Computer Programming using C
Lecture 7: Arrays and Strings

Dr Julian Miller
Room P/M/101
Ext 2383; E-mail: jfm@ohm
http://www.elec.york.ac.uk/intsys/users/jfm7/

Introduction

• Revision (in the lab last week)
• Answers to exercises
• In the lab this week
Arrays

- Variables which have space for more than one value, for example

```c
double velocity[2];
int matrix[10][20];
char words[20][30];
```

- Arrays are assigned values by specifying the index values and assigning a value

```c
velocity[0] = 10.5;
velocity[1] = -2.4;
for (i=0; i < 10; i++)
{
    for (j=0; j < 20; j++)
    {
        matrix[i][j]=i"j + 3;
    }
}
```

Exercise 6.2

- You were given a program that asks you to enter the heights of up to 10 people and the program calculated the average height. Here is the simplified version

```c
#include <stdio.h>
double mean(double values[], int num_values);
int main(void)
{
    int i, num_values;
    double mean_height, data_values[10];
    num_values = 0;
    for (i = 0; i < 10; i++)
    {
        printf("Please enter the height for person %d (or 0 if done): ", i + 1);
        scanf("%lf", &data_values[i]);
        if (data_values[1] != 0)
            num_values++;
        else
            break;
    }    
    mean_height = mean(data_values, num_values);
    printf("The mean height is %lf", mean_height);
    return 0;
}
Mean function

```c
double mean(double values[], int num_values)
{
    double mean;
    int i;
    mean = 0;
    for (i = 0; i < num_values; i++)
        mean = mean + values[i];
    if (num_values > 0)
        mean = mean/num_values;
    return mean;
}
```

- The exercise asked you to write a function that could calculate the standard deviation of the data.

Standard deviation

\[ \sigma = \sqrt{x^2 - (\bar{x})^2} \]

```c
double deviation (double values[], int num_values, double themean)
{
    int i;
    double squared_values[10];
    for (i = 0; i < num_values; i++)
        squared_values[i] = values[i]*values[i];
    return sqrt(mean(squared_values, num_values) - themean*themean);
}
```
Initialising arrays

- Arrays can be initialised as in the examples below

```c
int data_table[5][3] = {{1, 2, 3}, {2, 3, 4}, {3, 4, 5}, {4, 5, 6}, {5, 6, 7}};

int randperm[6]={0,1,2,3,4,5};
double x_coords[100]={0.0,0.1,0.2,0.3,0.4,0.5}
unsigned positive_ints[20]={0};
```

Copying arrays: there ain’t a fast way!

- If you want to copy the contents of one array into another you have to laboriously copy each element of the first array into the other
- e.g.

```c
int i;
int my_first_array[5]={1,2,3,5,7};
int my_second_array[5];

for (i=0;i<6;i++)
    my_second_array[i]=my_first_array[i];
```
Communicating information to and from functions

- You have found that you can pass information to functions using function arguments and get information from functions via the return statement of the function. Well, there is another way using arrays. Let's improve the program in exercise 6.2

```c
#include <stdio.h>

double mean(double values[], int num_values);
double deviation(double values[], int num_values, double the mean);

int main(void)
{
    int i, num_values;
double mean_height, data_values[10];
double standard deviation;
    num_values = 0;
    for (i = 0; i < 10; i++)
    {
        printf("Please enter the height:");
        printf(" for person %d (or 0 if done): ", i + 1);
        scanf("%f", &data_values[i]);
        if (data_values[i] == 0)
            num_values--;
    
    
    mean_height = mean(data_values, num_values);
    standard deviation = deviation(data_values, num_values, mean_height);
    printf("The mean height is %f in", mean_height);
    printf(" The standard deviation is %f in", standard deviation);
    return 0;
}
```
Solution to exercise 6.2 new version

```c
#include <stdio.h>

int get_data_values(double values[]), mean(double values[], int num_values),
  mean_height(double values[], int num_values),
  standard_deviation(double values[], int num_values)
{
    int num_values = 0;
    double mean_height, data_values[10];
    double standard_deviation;

    mean_height = mean(values, num_values);
    standard_deviation = standard_deviation(values, num_values, mean_height);
    printf("The mean height is %.2f", mean_height);
    printf("The standard deviation is %.2f", standard_deviation);
    return 0;
}

int get_data_values(double values[])
{
  int i, num_values = 0;
  for (i = 0; i < 10; i++)
  {
    printf("Please enter the height: ");
    printf("for person %d (or 0 if done): ", i + 1);
    scanf("%f", data_values[]);
    if (data_values[i] == 0)
      num_values++;
    else
      break;
  }
  return num_values;
}
```

Exercise 6.3

- Use the program lab6a.c and the table of distances between cities to create a program that can
  - Tell the user the distances between cities
  - Validate the indexes used for the cities.

<table>
<thead>
<tr>
<th></th>
<th>Cape Town</th>
<th>Hong Kong</th>
<th>London</th>
<th>New York</th>
<th>Rio de Janeiro</th>
<th>Tokyo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cape Town</td>
<td>0</td>
<td>7,375</td>
<td>6,012</td>
<td>7,764</td>
<td>3,773</td>
<td>9,156</td>
</tr>
<tr>
<td>Hong Kong</td>
<td>7,375</td>
<td>0</td>
<td>5,982</td>
<td>8,054</td>
<td>11,021</td>
<td>1,794</td>
</tr>
<tr>
<td>London</td>
<td>6,012</td>
<td>5,982</td>
<td>0</td>
<td>3,458</td>
<td>5,766</td>
<td>5,940</td>
</tr>
<tr>
<td>New York</td>
<td>7,764</td>
<td>8,054</td>
<td>3,458</td>
<td>0</td>
<td>4,817</td>
<td>6,740</td>
</tr>
<tr>
<td>Rio de Janeiro</td>
<td>3,773</td>
<td>11,021</td>
<td>5,766</td>
<td>4,817</td>
<td>0</td>
<td>11,533</td>
</tr>
<tr>
<td>Tokyo</td>
<td>9,156</td>
<td>1,794</td>
<td>5,940</td>
<td>6,740</td>
<td>11,533</td>
<td>0</td>
</tr>
</tbody>
</table>
First steps

1. We have to get the distance data given in the lab script into an array

```c
int distances[6][6] = {
    {0, 7375, 6102, 7764, 3773, 9156},
    {7375, 0, 5982, 8054, 13021, 1781},
    {6102, 5982, 0, 3458, 5766, 5940},
    {7764, 8054, 3458, 0, 4827, 6740},
    {3773, 13021, 5766, 4827, 0, 11933},
    {9156, 1794, 5940, 6740, 11933, 0}
};
```

2. We need a function to check that the indexes are in the correct range (0-5)

```c
int index_invalid(int index) {
    return ((index<1) || (index>6));
}
```

Next steps

• We want a function that will get the indexes from the user and make sure that they are valid. It would be nice if it handled both the start and end index

```c
int get_valid_index(int start) {
    int city;
    do {
        printf("Enter the index of the city ");
        if (start) printf("where you want to start : ");
        else printf("where you want to end : ");
        scanf("%d", &city);
        if (index_invalid(city))
            printf("Invalid index entered\n");
        else
            break;
    } while (1);
    return city;
}
```
Finally

- We want to call the functions that gets the start and end indexes,
- Change them to valid locations in the distances array
- Tell the user what the distance between the cities is.

```c
int main(void)
{
    int start_city, end_city;
    int distances[5][5] = {
        {0, 7375, 6102, 7764, 3773, 9356},
        {7375, 0, 7375, 8504, 11021, 1784},
        {6102, 7375, 0, 3198, 5766, 3940},
        {7764, 8504, 3198, 0, 4617, 8740},
        {3773, 11021, 5766, 4617, 0, 11021},
        {9356, 1784, 3940, 8740, 11021, 0}
    };
    display_options();
    start_city = get_valid_index(1) - 1;
    end_city = get_valid_index(0) - 1;
    printf("The distance between these cities is %d\n", distances[start_city][end_city]);
    return 0;
}
```

Strings

- Strings are arrays of characters. The last element in a string is a special symbol ‘\0’. This is referred to as a null terminator. Below are a few statements illustrating aspects of string handling

```c
char some_string[8], another_string[21];
char some_string[8] = "Hello";
if (another_string[0] == '\0')
    printf("The string is empty\n");
else
    printf("The string is %s", another_string);
printf("Enter a string (< 21 characters): ");
scanf("%20s", another_string);
```
Exercise 6.5: Finding the length of a string

```c
int string_length(char string[]) {
    int length = -1;
    do {
        length++;
    } while (string[length] != '\0');
    return length;
}
```

Exercise 6.5: main

```c
#include <stdio.h>
int string_length(char string[]);
int main(void) {
    char string_data[20];
    printf("Enter a word: ");
    scanf("%s", string_data);
    printf("The word you entered is: %s\n", string_data);
    printf("The length of the string is %d\n", string_length(string_data));
    return 0;
}
```
Exercise 6.6: Reversing a string

- Write a function that will reverse a string

```c
void reverse_string(char string[], char revstring[])
{
    int i, j = 0, length;
    length = string_length(string);
    for (i = length - 1; i >= 0; i--)
    {
        revstring[j] = string[i];
        j++;
    }
    revstring[length] = '\0';
}
```

Exercise 6.7: Using sprintf

- Convert integer entered by user into equivalent string of characters and determine the length of the resulting string using `sprintf` function and your own string length function.

```c
#include <stdio.h>
int string_length(char string[]);
int main(void)
{
    int integer;
    char integer_string[20];
    printf("Enter an integer: ");
    scanf("%d", &integer);
    printf("The integer you entered is: \d\n", integer);
    sprintf(integer_string, "%d", integer);
    printf("The length of the integer string is \d\n", string_length(integer_string));
    printf("sprintf says the length of the integer string is \d\n", sprintf_length);
    return 0;
}
```
String handling functions

• In `string.h` there are many useful string handling functions. Here are three that are very useful:
  
  `strcpy(string1, string2);`
  
  – Copy `string2` into `string1`

  `strcat(string1, string2);`
  
  – Concatenate (add to the end of) `string2` onto `string1`

  `strcmp(string1, string2);`
  
  – If `string1` is alphabetically before `string2` `strcmp` returns a negative number. If `string1` equals `string2`, it returns zero. If `string2` is alphabetically before `string1` then `strcmp` returns a positive number.

In the lab in remaining 3 weeks

• Software project
• Software Design
• Assignment