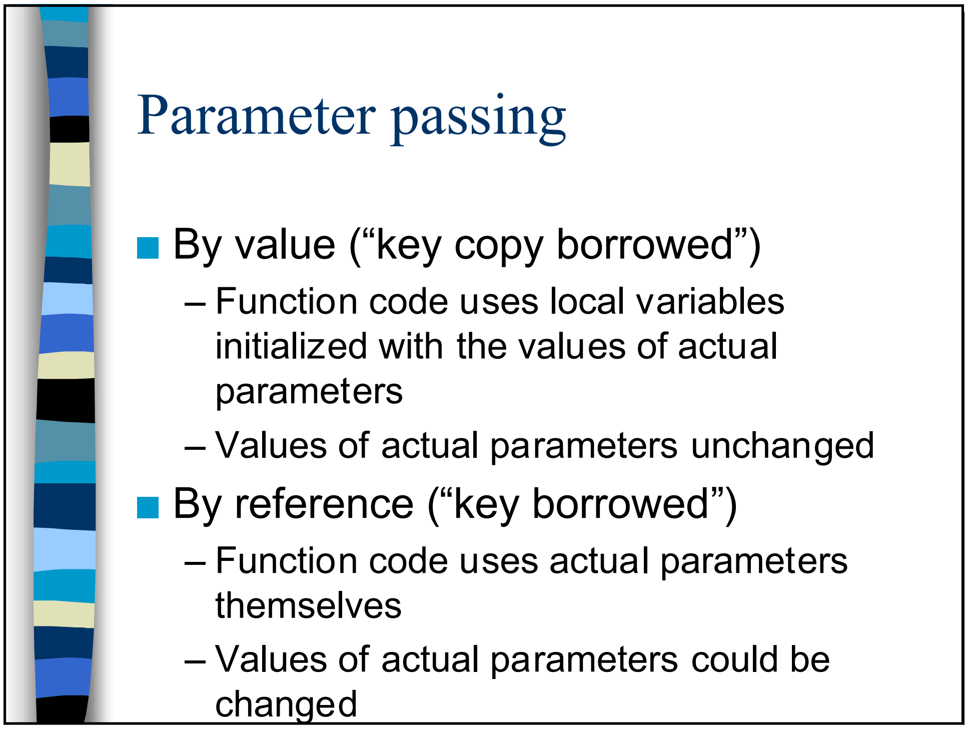
IS12 - Introduction to Programming Lecture 19: Functions and Arrays

Peter Brusilovsky

http://www2.sis.pitt.edu/~peterb/0012-051/

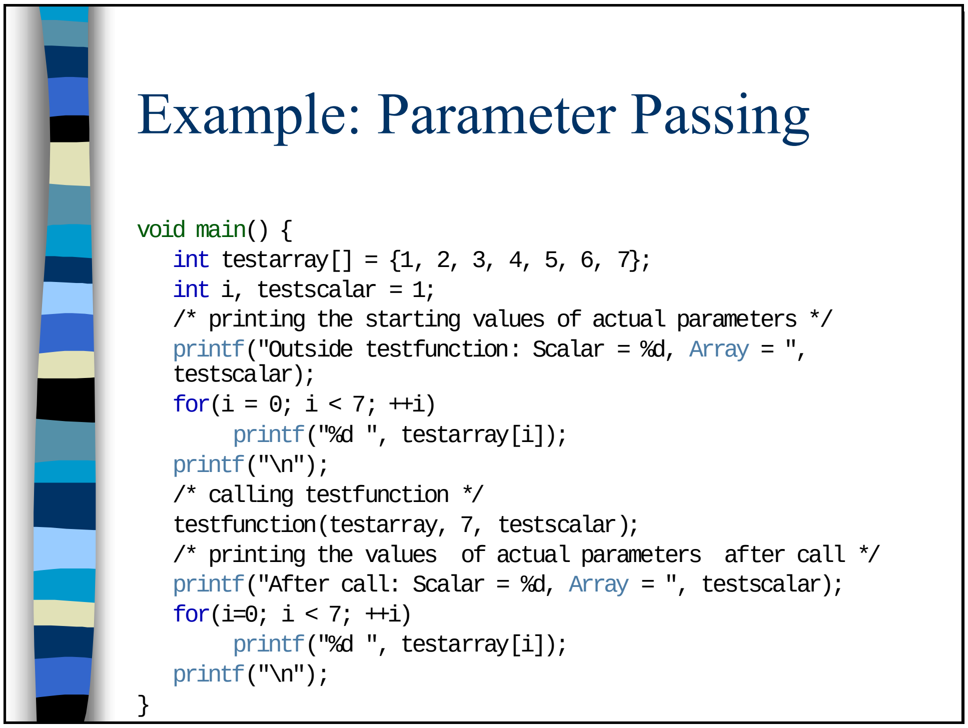
Outline

- Parameter passing by value and reference
  - Example: parameter passing
  - Example: read, sum, max
- Character arrays
- Strings, its initializing and printing
  - Example: Palindrome
  - Example: String to int



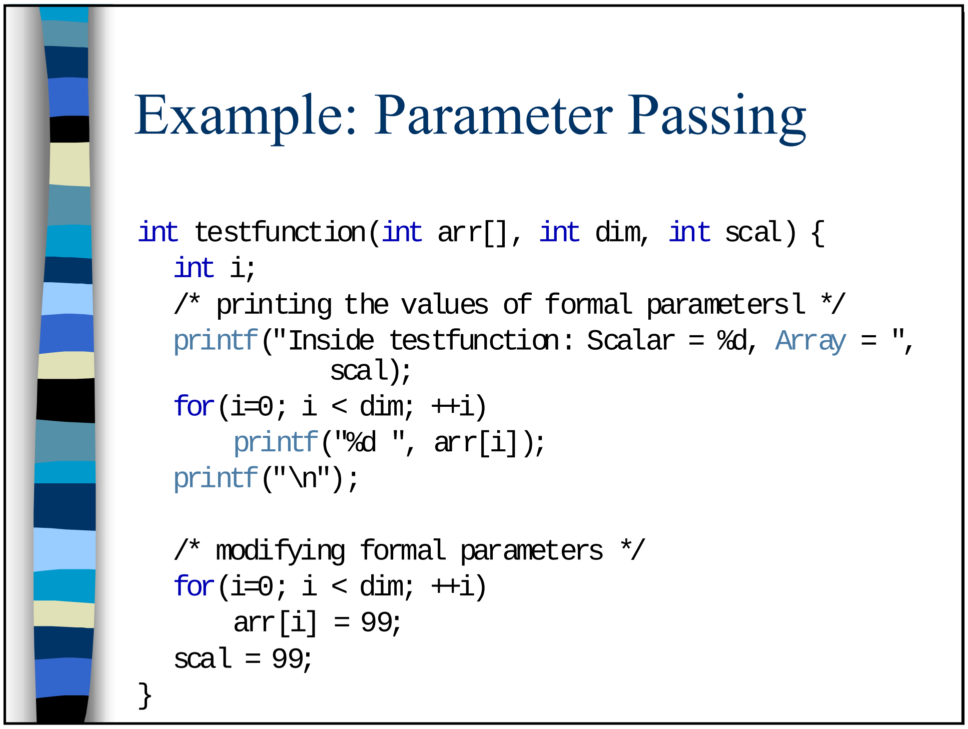
Parameter passing

- By value (“key copy borrowed”)
  - Function code uses local variables initialized with the values of actual parameters
  - Values of actual parameters unchanged
- By reference (“key borrowed”)
  - Function code uses actual parameters themselves
  - Values of actual parameters could be changed



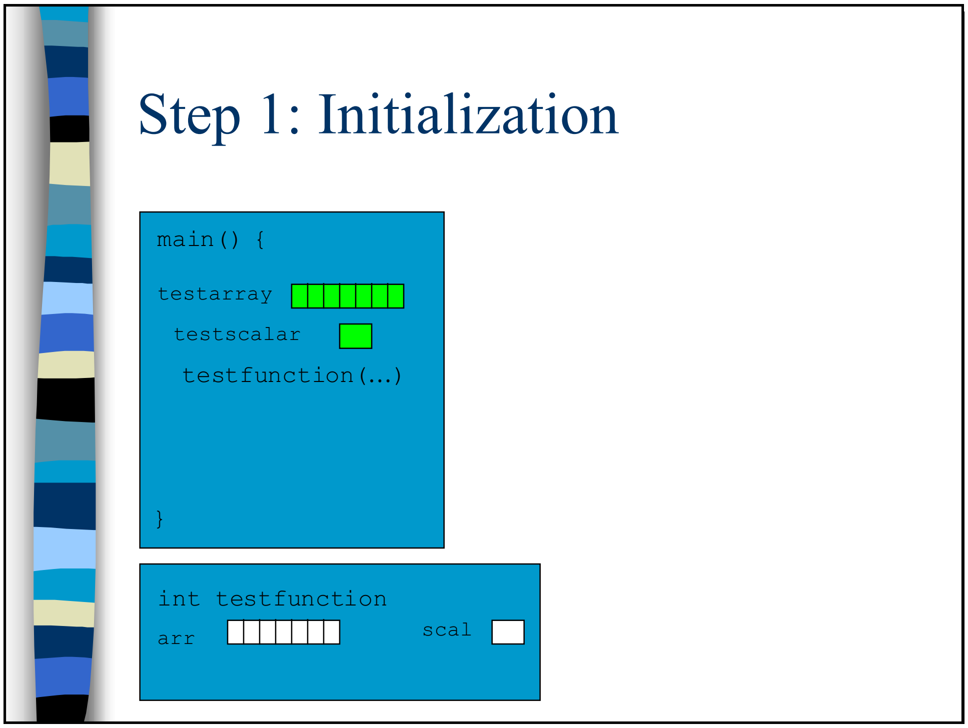
Example: Parameter Passing

```
void main() {
   int testarray[] = {1, 2, 3, 4, 5, 6, 7};
   int i, testscalar = 1;
   /* printing the starting values of actual parameters */
   printf("Outside testfunction: Scalar = %d, Array = ",
   testscalar);
   for(i = 0; i < 7; ++i)
        printf("%d ", testarray[i]);
   printf("\n");
   /* calling testfunction */
   testfunction(testarray, 7, testscalar);
   /* printing the values  of actual parameters  after call */
   printf("After call: Scalar = %d, Array = ", testscalar);
   for(i=0; i < 7; ++i)
        printf("%d ", testarray[i]);
   printf("\n");
}
```



Example: Parameter Passing

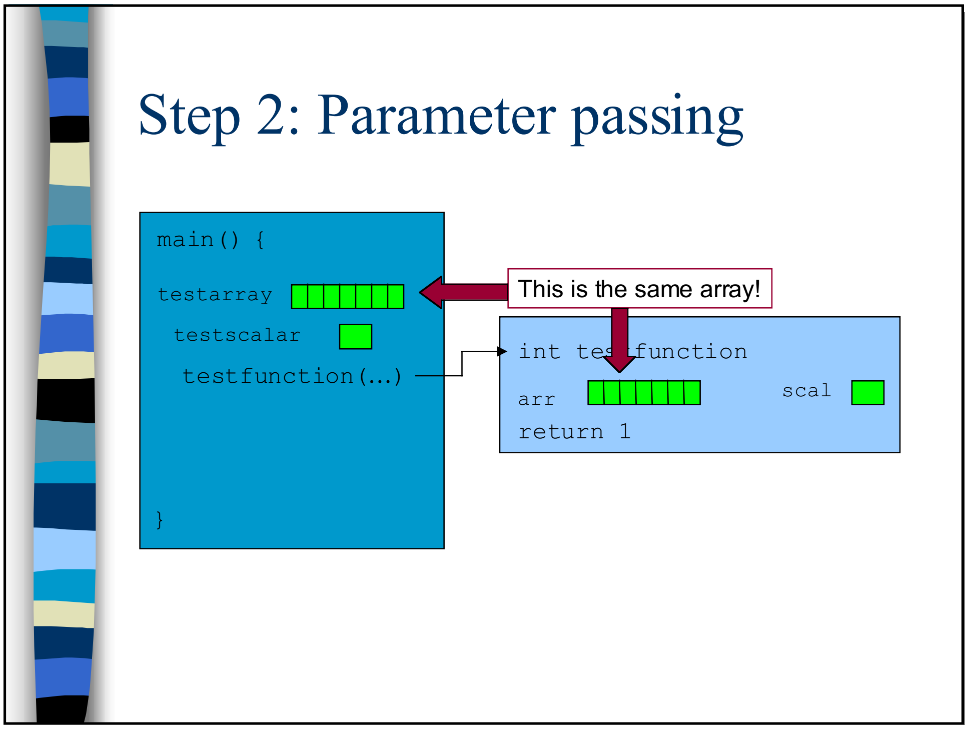
```
int testfunction(int arr[], int dim, int scal) {
  int i;
  /* printing the values of formal parametersl */
  printf("Inside testfunction: Scalar = %d, Array = ",
            scal);
  for(i=0; i < dim; ++i)
      printf("%d ", arr[i]);
  printf("\n");

  /* modifying formal parameters */
  for(i=0; i < dim; ++i)
      arr[i] = 99;
  scal = 99;
}
```



Step 1: Initialization

```
main() {

testarray  🟩🟩🟩🟩🟩🟩🟩

  testscalar   🟩

   testfunction(...)



}
```

```
int testfunction
arr  □□□□□□□          scal  □
```

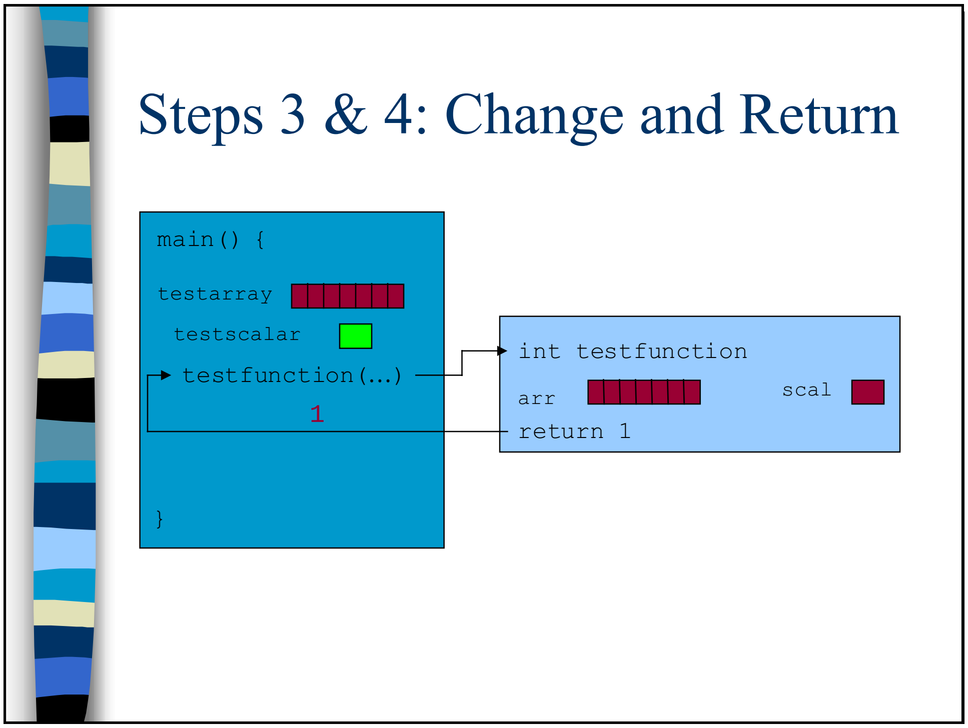


Step 2: Parameter passing

```
main() {
testarray
testscalar
testfunction(...)
}
```

This is the same array!

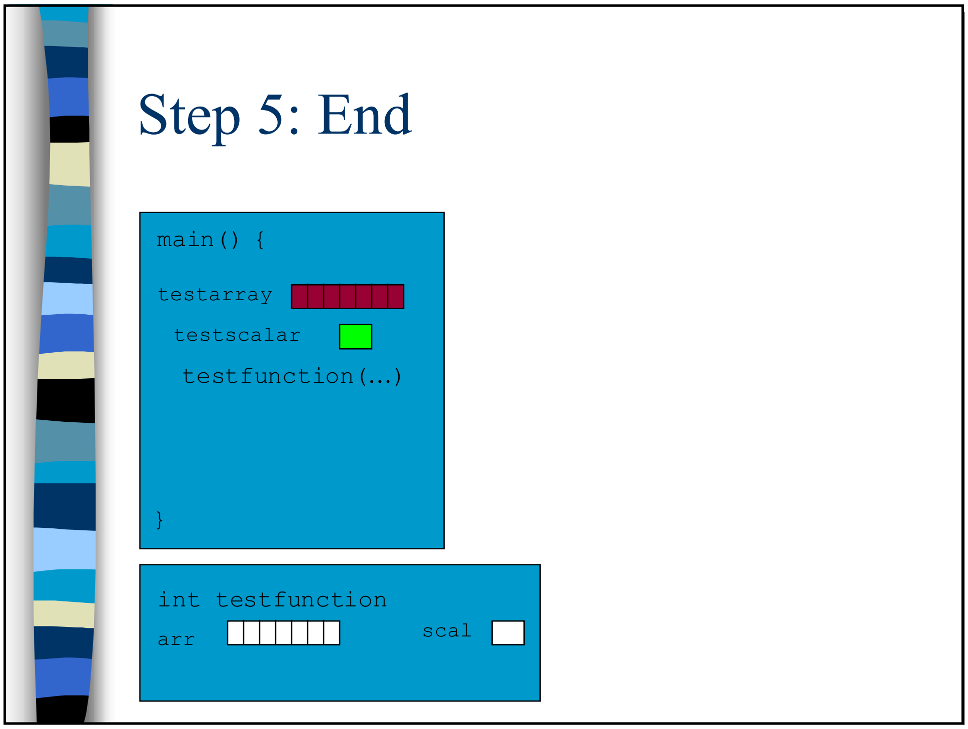
```
int testfunction
arr
scal
return 1
```



Steps 3 & 4: Change and Return

```
main() {
testarray
testscalar
testfunction(...)
1
}
```

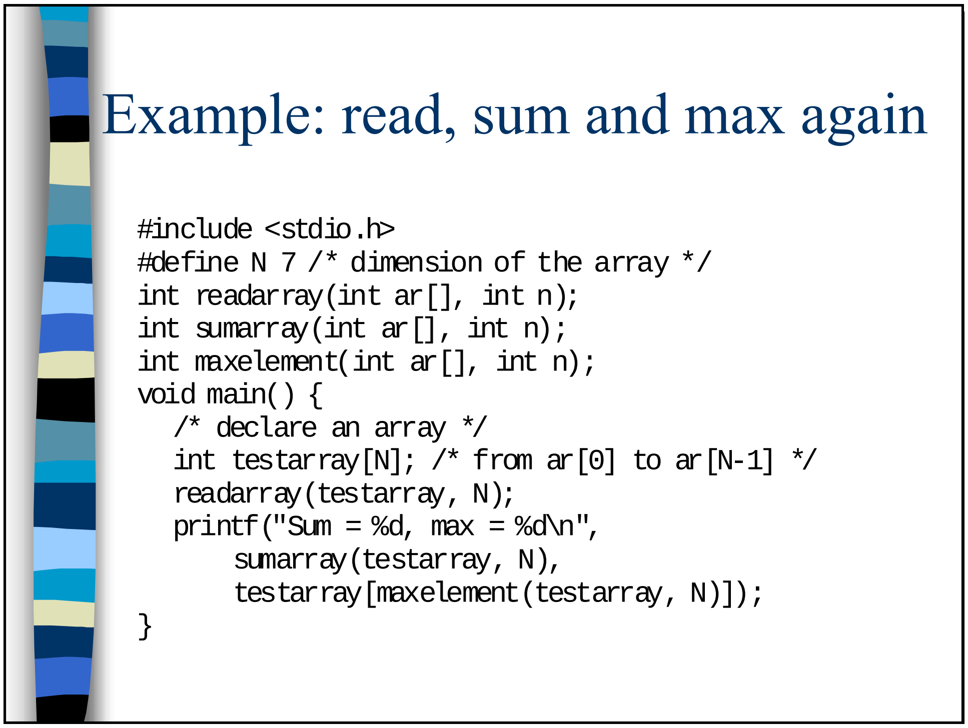
```
int testfunction
arr
scal
return 1
```



Step 5: End

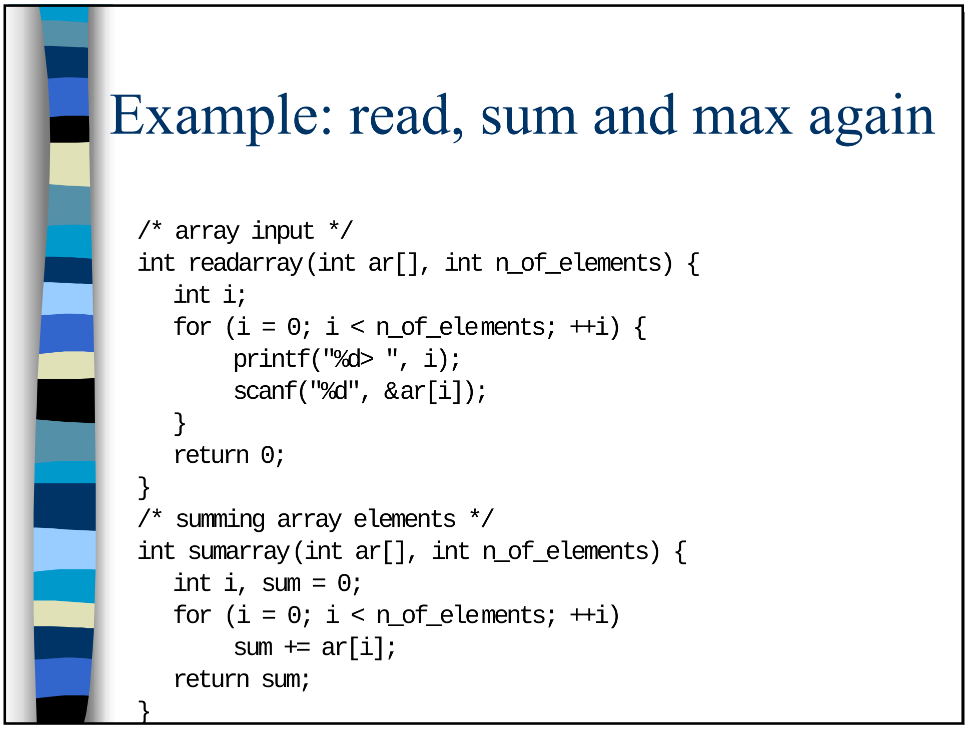
```
main() {

testarray 🟥🟥🟥🟥🟥🟥🟥

  testscalar 🟩

   testfunction(...)

}
```

```
int testfunction
arr □□□□□□□          scal □
```



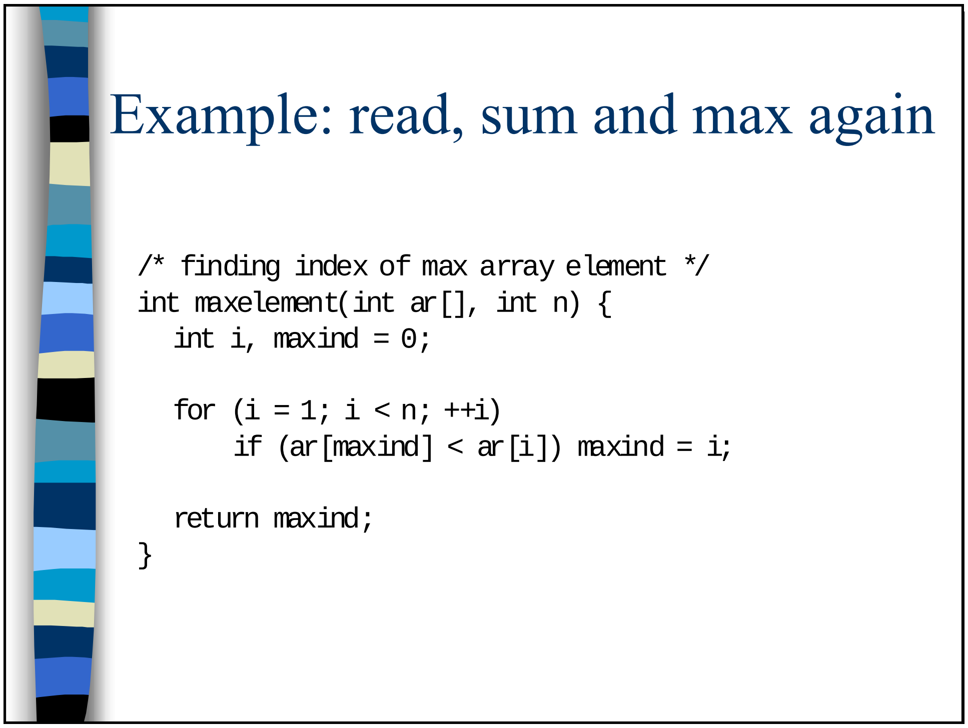
Example: read, sum and max again

```
#include <stdio.h>
#define N 7 /* dimension of the array */
int readarray(int ar[], int n);
int sumarray(int ar[], int n);
int maxelement(int ar[], int n);
void main() {
   /* declare an array */
   int testarray[N]; /* from ar[0] to ar[N-1] */
   readarray(testarray, N);
   printf("Sum = %d, max = %d\n",
        sumarray(testarray, N),
        testarray[maxelement(testarray, N)]);
}
```



Example: read, sum and max again

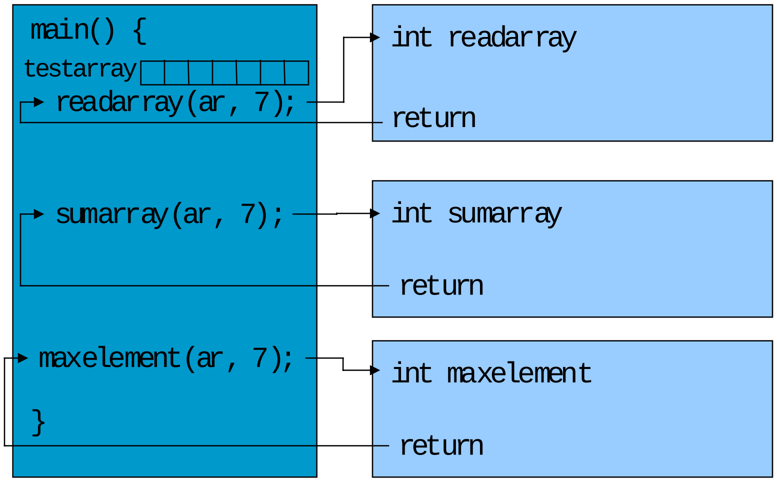
```
/* array input */
int readarray(int ar[], int n_of_elements) {
   int i;
   for (i = 0; i < n_of_elements; ++i) {
        printf("%d> ", i);
        scanf("%d", &ar[i]);
   }
   return 0;
}
/* summing array elements */
int sumarray(int ar[], int n_of_elements) {
   int i, sum = 0;
   for (i = 0; i < n_of_elements; ++i)
        sum += ar[i];
   return sum;
}
```



Example: read, sum and max again

```
/* finding index of max array element */
int maxelement(int ar[], int n) {
   int i, maxind = 0;

   for (i = 1; i < n; ++i)
        if (ar[maxind] < ar[i]) maxind = i;

   return maxind;
}
```

ReadSumMax: Control flow



main() {

testarray

readarray(ar, 7);

sumarray(ar, 7);

maxelement(ar, 7);

}

int readarray

return

int sumarray

return

int maxelement

return

Arrays of Characters

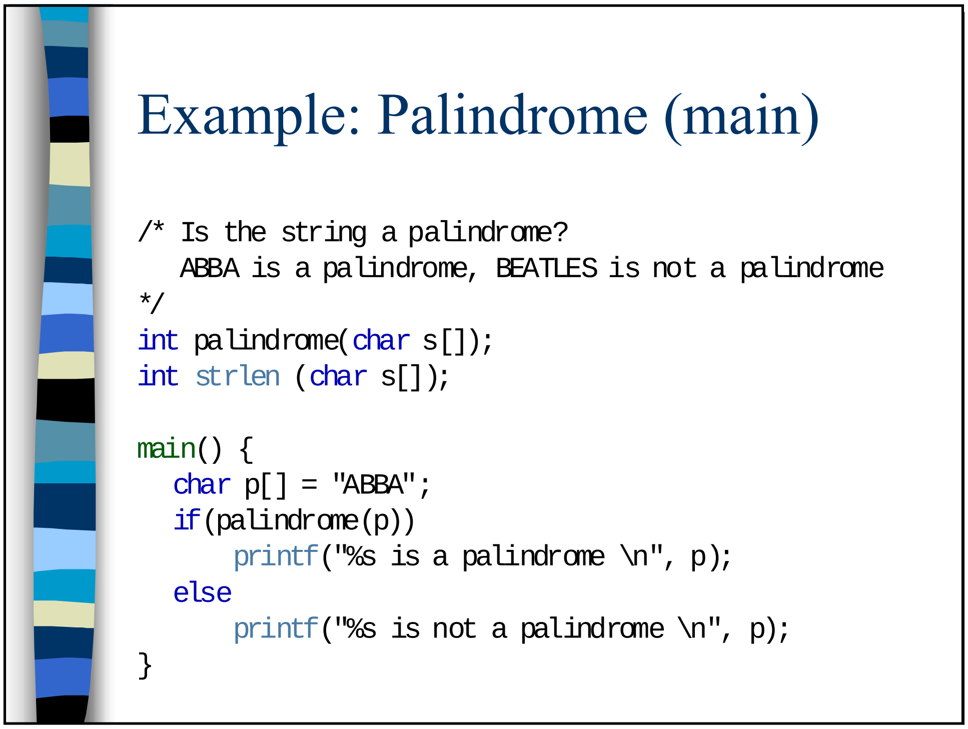
```
char str[10]; /* non-initialized */
char p1[] = { 'a', 'b' , 'c', '\0'};
```

- String: character array with `'\0'` at the end
- Strings can use special literal constants

```
char p1[] = "abc"; /* special way */
char mystring[] = "course";
```

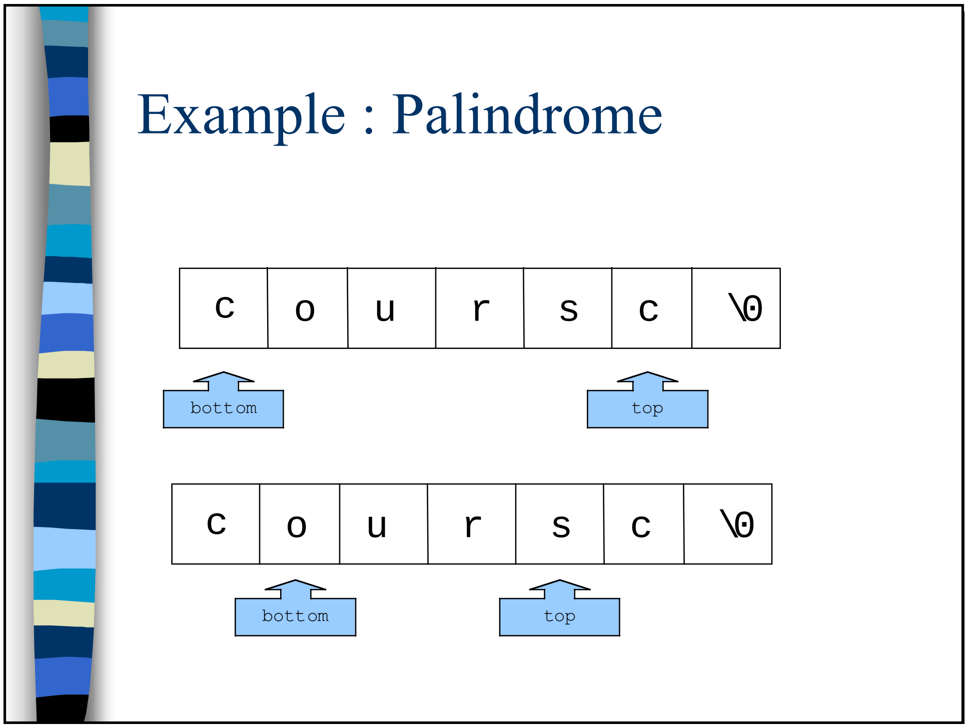
c	o	u	r	s	e	\0

```
printf("%s\n", mystring);
```



Example: Palindrome (main)

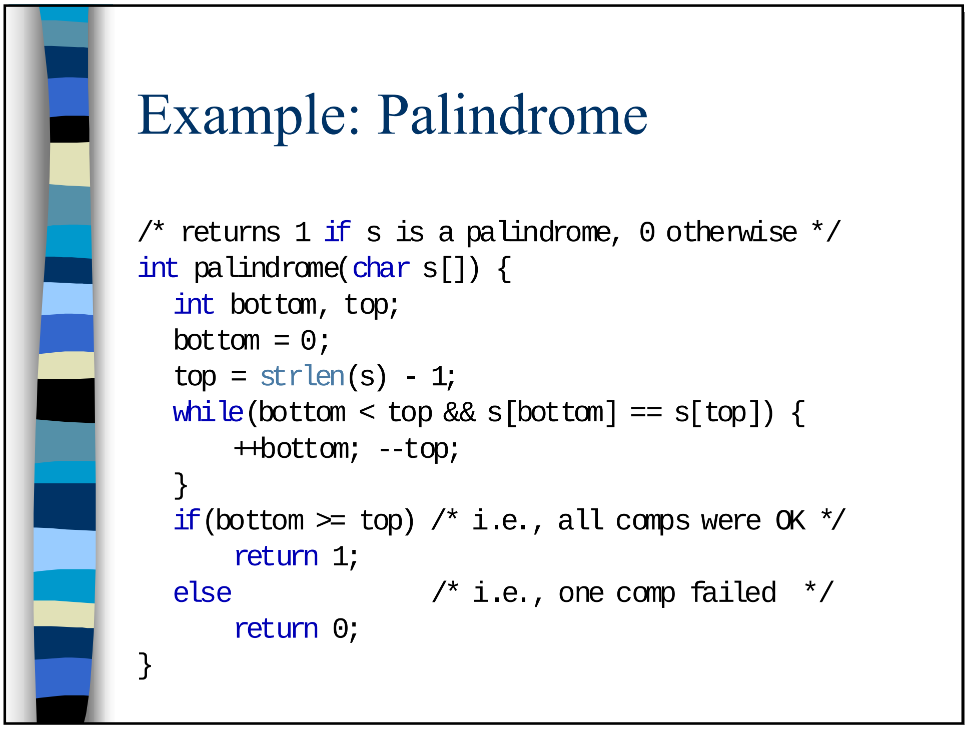
```
/* Is the string a palindrome?
   ABBA is a palindrome, BEATLES is not a palindrome
*/
int palindrome(char s[]);
int strlen (char s[]);

main() {
   char p[] = "ABBA";
   if(palindrome(p))
        printf("%s is a palindrome \n", p);
   else
        printf("%s is not a palindrome \n", p);
}
```



Example : Palindrome

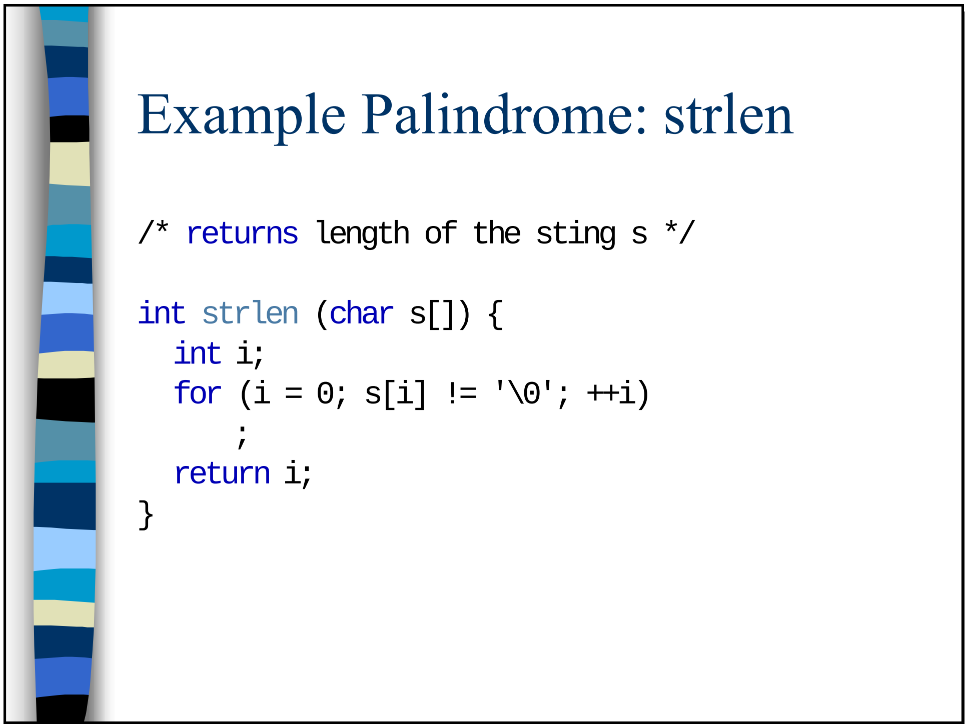
c	o	u	r	s	c	\0
bottom					top	

c	o	u	r	s	c	\0
	bottom			top		



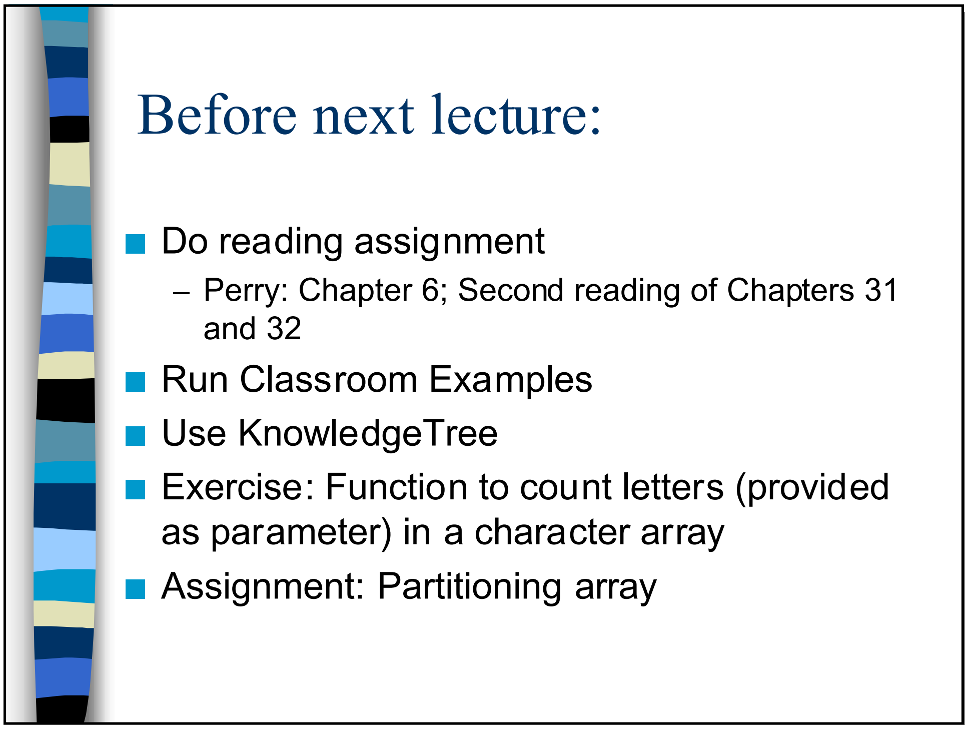
Example: Palindrome

```
/* returns 1 if s is a palindrome, 0 otherwise */
int palindrome(char s[]) {
   int bottom, top;
   bottom = 0;
   top = strlen(s) - 1;
   while(bottom < top && s[bottom] == s[top]) {
        ++bottom; --top;
   }
   if(bottom >= top) /* i.e., all comps were OK */
        return 1;
   else              /* i.e., one comp failed  */
        return 0;
}
```



Example Palindrome: strlen

```
/* returns length of the sting s */

int strlen (char s[]) {
   int i;
   for (i = 0; s[i] != '\0'; ++i)
        ;
   return i;
}
```



Before next lecture:

- Do reading assignment
  - Perry: Chapter 6; Second reading of Chapters 31 and 32
- Run Classroom Examples
- Use KnowledgeTree
- Exercise: Function to count letters (provided as parameter) in a character array
- Assignment: Partitioning array