Topics 3. C Basics (2). Examples

Data Types, Operators, Decision and Repetition Statements

COMP ENG 2SH4

Principles of Programming

McMaster University, 2015

Instructor: Sorina Dumitrescu

Example 1

- Read 100 floating point numbers input by the user.
- Print them in reverse order (the last input number is printed first, etc.).
- We have to store all of them.
- In C we can use an **array** to store a **sequence** of values

Arrays in C

- An **array** can be used to represent a **sequence** of items
  - Items must be of the **same type**
- To refer to an individual item use
  - **array name** and a **subscript** (or **index**) representing the position of the element in the array
  - Indexing **starts with 0**
- Before using an array you must first **define** it (declare it), i.e., specify:
  - **Name**
  - **Type** of items
  - **Size** (total number of slots) (size will remain fixed)
- When the array is defined **memory is allocated** to store all items.

Arrays in C. Examples

- Define an array **data** to hold 10 integers:
- **int data**[10];
- Define an array to store 100 values of type double
- **double my_array**[100];
- Assign value 34 to element stored at index 4 in **my_array**:
- **my_array**[4] = 34;

Example 1. Algorithm

- Read 100 floating point numbers input by the user.
- Print them in reverse order (the last input number is printed first, etc.).

Algorithm

- Create an array
- Read input numbers one at a time and store them in array – for loop
- Print array contents in reversed order - ??

Example 1. C Code

```c
/* partial C code – declarations*/
#include <stdio.h>
int main(void)
{ //declarations
    double a[100];
    int i;




    return 0;
}
```

Example 1. C Code

```
/* partial C code – reading input */
#include <stdio.h>
int main(void)
{ //declarations
    double a[100];
    int i;
   printf("Please input 100 floating point numbers: ");

  for(i=0;i<100;i++){
    //store current input in array at index i
       ….
    } //end for loop



    return 0;
}
```

Example 1. C Code

```
/* partial C code – reading input */
#include <stdio.h>
int main(void)
{ //declarations
    double a[100];
    int i;
   printf("Please input 100 floating point numbers: ");

    for(i=0;i<100;i++){
       //store current input in array at index i
       scanf("%lf", &a[i]);
    }



    return 0;
}
```

Example 1. C Code

- How to print array contents in reversed order?
- Walk through the array from the back to the beginning and print each element.

```
/* partial C code – printing array contents in
   reversed order */
#include <stdio.h>
int main(void)
{
    ….
    for(i=99; i>=0; i--)
       printf("%f\n", a[i]);

    return 0;
}
```

Example 1. C Code

```
/* complete C code */
#include <stdio.h>
int main(void)
{ //declarations
    double a[100];
    int i;
    printf("Please input 100 floating point numbers: ");

    for(i=0;i<100;i++){
       //store current input in array at index i
       scanf("%lf", &a[i]);
    }
    for(i=99; i>=0; i--)
       printf("%f\n", a[i]);
    return 0;
}
```

Example 2

- Read 20 integers input by the user and store them in an array then determine if the array is sorted in non-decreasing order.

Algorithm:

- Read input
- Loop through the array
  - check if current item <= next item
- If the condition is satisfied for all items then the array is sorted
- If the condition is violated for at least one item, then the array is not sorted
- Use a **flag** variable.
  - Initialize flag to 1 before the loop.
  - Inside the loop, only if condition is false set flag to 0.
- After loop ends check value of flag
  - flag == 1 means array is sorted
  - flag == 0 means array not sorted

C Code.

```
// partial C code - declarations
#include <stdio.h>
int main(void){
    int a[20], i, flag=1;









    return 0;     } // end of main
```

C Code.

```
// partial C code - reading the input
#include <stdio.h>
int main(void){
    int a[20], i, flag=1;
    printf("Please input 20 integers: ");
    for(i=0; i<20; i++)
    {     scanf("%d",&a[i]);    }




    return 0;     } // end of main
```

C Code.

```
// partial C code – looping through array
#include <stdio.h>
int main(void){
    int a[20], i, flag=1;
    printf("Please input 20 integers: ");
    for(i=0; i<20; i++)
    {     scanf("%d",&a[i]);  }
    for(i=0; i<?; i++)
    {
                                        } //end for loop



    return 0;     } // end of main
```

C Code.

```
// partial C code – looping through array
#include <stdio.h>
int main(void){
    int a[20], i, flag=1;
    printf("Please input 20 integers: ");
    for(i=0; i<20; i++)
    {     scanf("%d",&a[i]);     }
    for(i=0; i<?; i++)
    { /*  if (  current item > next item )
              flag = 0;     */  } //end for loop
```

C Code.

```
// partial C code
#include <stdio.h>
int main(void){
    int a[20], i, flag=1;
    printf("Please input 20 integers: ");
    for(i=0; i<20; i++)
    {     scanf("%d",&a[i]);    }
    for(i=0; i<?; i++)
    {     if(a[i]>a[i+1])
              flag = 0;      }



    return 0;     } // end of main
```

C Code

```
// partial C code
#include <stdio.h>
int main(void){
    int a[20], i, flag=1;
    printf("Please input 20 integers: ");
    for(i=0; i<20; i++)
    {     scanf("%d",&a[i]);     }
    for(i=0; i<19; i++)
    {     if(a[i]>a[i+1])
              flag = 0;     }



    return 0;     } // end of main
```

C Code. Variant 1

```
// complete C code
#include <stdio.h>
int main(void){
    int a[20], i, flag=1;  printf("Please input 20 integers: ");
    for(i=0; i<20; i++)
    {      scanf("%d",&a[i]);      }
    for(i=0; i<19; i++)
    {      if(a[i]>a[i+1])
                flag = 0;                }
    if (flag==1)
          printf("The array is sorted");
    else
          printf("The array is not sorted");
    return 0;     } // end of main
```

C Code. Variant 2

```
//when flag becomes 0, may exit loop - modify loop continuation condition
#include <stdio.h>
int main(void){
    int a[20], i, flag=1; printf("Please input 20 integers: ");
    for(i=0; i<20; i++)
    {     scanf("%d",&a[i]);      }
    for(i=0; i<19 && flag ==1; i++)
    {     if(a[i]>a[i+1])
              flag = 0;                }
    if (flag==1)
          printf("The array is sorted");
    else
          printf("The array is not sorted");
    return 0;     } // end of main
```

C Code. Variant 3

```
//when flag becomes 0, may exit loop – use break
#include <stdio.h>
int main(void){
    int a[20], i, flag=1; printf("Please input 20 integers: ");
    for(i=0; i<20; i++)
    {     scanf("%d",&a[i]);     }
    for(i=0; i<19; i++)
    {     if(a[i]>a[i+1])
          {    flag = 0; break;    }          }
    if (flag==1)
          printf("The array is sorted");
    else
          printf("The array is not sorted");
    return 0;     } // end of main
```