GEDT019: Basic and Practice in Programming

- Lab 6: Array

In this lab ...

- Array
  - Declaration and Initialization
  - Character String
  - Multidimensional Arrays

- What you need to submit in this lab (Lab #6):
  - Lab Exercise #6 by Wednesday 23:59 pm
  - Assignment #6 by Tuesday 11:59 pm

Declaring Arrays

```
int ArrayA[5];   // integer array with 5 items
float ArrayB[4]; // float point number array with 4 items
char ArrayC[7];  // Character array with 7 items
```

Array has to be in same type.

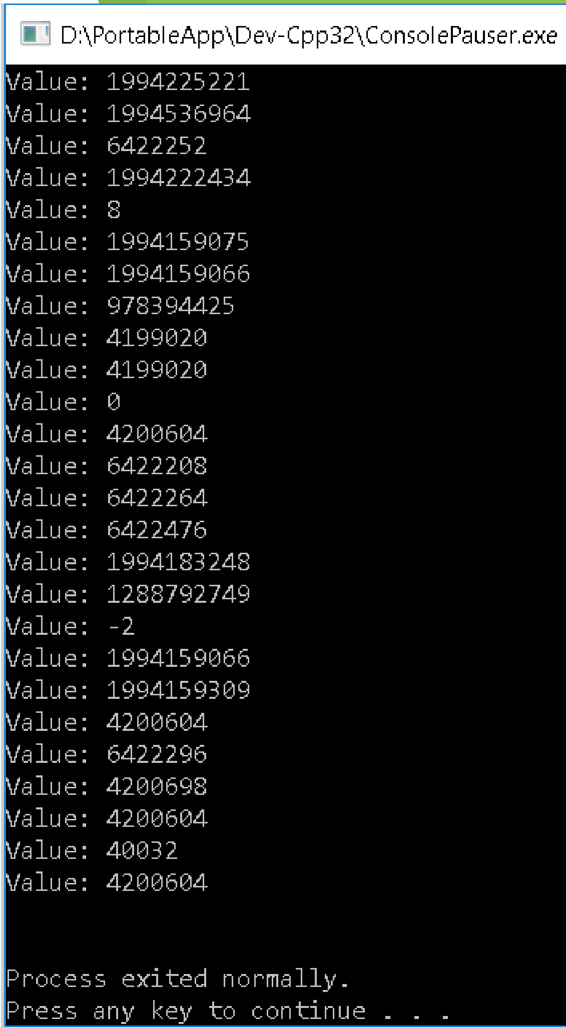
Declaring Variables without Initialization

When you declare a variable, memory is allocated to the variable.
But there is possibility that there are residual (left over data) in the memory.
If you declare a variable without initialising it, the value of the data may be the left over value in the memory allocated.

Declaring Variables without Initialization

Try the following example:

```
#include <stdio.h>

int main(void)
{  int x[10000];
   // declare an integer array with 10000
   // items, without initializing them

   for(int i=0;i<10000;i++)
    printf("Value: %i\n",x[i]);

   return 0;
}
```



```
Value: 1994225221
Value: 1994536964
Value: 6422252
Value: 1994222434
Value: 8
Value: 1994159075
Value: 1994159066
Value: 978394425
Value: 4199020
Value: 4199020
Value: 0
Value: 4200604
Value: 6422208
Value: 6422264
Value: 6422476
Value: 1994183248
Value: 1288792749
Value: -2
Value: 1994159066
Value: 1994159309
Value: 4200604
Value: 6422296
Value: 4200698
Value: 4200604
Value: 40032
Value: 4200604


Process exited normally.
Press any key to continue . . .
```

Initializing Array

```
int ArrayA[5] = {1,2,3,4,5};     // Array with 5 items
float ArrayA[] = {1.0,2.0,3.0};  // Array with 3 items
char String1[] = “Hello, World”; // Array with 11 items

int ArrayB[5] = {0};   // Initialize all items with 0
Int ArrayC[4] = {1,2};     // 4 items: {1,2,0,0}
```

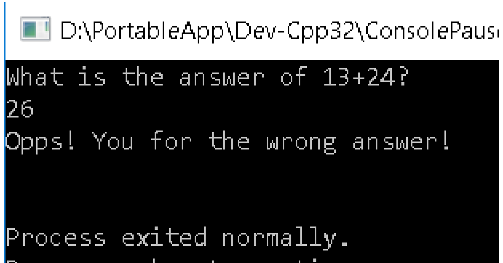
Character Array

```
char String1[] = {‘H’,’e’,’l’,’l’,’o’,’\0’};
char String2[] = “Hello”;
```

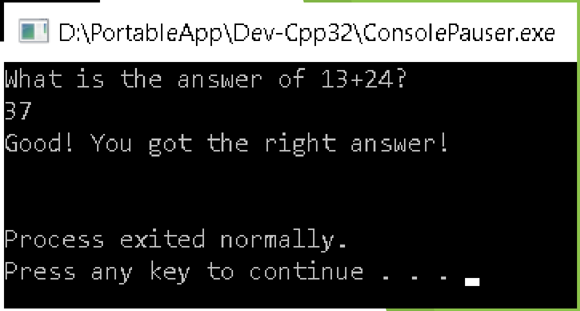
`String1` and `String2` are identical.
`’\0’` is a null character used to signify end of sequence.
A final null character ('\0') will be put at the end of the character array when it is initialized using the “”.

Example

```
#include <stdio.h>

int main (void)
{ char Question[] = “What is the answer of 13+24?”;
  char Correct[] = “Good! You got the right answer!”;
  char Wrong[] = “Opps! You for the wrong answer!”;
  int answer = 0;

  printf(“%s\n”,Question);
  scanf(“%d”,&answer);
  if(answer == 37)
    printf(“%s\n”,Correct);
  else
    printf(“%s\n”,Wrong);

  return 0;
}
```



```
What is the answer of 13+24?
26
Opps! You for the wrong answer!


Process exited normally.
Press any key to continue .
```



```
What is the answer of 13+24?
37
Good! You got the right answer!


Process exited normally.
Press any key to continue . . .
```

Lab Exercise #6: Array

1. Create an array consisting of **20 random integers** in the range of 25 -100 (including 25 and 100).
2. Print out this array.
3. Find the smallest value in the array and display its value.
4. Find the largest value in the array and display its value.
5. Calculate the average value in the array and display its value.

Submit it on iCampus before Wednesday 23:59 pm.

Multi-dimensional Array

You can declare arrays with multi-dimension, for example:

```
int StudentScore[3][3] = {{95, 92, 93},
                        {89, 98, 82},
                        {90, 87, 88}};
```

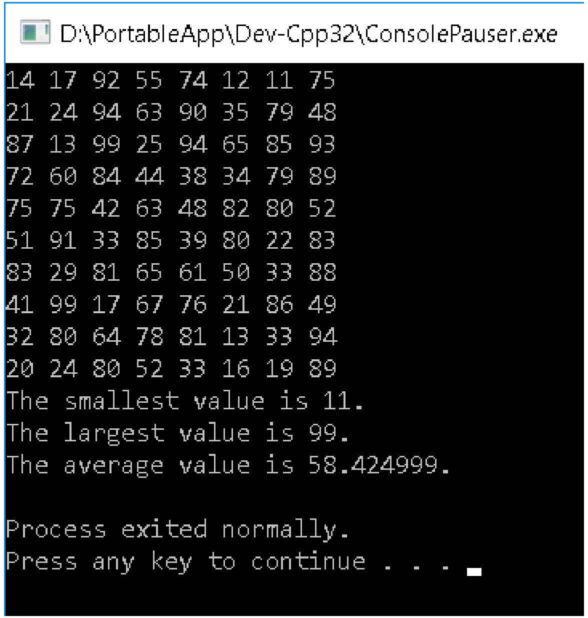
will declare a 3x3 two dimensional array.

Assignment #6: Array

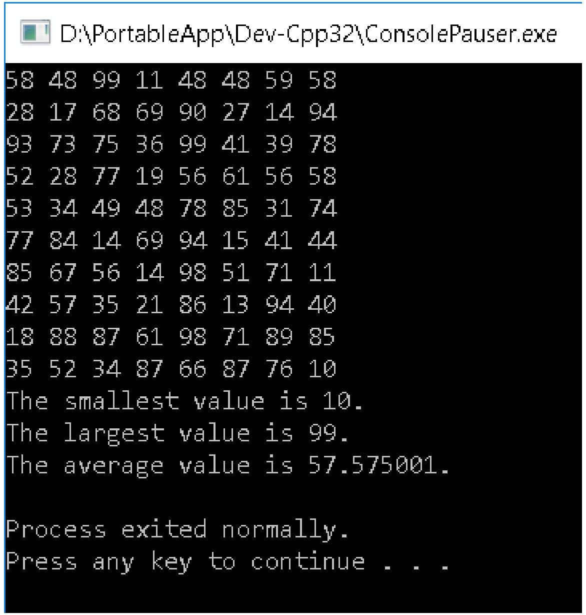
1. Create a 8 x 10 two dimensional array consisting of random integers in the range of 10-100 (including 10 and 100).
2. Print out this 8 x 10 two dimensional array in a table form.
3. Find the smallest value in the two dimensional array and display its value.
4. Find the largest value in the two dimensional array and display its value.
5. Calculate the average value in the two dimensional array and display its value.

Assignment #6: Array

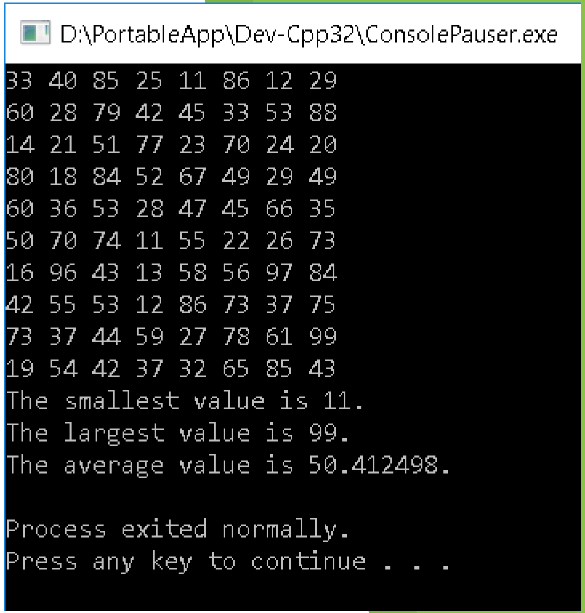
Sample outputs:



```
14 17 92 55 74 12 11 75
21 24 94 63 90 35 79 48
87 13 99 25 94 65 85 93
72 60 84 44 38 34 79 89
75 75 42 63 48 82 80 52
51 91 33 85 39 80 22 83
83 29 81 65 61 50 33 88
41 99 17 67 76 21 86 49
32 80 64 78 81 13 33 94
20 24 80 52 33 16 19 89
The smallest value is 11.
The largest value is 99.
The average value is 58.424999.

Process exited normally.
Press any key to continue . . .
```



```
58 48 99 11 48 48 59 58
28 17 68 69 90 27 14 94
93 73 75 36 99 41 39 78
52 28 77 19 56 61 56 58
53 34 49 48 78 85 31 74
77 84 14 69 94 15 41 44
85 67 56 14 98 51 71 11
42 57 35 21 86 13 94 40
18 88 87 61 98 71 89 85
35 52 34 87 66 87 76 10
The smallest value is 10.
The largest value is 99.
The average value is 57.575001.

Process exited normally.
Press any key to continue . . .
```



```
33 40 85 25 11 86 12 29
60 28 79 42 45 33 53 88
14 21 51 77 23 70 24 20
80 18 84 52 67 49 29 49
60 36 53 28 47 45 66 35
50 70 74 11 55 22 26 73
16 96 43 13 58 56 97 84
42 55 53 12 86 73 37 75
73 37 44 59 27 78 61 99
19 54 42 37 32 65 85 43
The smallest value is 11.
The largest value is 99.
The average value is 50.412498.

Process exited normally.
Press any key to continue . . .
```

Assignment #6:

Submit your source code on iCampus before Tuesday 23:59 pm