

## Array : 1-dimension

อนันต์ ผลเพิ่ม

Anan Phonphoem  
[anan@cpe.ku.ac.th](mailto:anan@cpe.ku.ac.th)

## Array?

- A type of structure that divided into small pieces
- Each piece has the same data type
- Array is used to store:
  - List of integer
  - List of characters
  - List of scores
  - ...

## Understanding Array

To keep scores of 6 students

### By not using array

```
Var  
score1, score2, score3: integer;  
score4, score5, score6: integer;
```

### By using array

```
var  
score: array[1..6] of integer;
```

## Understanding Array

```
var score: array[1..6] of integer;
```

Score[1]	90
Score[2]	75
Score[3]	40
Score[4]	67
Score[5]	55
Score[6]	30

## Syntax

ARRAY [ *Index Range* ] of *Data Type*;

```
Array[1..6] of integer  
Array[1..6] of 1..100  
Array[10..15] of real  
Array[3..5] of Boolean  
Array['a'..'z'] of integer  
Array[Monday..Sunday] of Boolean
```

## Array Type Declaration

```
Var score: array[1..6] of integer;
```

```
Type  
MyArray: array[1..6] of integer;  
Var  
score: MyArray;
```

```
Type  
IndexRange: 1..6;  
MyArray: array[IndexRange] of integer;  
Var  
score: MyArray;
```

```
Const Range = 6;  
Type  
MyArray: array[1..Range] of integer;  
Var  
score: MyArray;
```

## Accessing Array Elements

**Var Score: array[1..6] of integer;**

```

Score[1] := 10;
Score[6] := 30;

For I := 1 to 6 do
    score[I] := I+1;

I:= 2;
Writeln(Score[I+2]);

```

## Accessing Array Elements Test...

Score[1]	90
Score[2]	75
Score[3]	40
Score[4]	67
Score[5]	55
Score[6]	30

**X:=1; Y:=2;**

```

Score[1] =
Score[X] =
Score[Y+1] =
Score[Y] + 1 =
Score[Y] + Score[X] =
Score[X+Y] =

```

## Write a program to find an average of 6 real number

```

Program Find_score2(input,output);
Const ListSize = 6;
Type
    MyArray = array[1..ListSize] of real;
Var
    I: integer; sum: real;
    Score : MyArray;
Begin
    for I := 1 to 6 do
        readln(Score[I]);
    sum := 0;
    for I := 1 to 6 do
        sum := sum + Score[I];
    average_score := sum /6;
    writeln('the average is',average_score:5:2);
End.

```

## Write a program find average score of 6 students (not using array)

```

Program assignment1(input,output);
Var score1,score2,score3,score4,score5,score6: real;
    ans:real;
Function find_avg(s1,s2,s3,s4,s5,s6:real): real;
Begin
    find_avg := (s1+s2+s3+s4+s5+s6)/6;
End;
Procedure read_score(var s1,s2,s3,s4,s5,s6:real);
Begin
    readln(s1); readln(s2);readln(s3);
    readln(s4); readln(s5);readln(s6);
End;
Begin
    read_score(score1,score2,score3,score4,score5,score6);
    ans := find_avg(score1,score2,score3,score4,score5,score6);
    writeln('The average is',ans:5:2);
End.

```

## Write a program find average score of 6 students (using array)

```

Program assignment2(input,output);
Const LRange = 6;
Type
    MyArray = array[1..LRange] of real;
Var
    score: MyArray;
    ans:real;
Function find_avg(s:MyArray): real;
Var I:integer; sum: real;
Begin
    for I := 1 to LRange do
        sum := sum + s[I];
    find_avg := sum/LRange;
End;
Procedure read_score(var s:MyArray);
Var I: integer;
Begin
    for I := 1 to LRange do
        readln(s[I]);
    End;
Begin
    read_score(score);
    ans := find_avg(score);
    writeln('The average is',ans:5:2);
End.

```

## Practice... Write a program to count letters

- What to do?
  - Read input data
  - Count each letter
  - Display the result
- Plan your program:
  - Use procedure to write each task
  - Use 1 dimension array

## Practice...

### Write a program to count letters

- Procedure:
  - Initialization the array
  - Read input and count letters
  - Display the data

## Practice ...

### Sample result

Please input any letter and press enter when done

Hello 204111 students

Letter	Count
d	1
e	2
l	2
n	1
o	1
s	2
t	2
u	1

## Main Program

```
program Letter_counter(input,output);
{ Count the number of the occurrence of each letter
and display it to screen }
type
  letter = 'a'..'z';
  letter_array = array[letter] of Integer;
Var
  letter_count : letter_array;
...
begin {main}
  Init(letter_count,0); {Initialization}
  Count(letter_count); {Count letter}
  display(letter_count); {Display the letter count results}
end. {main}
```

## Procedure : Init

```
procedure init(var letter_count : letter_array;
               value:integer);
{ Setup all elements in the array to the desired value }
Var Ch :char;
begin
  for Ch := 'a' to 'z' do
    letter_count[Ch] := value;
end; {procedure initialization}
```

## Procedure : count

```
procedure count(var letter_count : letter_array);
{ count each letter and keep it in the array }
Var Ch : char;
begin
  writeln('Please input any letter and press enter when done');
  while not EOLN do
    begin
      Read(Ch);
      if (ch >= 'a') and (ch <= 'z') then
        letter_count[Ch] := letter_count[Ch] + 1;
      end; {while}
    Readln;
  end; {Procedure count}
```

## Procedure : display

```
procedure display( letter_count : letter_array);
{ Display the counting result to screen }
Var next : letter;
begin
  writeln('Letter':8,'Count':8);
  for next := 'a' to 'z' do
    if letter_count[next] > 0 then
      writeln(next:8, letter_count[next]:8);
  end; {Procedure display}
```