i]);
  printf("\b]");
  return;
}</pre>
```

Arguments in functions II

 When a function is called it is given actual arguments that must be consistent with the formal arguments.
 For example, if seq is an array of size 6 then printArray(seq, 1, 6) is a function call.

Argument passing by value

- When a function is called each actual argument is evaluated to yield a value and a copy of this value is bound to the corresponding formal argument. Then the body of the function is executed. This way of passing arguments to functions is called **pass-by-value**. C passes all arguments by value.
- Any change to the formal variable inside the function does not affect its value outside the function since inside it is operating on a copy. In particular it does not affect its value in the function from where it was called.
- To make changes that are made inside the function visible outside it an address has to be passed. This happens by default when we pass an array. And that is the reason an array passed as an argument reflects changes made inside the function. This will become clearer after we discuss pointers in more detail.