Lesson 5-2

Arrays 2

In this lesson, you will learn how:

- How to initialize arrays
- How to search an array
- How to sort an array

Important Notes:

- **You should type in all the programs in this handout, and run them more than once with different data**
- **You should read, and understand everything in this handout, the material in it forms the basis of the quizzes**
- **If you don't understand something; ask me to explain.**

Program (states1.c)

```
/*
  Program "states1.c"
  Written by: Joe Dorward
  Date: 06/19/00

  This program uses an array of structures to store the capitals, and
  names of states.
  It then prints the name, and capital of each state
*/

#include <stdio.h>
#include <string.h>

void main(void)
{
int state_number = 0;   // used as loop counter

struct a_state_record   // define the structure
{
  char state_name_field[15];
  char capital_name_field[15];
};

struct a_state_record array_of_states[55] = {"Alabama", "Montgomery",
                                             "Alaska", "Juneau",
                                             "Arizona", "Phoenix",
                                             "Arkansas", "Little Rock",
                                             "California", "Sacramento",
                                             "Colorado", "Denver"};

  // ============================================================

  while( strcmp(array_of_states[state_number].state_name_field,"") != 0)
  {
    printf("The capital ");
    printf("of: %s ",array_of_states[state_number].state_name_field);
    printf("is: %s \n",array_of_states[state_number].capital_name_field);

    state_number++;
  }

  printf("\n\n ** Exiting Program ** \n\n");
}
```

Program (states2.c)

```
/*
  Program "states2.c"
  Written by: Joe Dorward
  Date: 06/19/00

  This program uses an array of structures to store the capitals, and
  names of states.
  It then prompts the user to enter the capital of each state, and
  checks their answer
*/
#include <stdio.h>
#include <string.h>

void main(void)
{
int state_number = 0;   // used as loop counter

char temp_capitol_name[15] = "";

struct a_state_record   // define the structure
{
  char state_name_field[15];
  char capital_name_field[15];
};

struct a_state_record array_of_states[55] = {"Alabama", "Montgomery",
                                             "Alaska", "Juneau",
                                             "Arizona", "Phoenix",
                                             "Arkansas", "Little Rock",
                                             "California", "Sacramento",
                                             "Colorado", "Denver"};

  // ================================================================================

  while( strcmp(array_of_states[state_number].state_name_field,"") != 0)
  {
    printf("\nWhat is the capital ");
    printf("of: %s ",array_of_states[state_number].state_name_field);
    gets(temp_capitol_name);

    if ( strcmp(temp_capitol_name,array_of_states[state_number].capital_name_field) == 0)
    {
      printf("Right! \n");
      printf("Good job. \n");
    }
    else
    {
      printf("Wrong! \n");
      printf("It\'s: %s \n",array_of_states[state_number].capital_name_field);
      printf("You dummy. \n");
    }
    state_number++;
  }
  printf("\n\n ** Exiting Program ** \n\n");
}
```

Searching an array

Ask for data value to search for

Start loop
    Use loop counter to inspect first element for data value
    (if found, do something about it)
    (else, increment loop counter)
End loop

array_of_states[55]

	state_name_field	capitol_name_field
[0]	Alabama	Montgomery
[1]	Alaska	Juneau
[2]	Arizona	Phoenix
[3]	Arkansas	Little Rock
[4]	California	Sacramento
[5]	Colorado	Denver

If "California" entered for the state name

loop counter = 0	state name = "Alabama"	not found ⇨ increment loop counter
loop counter = 1	state name = "Alaska"	not found ⇨ increment loop counter
loop counter = 2	state name = "Arizona"	not found ⇨ increment loop counter
loop counter = 3	state name = "Arkansas"	not found ⇨ increment loop counter
loop counter = 4	state name = "California"	found ⇨ print other information ⇨ stop searching

Program (states3.c)

```
/*
  Program "states3.c"
  Written by: Joe Dorward
  Date: 06/19/00

  This program uses an array of structures to store the capitals, and
  names of states.

  It then prompts the user for a state name, and searches for that state name.
  If the state name is found, the program prints out the capitol name.
  If the state name is not found, the program prints a not-found message.
*/

#include <stdio.h>
#include <string.h>

void main(void)
{
int state_number = 0,   // used as loop counter
    flag = 0;

char temp_state_name[15];

struct a_state_record   // define the structure
{
  char state_name_field[15];
  char capital_name_field[15];
};

struct a_state_record array_of_states[55] = {"Alabama", "Montgomery",
                                             "Alaska", "Juneau",
                                             "Arizona", "Phoenix",
                                             "Arkansas", "Little Rock",
                                             "California", "Sacramento",
                                             "Colorado", "Denver"};

  // ============================================================================

  printf(" The state capitol database \n\n");
  printf(" Please enter a state name : ");
  gets(temp_state_name);


  while( ( (strcmp(array_of_states[state_number].state_name_field,"") != 0) ) && (flag == 0) )
  {
    if ( strcmp(array_of_states[state_number].state_name_field, temp_state_name) == 0)
    {
      printf(" The capital ");
      printf("of: %s ",array_of_states[state_number].state_name_field);
      printf("is: %s \n",array_of_states[state_number].capital_name_field);

      flag = 1;   // state found
    }
    else
    {
      state_number++;
    }
  }

  if (flag == 0)   // state not found
  {
     printf("\n The state: %s, is not in the database. \n",temp_state_name);
  }
  printf("\n\n ** Exiting Program ** \n\n");
}
```

Program (`sort1.c`)

```
/*
  Program "sort1.c"
  Written by: Joe Dorward
  Date: 06/19/00

  This program declares an array of characters, initializes it
  then sorts it.
*/

#include <stdio.h>

void main(void)
{
int loop_counter_one = 0,
    loop_counter_two = 0,
    element_number = 0;

char character_array[15] = {'H', 'F', 'A', 'G', 'Z', 'R', 'U', 'C', 'I', 'L'};

  printf(" Print the unsorted array: \n");

  for (loop_counter_one = 0; loop_counter_one < 10; loop_counter_one++)
  {
    printf(" %c ",character_array[loop_counter_one]);
  }

  // ========================================================================
  // sort array

  for (loop_counter_one = 0; loop_counter_one < 10; loop_counter_one++)
  {
    for (loop_counter_two = (loop_counter_one + 1); loop_counter_two < 10; loop_counter_two++)
    {
      if ( character_array[loop_counter_two] < character_array[loop_counter_one] )   // swap
      {
        character_array[14] = character_array[loop_counter_one];

        character_array[loop_counter_one] = character_array[loop_counter_two];

        character_array[loop_counter_two] = character_array[14];
      }
    }
  }
  // ========================================================================

  printf("\n\n ---------------------------------- \n");
  printf(" Print the sorted array: \n");

  for (loop_counter_one = 0; loop_counter_one < 10; loop_counter_one++)
  {
    printf(" %c ",character_array[loop_counter_one]);
  }

  printf("\n\n");
}
```