CSE1301
Computer Programming
Lecture 18
Arrays (Part 1)

1

Topics

- Arrays
  - Declaration
  - Initialization
  - Input/Output
  - Passing arrays to functions

2

Arrays

- A group of contiguous memory locations used to store a series of related values
- The array name is a pointer to the first element
- All values have the same type
- Individual elements of an array are accessed via an integer index: `array[index]`
- Element indices start at **0**: `array[0]` is the first element

3

Initialization

- Arrays may be initialized with a list of suitable values
- No need to specify the number of elements for a 1D (1-dimensional) array

4

Example: MonthlyRainfall

Problem: using *Rainfall Table*
- input month
- output mean rainfall for that month

month	mean rainfall (in mm)
0	30
1	40
2	45
3	95
4	130
5	220
6	210
7	185
8	135
9	80
10	40
11	45

Rainfall Table

5

Example (cont): MonthlyRainfall (v.1)

```
#include <stdio.h>

int main()
{
  int    month;
  int    table[12] = { 30, 40, 45, 95, 130, 220,
                       210, 185, 135, 80, 40, 45 };

  printf("Enter month: ");
  scanf("%d", &month);

  printf("Average rainfall: %d mm.\n", table[month-1]);

  return 0;
}
```

rainfall1.c

6

Example (cont): MonthlyRainfall (v.1)

```
#include <stdio.h>

int main()
{
  int    month;
  int    table[12] = { 30, 40, 45, 95, 130, 220,
                       210, 185, 135, 80, 40, 45 };

  printf("Enter month: ");
  scanf("%d", &month);

  printf("Average rainfall: %d mm.\n", table[month-1]);

  return 0;
}
```

rainfall1.c

7

Input / Output of Arrays

- Library functions printf() and scanf() do not know about arrays

  ⇒ So we have to do I/O ourselves

8

Example: IORainfall-1

```
#include <stdio.h>
#define  NMONTHS 12

/* Store and print rainfall */

int main()
{
  int data[NMONTHS];
  int month;

  for ( month=0; month < NMONTHS; month++ )
  {
     scanf("%d", &data[month] );
  }

  ...
```

rainio1.c

9

Example (cont): IORainfall-1

```
#include <stdio.h>
#define  NMONTHS 12

/* Store and print rainfall */

int main()
{
  int data[NMONTHS];
  int month;

  for ( month=0; month < NMONTHS; month++ )
  {
     scanf("%d", &data[month] );
  }

  ...
```

rainio1.c

10

Example (cont): IORainfall-2 (v.1)

```
#include <stdio.h>
#define  NMONTHS 12
 ...
  /* Print from January to December */
  for ( month=0; month < NMONTHS; month++ )
  {
     printf( "%d ", data[month] );
  }
  printf("\n");

  /* Print from December to January */
  for ( month = NMONTHS - 1; month >= 0; month-- )
  {
    printf( "%d ", data[month] );
  }
  printf("\n");
  return 0;
}
```

rainio1.c

11

Example (cont): IORainfall-2 (v.1)

```
#include <stdio.h>
#define  NMONTHS 12
 ...
  /* Print from January to December */
  for ( month=0; month < NMONTHS; month++ )
  {
     printf( "%d ", data[month] );
  }
  printf("\n");

  /* Print from December to January */
  for ( month = NMONTHS - 1; month >= 0; month-- )
  {
    printf( "%d ", data[month] );
  }
  printf("\n");
  return 0;
}
```

rainio1.c

12

Example (cont): IORainfall-2 (v.2)

```
#include <stdio.h>
#define  NMONTHS 12
...
  /* Print from January to December */
  for ( month=0; month < NMONTHS; month++ )
  {
     printf( "%5d " , data[month] );
  }
  printf("\n");

  /* Print from December to January */
  for ( month = NMONTHS - 1; month >= 0; month-- )
  {
    printf( "%5d " , data[month] );
  }
  printf("\n");

  return 0;
}
```

rainio2.c

13

Handling Indices

- Arrays have a fixed size
- There is no built-in way of checking if the supplied index is within range
- We must check for valid indices ourselves

14

Example (cont): MonthlyRainfall (v.2)

```
#include <stdio.h>
#define MAXLEN 1024
int main()
{
  int    month;
  char   line[MAXLEN];
  char   dummy[MAXLEN];
  int table[12] = { 30, 40, 45, 95, 130, 220, 210, 185, 135, 80, 40, 45 };

  while(1)
  {
    printf("Enter month or ctrl-c to end: ");
    fgets(line, MAXLEN, stdin);
    if (sscanf(line, "%d%s", &month, dummy) != 1)   /* valid input? */
    {
      printf("Invalid input. Try again.\n");
    }
    else if (1 <= month && month <= 12)             /* input in range? */
    {
      printf("Average rainfall for month %d is %d mm.\n",month,table[month-1]);
    }
    else
    {
      printf("Month should be between 1 and 12. Try again.\n");
    }
  }
  return 0;
}
```

rainfall2.c

15

Example (cont): MonthlyRainfall-1 (v.3)

```
#include <stdio.h>
#define MAXLEN 1024
int rainfall(int month);
/* Main program to test rainfall() function */
int main()
{
  int    month;
  char   line[MAXLEN];
  char   dummy[MAXLEN];
  while(1)
  {
    printf("Enter month or ctrl-c to end: ");
    fgets(line, MAXLEN, stdin);
    if (sscanf(line, "%d%s", &month, dummy) != 1)
    {
      printf("Invalid input. Try again.\n");
    }
    else if (1 <= month && month <= 12)
    {
      printf("Average rainfall for month %d is %d mm.\n",month,rainfall(month-1));
    }
    else
    {
      printf("Month should be between 1 and 12. Try again.\n");
    }
  }
  return 0;
}
```

rainfall3.c

16

Example (cont): MonthlyRainfall-2 (v.3)

```
/*************************************************************\
 * NAME:
 *    int rainfall(int month)
 * DESCRIPTION:
 *    Returns the mean monthly rainfall (in millimeters)
 *    in a given month
 * PRE:
 *    The integer `month' must be between 0 and 11, where
 *    0 = January, 1 = February, etc.  Otherwise, the behaviour
 *    is undefined
 *    The local array `table' should be initialized to contain
 *    the average rainfall in a given month
 * POST:
 *    It returns an integer value corresponding to the mean
 *    rainfall (in millimeters) for the given `month'
\*************************************************************/
int rainfall ( int month )
{
  int table[12] = { 30, 40, 45, 95, 130, 220,
                    210, 185, 135, 80, 40, 45 };
  return (table[month]);
}
```

rainfall3.c

17

Example (cont): MonthlyRainfall-2 (v.3)

```
/*************************************************************\
 * NAME:
 *    int rainfall(int month)
 * DESCRIPTION:
 *    Returns the mean monthly rainfall (in millimeters)
 *    in a given month
 * PRE:
 *    The integer `month' must be between 0 and 11, where
 *    0 = January, 1 = February, etc.  Otherwise, the behaviour
 *    is undefined
 *    The local array `table' should be initialized to contain
 *    the average rainfall in a given month
 * POST:
 *    It returns an integer value corresponding to the mean
 *    rainfall (in millimeters) for the given `month'
\*************************************************************/
int rainfall ( int month )
{
  int table[12] = { 30, 40, 45, 95, 130, 220,
                    210, 185, 135, 80, 40, 45 };
  return (table[month]);
}
```

rainfall3.c

18

Passing Arrays to Functions

- The array is passed
  - as an array of unspecified size (**`int array[]`**) OR
  - as a pointer (**`int *array`**)
- Changes to the array within the function affect the "original" array

19

Example (cont): IORainfall-1 (v.3)

```
#include <stdio.h>
#define  NMONTHS 12

void loadRain ( int arrayPtr[] )
{
  int month;

  for (month=0; month < NMONTHS; month++)
  {
     scanf("%d", &arrayPtr[month]);
  }
}
```

rainio3.c

20

Example (cont): IORainfall-2 (v.3)

```
void printRain ( const int arrayPtr[] )
{
  int month;

  for (month=0; month < NMONTHS; month++)
  {
     printf("%5d", arrayPtr[month]);
  }

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

rainio3.c

21

Example (cont): IORainfall-3 (v.3)

```
#include <stdio.h>
#define  NMONTHS 12

void loadRain ( int arrayPtr[] );
void printRain ( const int arrayPtr[] );

/* Store and print rainfall */
int main()
{
  int data[NMONTHS];

  loadRain(data);
  printRain(data);

  return 0;
}
```

rainio3.c

22

Example: IORainfall -- v.3 (cont)

```
#include <stdio.h>
#define  NMONTHS 12
void loadRain ( int arrayPtr[] );
void printRain ( const int arrayPtr[] );
/* Store and print rainfall */
int main()
{
  int data[NMONTHS];
  loadRain(data);
  printRain(data);
  return 0;
}
/* Read in rainfall for each month*/
void loadRain ( int arrayPtr[] )
{
  int month;
  for (month=0; month < NMONTHS; month++)
  {  scanf("%d", &arrayPtr[month]);  }
}
/* Print rainfall for each month*/
void printRain ( const int arrayPtr[] )
{
  int month;
  for (month=0; month < NMONTHS; month++)
  {  printf("%5d", arrayPtr[month]);  }
  printf("\n");
}
```

rainio3.c

23

Reading

- King
  - Chapter 8, Chapter 12 (12.2-12.4)
- Deitel and Deitel
  - Chapter 6 (6.1-6.5)

24