

C PROGRAMMING LANGUAGE

SINGLE-DIMENSIONAL ARRAYS

Le Thi Ngoc Tho, Ph.D

Faculty of Information Technology, HUTECH

ltn.tho@hutech.edu.vn



1

CONTENTS

- What?
- How to declare?
- How to use?
- Basic Operations:
 - Input, Output,
 - Count, Sum, Average,
 - Insert, Delete,
 - Search, Sort
- Character arrays

2

REMIND

- Please solve the following problems:
 - Input 2 integers and sum up them.
 - Input 5 integers and sum up them.
 - Input 10 integers and sum up them.
 - Input N integers and sum up them. N is input by user.
- How will you do?

REMIND

- Input 2 integers:

```
#include<stdio.h>
#include<conio.h>
int main() {
    int a, b, sum;
    printf("Input two integers:");
    printf("\n a = "); scanf("%d", &a);
    printf("\n b = "); scanf("%d", &b);
    sum = a + b;
    printf("\nSum is %d\n", sum);
    return 0;
}
```

REMIND

- Input 5 integers:

```
#include<stdio.h>
#include<conio.h>
int main() {
    int a, b, c, d, e, sum;
    printf("Input two integers:");
    printf("\n a = "); scanf("%d", &a);
    ...
    printf("\n e = "); scanf("%d", &e);
    sum = a + b + c + d + e;
    printf("\nSum is %d\n", sum);
    return 0;
}
```

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

5

5

REMIND

- Input 10 integers:

```
#include<stdio.h>
#include<conio.h>
int main() {
    int a0, a1, a2, a2, a4;
    int a5, a6, a7, a8, a9, sum;
    printf("Input two integers:");
    printf("\n a1 = "); scanf("%d", &a0);
    ...
    printf("\n a9 = "); scanf("%d", &a9);
    sum = a0 + a1 + a2 + a3 + a4 +
          a5 + a6 + a7 + a8 + a9;
    printf("\nSum is %d\n", sum);
    return 0;
}
```

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

6

6

REMIND

- Input N integers:

```
#include<stdio.h>
#include<conio.h>
int main() {
    int a0, a1, a2, a3, a4;
    int a5, a6, a7, a8, a9, sum;
    printf("Input two integers:");
    printf("\n a1 = "); scanf("%d", &a0);
    ...
    printf("\n a9 = "); scanf("%d", &a9);
    sum = a0 + a1 + a2 + a3 + a4 +
        a5 + a6 + a7 + a8 + a9;
    printf("\nSum is %d\n", sum);
    return 0;
}
```

How many
variables will you
declare here?

Use array to store
sequential collection
of elements of the
same type

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

7

7

WHAT IS ARRAY?

- Array: a range of a particular type of thing
 - This is an array of cars



- In programming:
 - collection of items stored at contiguous memory locations;
 - store multiple items of the same type together.

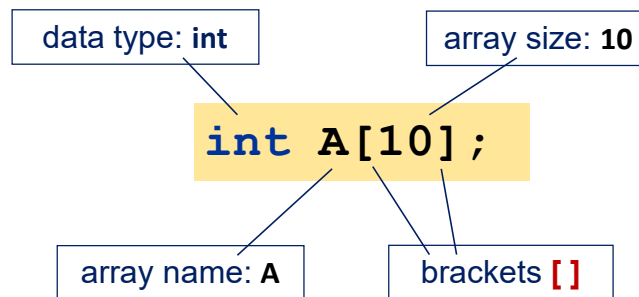
Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

8

8

HOW TO DECLARE?

- Syntax: declare an array of ten integers



Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

9

9

HOW TO DECLARE?

- Syntax: declare an array of ten integers

```
int A[10];
```

- Meaning: Tell computer to allocate 10 contiguous positions in memory to store 10 integers.

pos	200	201	202			...				209
value										
index	0	1	2	3	4	5	6	7	8	9

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

10

10

HOW TO DECLARE?

- Examples: Choose declarations of

- An array of 10 floats: _____
- An array of 8 characters: _____
- An array of 4 integers: _____
- An array of 2 doubles: _____

<code>float A[5];</code>	<code>double A[2];</code>	<code>int A[4];</code>
<code>float A[10];</code>	<code>char A[8];</code>	<code>double A[5];</code>

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

11

11

HOW TO USE?

- Given an array of five integers, `int A[5];`, how to access to elements in array?

pos	200	201	202	203	204
value					
index	0	1	2	3	4

- Directly access using index: `A[0]`, `A[1]`, `A[2]`, `A[3]`, `A[4]`
- Note that: Index always begin at **ZERO**.

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

12

12

EXAMPLES

- Problems:
 - Input 2 integers and sum up them.
 - Input 5 integers and sum up them.
 - Input 10 integers and sum up them.
 - Input N integers and sum up them.

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

13

13

BASIC OPERATIONS

- Input, Output,
- Count, Sum, Average,
- Insert, Delete,
- Search, Sort

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

14

14

BASIC OPERATIONS

- Write a program to input 10 integers and print all them out.

```
#include<stdio.h>
int main() {
    int A[10];
    printf("Input integers:\n");
    for(int i = 0; i < 10; i++){
        printf("A[%d] = ", i);
        scanf("%d", &A[i]);
    }
    printf("Input array: ");
    for(int i = 0; i < 10; i++){
        printf("%d ", A[i]);
    }
    return 0;
}
```

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

15

15

BASIC OPERATIONS

- Note: We can input values for elements of array in declaration as the following:

- Clearly specify the array size:

```
int A[5] = {1, 2, 3, 4, 5};
```

- Let computer automatically decide the array size:

```
int A[ ] = {1, 2, 3, 4, 5};
```

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

16

16

BASIC OPERATIONS

- Count the number of evens in array of 10 integers:

```
int A[10];
int count = 0;
//Note: Write code to input array A by yourself

for(int i = 0; i < 10; i++){
    if(A[i]%2 == 0)
        count++;
}
printf("The number of evens is: %d", even);
```

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

17

17

BASIC OPERATIONS

- Sum up all evens in array of 10 integers:

```
int A[10];
int sum = 0;
//Note: Write code to input array A by yourself

for(int i = 0; i < 10; i++){
    if(A[i]%2 == 0)
        sum += A[i];
}
printf("Sum of all evens is: %d", sum);
```

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

18

18

BASIC OPERATIONS

- Average of all evens in array of 10 integers:

```
int A[10];
int sum = 0;
int count = 0;
//Note: Write code to input array A by yourself

for(int i = 0; i < 10; i++){
    if(A[i]%2 == 0){
        sum += A[i];
        count++;
    }
}
printf("Average of all evens is: %f", (float)sum/count);
```

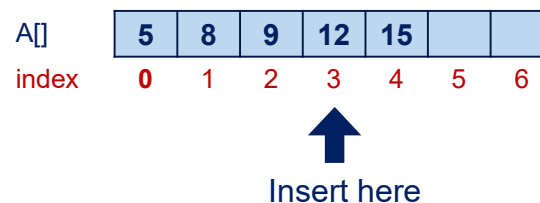
Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

19

19

BASIC OPERATIONS

- Insert an integer x to position i^{th} in array A
- What would you do to insert $x = 10$ to array A, at index 3?



Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

20

20

BASIC OPERATIONS

- What would you do to insert $x = 10$ to array A, at index 3?

A[]	5	8	9	12	15		
index	0	1	2	3	4	5	

```
for(i = N-1; i>3; i++){
    A[i+1] = A[i];
}
N++;
```

A[]	5	8	9		12	15	
index	0	1	2	3	4	5	

A[]	5	8	9	10	12	15	
index	0	1	2	3	4	5	

```
A[3] = 10;
```

↑
Insert here

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

21

21

BASIC OPERATIONS

- Delete an integer x from position i^{th} in array A
- What would you do to delete $x = 10$ from array A, at index 3?

				N = 6		
				↓		
A[]	5	8	9	10	12	15
index	0	1	2	3	4	5

```
for(i = 3; i<N-1; i++){
    A[i] = A[i+1];
}
N--;
```

A[]	5	8	9	12	15	15
index	0	1	2	3	4	5

↑
N = 5

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

22

22

BASIC OPERATIONS

- Search for an element x in array A
- What would you do to find the index of element $x = 10$?

$A[]$	5	8	9	10	12	15	
index	0	1	2	3	4	5	6

- Expected output: 3

```
int position = -1;
for(i = 0; i < N; i++){
    if(A[i] == x)
        position = i;
}
```

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

23

23

EXERCISE 1

- Write a C program to input an array A of N integers, where N is input by user ($0 < N < 100$). Then, print out the follows:
 1. All elements in array A .
 2. Elements which are even integers.
 3. Elements which are odd integers.
 4. Elements which are positive.
 5. Elements which are negative.
 6. Elements which are prime numbers.

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

24

24

EXERCISE 2

- Write a C program to input an array A of N integers, where N is input by user ($0 < N < 100$). Then, **count** and **print out** the follows:
 1. The number of even integers.
 2. The number of odd integers.
 3. The number of positive numbers.
 4. The number of negative numbers.
 5. The number of prime numbers.

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

25

25

EXERCISE 3

- Write a C program to input an array A of N integers, where N is input by user ($0 < N < 100$). Then, **sum up** and **print out** the follows:
 1. The number of even integers.
 2. The number of odd integers.
 3. The number of positive numbers.
 4. The number of negative numbers.
 5. The number of prime numbers.

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

26

26

EXERCISE 4

- Write a C program to input an array A of N integers, where N is input by user ($0 < N < 100$). Then, **compute** the follows and **print out**:
 1. The average of even integers.
 2. The average of odd integers.
 3. The average of positive numbers.
 4. The average of negative numbers.
 5. The average of prime numbers.

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

27

27

EXERCISE 5

- Write a C program to input an array A of N integers, where N is input by user ($0 < N < 100$). Then, **insert** a new element at the following positions:
 1. At the end of the array.
 2. At the beginning of the array.
 3. At the middle of the array. (middle = $N/2$)
 4. Position specified by user.

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

28

28

EXERCISE 6

- Write a C program to input an array A of N integers, where N is input by user ($0 < N < 100$). Then, **delete** an element at the following positions:
 1. At the end of the array.
 2. At the beginning of the array.
 3. At the middle of the array. (middle = $N/2$)
 4. Position specified by user.

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

29

29

EXERCISE 7

- Write a C program to input an array A of N integers, where N is input by user ($0 < N < 100$). Then, **search** for the position of element:
 1. Whose value is specified by user.
 2. Which is the first / last even integer.
 3. Which is the first / last odd integer.
 4. Which is the first / last positive number.
 5. Which is the first / last negative number.
 6. Which is the first / last prime number.

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

30

30

CHARACTER ARRAY

- Use for storing array of characters.
- E.g., `char c[10]` stores at most 10 characters.
- String: A character array ending by a **NULL** character (`'\0'`)
- E.g.,

N	G	O	C	␣	T	H	O	\0
0	1	2	3	4	5	6	7	8

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

31

31

CHARACTER ARRAY

- Print out an array of characters:

```
int main() {
    char name[] = "NGOC THO" ;
    int i = 0 ;
    while(i <= 8) {
        printf("%c", name[i]);
        i++ ;
    }
    return 0;
}
```

```
printf("%s", name);
```

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

32

32

CHARACTER ARRAY

- `scanf()` is not capable of receiving multi-words strings → What should we do?
- We use `gets()` to receive strings, and `puts()` to print string.

```
int main() {
    char name[25] ;
    printf("Enter your name: ");

    gets(name);
    puts("Hello!");
    puts(name);

    return 0;
}
```

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

33

33

CHARACTER ARRAY

- Some functions for strings:

strlen(str)	Finds length of a string
strcat(target, source)	Appends one string at the end of another
strcpy(target, source)	Copies a string into another
strcmp(str1, str2)	Compares two strings

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

34

34

CHARACTER ARRAY

- Example: write a program
 - Input your first name and your last name
 - Combine them into your full name
 - Print out the length of your full name
 - Tell me whether your first name should be before or after the name *"hellen"* in a list.

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

35

35

CHARACTER ARRAY

- Example: write a program

```
#include<stdio.h>
#include <string.h>
int main() {
    char first[10], last[10], full[20];
    printf("First name: "); gets(first);
    printf("Last name: "); gets(last);
    strcpy(full, first);
    strcat(full, last);
    printf("Your full name: ");    puts(full);

    return 0;
}
```

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

36

36

CHARACTER ARRAY

- Example: write a program

```
#include<stdio.h>
#include <string.h>
int main() {
    // Continue with code in the previous slide
    int length = strlen(full);
    printf("Your name length is: %d", length);
    int compare = strcmp(first, "hellen");
    printf("\nComparision result is %d\n", compare);
    if(compare == 0)
        printf("Your name is identical with \"hellen\"");
    else if(compare > 0)
        printf("Your name is after \"hellen\"");
    else
        printf("Your name is before \"hellen\"");
    return 0;
}
```

Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

37

37

Any Questions?



Le T.N. Tho - Basic Programming - Single-Dimensional Arrays

38

38