



eng. Keroles Shenouda

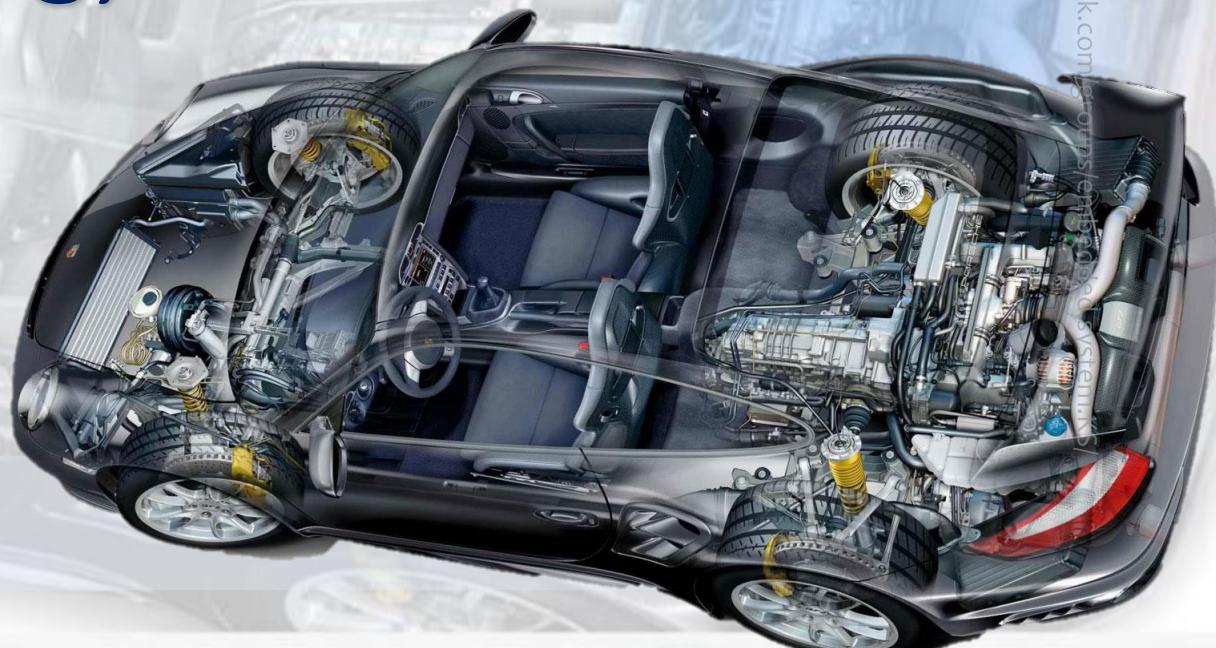
<https://www.facebook.com/groups/embedded.system.KS/>

Embedded System

PART 3(C Programming)

➤ Array/String

ENG.KEROLES SHENOUDA



Data Conversion and Type Casting

```
main.c ✘
1 //Prepared by keroles
2 #include <stdio.h>
3
4 //Prepared by Keroles
5 int main()
6 {
7     int radius = 5;
8     double area;
9     area = 3.14159 * radius * radius;
10    float x = 1.2;
11    int y = 5;
12    int z = 0x878925;
13    int a;
14    char b;
15    a = x; //Problem, Fraction part is lost
16    b = y; //No problem, 5 < char range(-128 to 127)
17    b = z;//Problem, 0x878925 >>> char range(-128 to 127)
18    return 0 ;
19 }
20
```





Press
here

#LEARN_IN_DEPTH

#Be_professional_in
embedded_system

3

Data Conversion and Type Casting

first_c_code.exe [C/C++ Application]
 first_c_code.exe [6608]
 Thread #1 0 (Suspended : Step)
 main() at main.c:18 0x4014d6
 Thread #2 0 (Suspended : Container)
 gdb (7.6.1)

```
main.c //Prepared by keroles
1 //Prepared by keroles
2 #include <stdio.h>
3
4 //Prepared by Keroles
5 int main()
6 {
7     int radius = 5;
8     double area;
9     area = 3.14159 * radius * radius;
10    float x = 1.2;
11    int y = 5;
12    int z = 0x878925;
13    int a;
14    char b;
15    a = x; //Problem, Fraction part is lost
16    b = y; //No problem, 5 < char range(-128 to 127)
17    b = z;//Problem, 0x878925 >>> char range(-128 to 127)
18    return 0 ;
19 }
```

Console Tasks Problems Executables Memory

:c_code.exe [C/C++ Application] first_c_code.exe

Name	Type	Value
(*)= radius	int	5
(*)= area	double	78.53974999999998
(*)= x	float	1.20000005
(*)= y	int	5
(*)= z	int	8882469
(*)= a	int	1
(*)= b	char	37 %

Name : b
 Details:37 '%'
 Default+37 '%'

Calculator

PROGRAMMER

25

HEX 25
 DEC 37
 OCT 45
 BIN 0010 0101

QWORD MS M⁺

0000 0000 0000 0000	60 56 52 48
0000 0000 0000 0000	44 40 36 32
0000 0000 0000 0000	28 24 20 16
0000 0000 0010 0101	12 8 4 0

mbedded.system.KS/

Data Conversion and Type Casting

- The conversion from larger to smaller data types may lead to some data losses. For that reason the compiler warns you against this type of operation. However sometimes the conversion does not affect the data like the second expression ($b = y$). Compiler is not able to differentiate between safe or unsafe situation, for that reason you must use the **type casting** to force the conversion if you decide that it is safe.

```

1 //Prepared by keroles
2 #include <stdio.h>
3
4 //Prepared by Keroles
5 int main()
6 {
7     int radius = 5;
8     double area;
9     area = 3.14159 * radius * radius;
10    float x = 1.2;
11    int y = 5;
12    int z = 0x878925;
13    int a;
14    char b;
15    a = (int)x; //Problem, Fraction part is lost
16    b = (char)y; //No problem, 5 < char range(-128 to 127)
17    b = (char)z;//Problem, 0x878925 >>> char range(-128 to 127)
18    return 0 ;
19
20

```

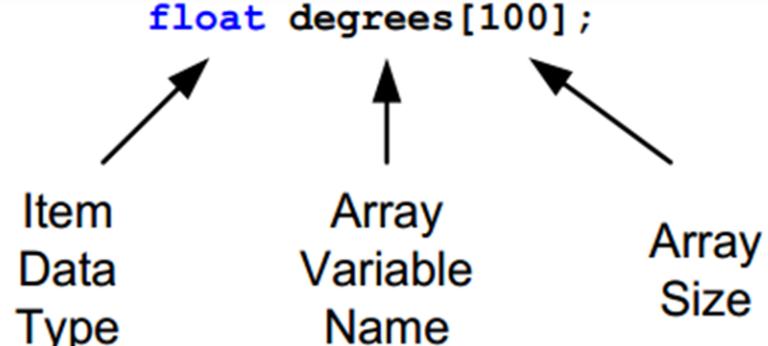
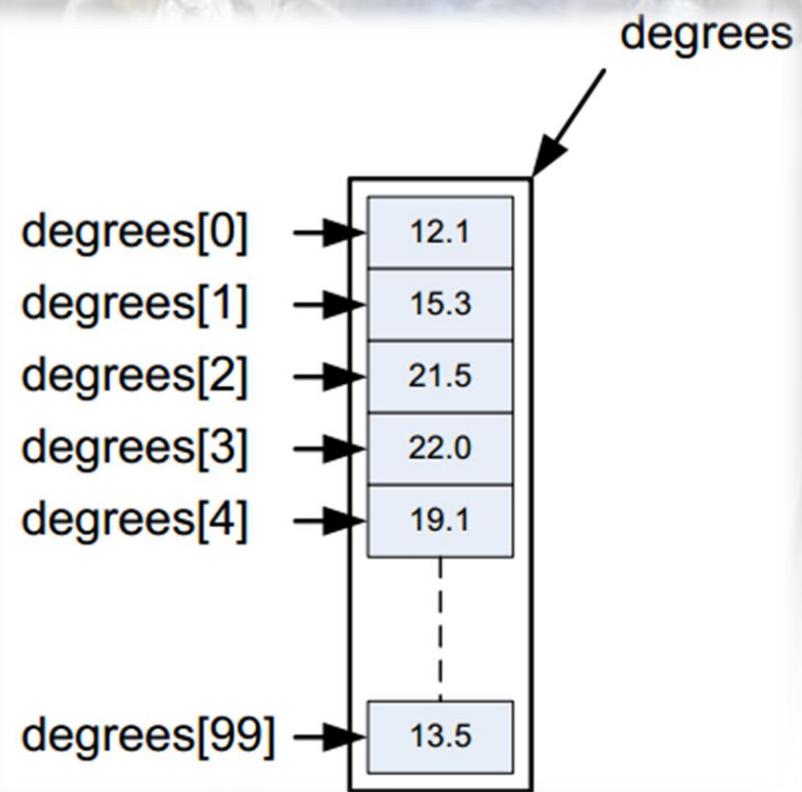


Advanced Data Types: *Arrays*

- ▶ Array is a single variable that able to hold several values.
Programmer can set or get specific value in the array.

To define an array variable containing 100 (**float**) values:

```
float degrees[100];
```

Arrays

- ▶ The position of each value is called the index. The **index** of the first value is 0, which means if the array contains 10 items the **index range from 0 to 9**.
- ▶ Dealing with array elements is very similar to normal variables; you can assign a value to any item or assign any item to other program variables. Following code illustrates this idea:

```
float degrees[100];
float x=9.9;
float y;
//assign x value to the fifth item
degrees[4] = x;
//assign fifth item to y
y = degrees[4];
//scan eighth item from user
scanf("%f", &degrees[7]);
//print eighth item from user
printf("The eighth item value is %f", degrees[7]);
```

<https://www.facebook.com/groups/embedded.system.KS/>

Lab: Store and Print 10 Students Degrees

- ▶ shows how to scan 10 students degree from user, stores them in a single area, and then prints them.

```

Problems Tasks Console Properties AVR Device Explorer AVR Supported MCUs
<terminated> (exit value: 0) session2.exe [C/C++ Application] D:\courses\C_Course\session2\Debug\session2.exe (3/31/17, 9:34 AM)

Enter student 1 degree : 1
Enter student 2 degree : 2
Enter student 3 degree : 3
Enter student 4 degree : 4
Enter student 5 degree : 5
Enter student 6 degree : 6
Enter student 7 degree : 7
Enter student 8 degree : 8
Enter student 9 degree : 9
Enter student 10 degree : 10

Student 1 degree is 1.000000
Student 2 degree is 2.000000
Student 3 degree is 3.000000
Student 4 degree is 4.000000
Student 5 degree is 5.000000
Student 6 degree is 6.000000
Student 7 degree is 7.000000
Student 8 degree is 8.000000
Student 9 degree is 9.000000
Student 10 degree is 10.000000

```

Lab



```

1  /*  

2  * main.c  

3  *  

4  * Created on: Mar 23, 2017  

5  * Author: Keroles  

6  */  

7 #include <stdio.h>  

8  

9 int main()  

10 {  

11     int i;  

12     float degrees[10];  

13     //Scanning students degrees and storing them in array  

14     for(i=0; i<10; i++)  

15     {  

16         printf("\n Enter student %d degree : ", i+1);  

17         fflush(stdin); fflush(stdout);  

18         scanf("%f", &degrees[i]);  

19     }  

20     //Printing all students degrees  

21     for(i=0; i<10; i++)  

22     {  

23         printf("\n Student %d degree is %f",
24                i+1, degrees[i]);  

25     }  

26  

27     return 0 ;  

28 }
29

```



Lab

Solution



<https://www.facebook.com/groups>

S/





Press here

#LEARN IN DEPTH

#Be_professional_in_embedded_system

9

Name	Type	Value
(x)= i	int	4194432
(x)= degrees	float [10]	0x61ff04
(x)= degrees[0]	float	128.419312
(x)= degrees[1]	float	-nan(0x7ffffe)
(x)= degrees[2]	float	9.28242452e+031
(x)= degrees[3]	float	9.28265953e+031
(x)= degrees[4]	float	5.88661943e-039
(x)= degrees[5]	float	8.99972728e-039
(x)= degrees[6]	float	5.88674695e-039
(x)= degrees[7]	float	5.88661943e-039
(x)= degrees[8]	float	0

Name : degrees

Details:{128.419312, -nan(0x7ffffe), 9.28242452e+031, 9.28265953e+031, 5.88661943e-039, 8.99972728e-039, 5.88674695e-039, 5.88661943e-039, 0}

Default:0x61ff04

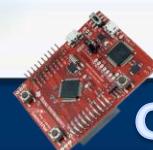
Decimal:6422276

Binary:0b10000000000000000000000000000000

```

1 * main.c
2 *
3 * Created on: Mar 23, 2017
4 * Author: Keroles
5 */
6 #include <stdio.h>
7
8
9 int main()
10 {
11     int i;
12     float degrees[10];
13     //Scanning students degrees and storing them in array
14     for(i=0; i<10; i++)
15     {
16         printf("\n Enter student %d degree : ", i+1);
17         fflush(stdin); fflush(stdout);
18         scanf("%f", &degrees[i]);
19     }
20     //Printing all students degrees

```





(x)= Variables Breakpoints Registers Modules

Name	Type	Value
(x)= i	int	2
degrees	float [10]	0x61ff04
(x)= degrees[0]	float	12
(x)= degrees[1]	float	10
(x)= degrees[2]	float	7
(x)= degrees[3]	float	9.28265953e+031
(x)= degrees[4]	float	5.88657459e-039
(x)= degrees[5]	float	8.99972728e-039
(x)= degrees[6]	float	5.88670211e-039

```

9 int main()
10 {
11     int i;
12     float degrees[10];
13     //Scanning students degrees and storing them in array
14     for(i=0; i<10; i++)
15     {
16         printf("\n Enter student %d degree : ", i+1);
17         // fflush(stdin); fflush(stdout);
18         scanf("%f", &degrees[i]);
19     }
20     //Printing all students degrees
21     for(i=0; i<10; i++)
22     {
23         printf("\n Student %d degree is %.2f"

```

Console Tasks Problems Executables Memory



Arrays

► Initializing the array:

When the array is defined, there is no data assigned to their elements.

Assigning initial values for array elements is called initializing the array. Following code section shows how to initialize the array:

```
float degrees[10] = {75.5, 88.0, 89.5, 23.5, 72.0, 63.5, 57.5,
62.0, 13.5, 46.5};
```

It is applicable in C to not to mention the array size, and let the compiler calculates it from the number of the initial values.

```
float degrees[] = {75.5, 88.0, 89.5, 23.5, 72.0, 63.5, 57.5,
62.0, 13.5, 46.5};
```



Lab: Calculate Polynomial Value for a Set of Inputs

```
#include "stdio.h"

void main()
{
    float x, y;

    x = 5;
    y = 5 * x * x + 3 * x + 2;
    printf("y(%f) = %f\r\n", x, y);

    x = 16;
    y = 5 * x * x + 3 * x + 2;
    printf("y(%f) = %f\r\n", x, y);

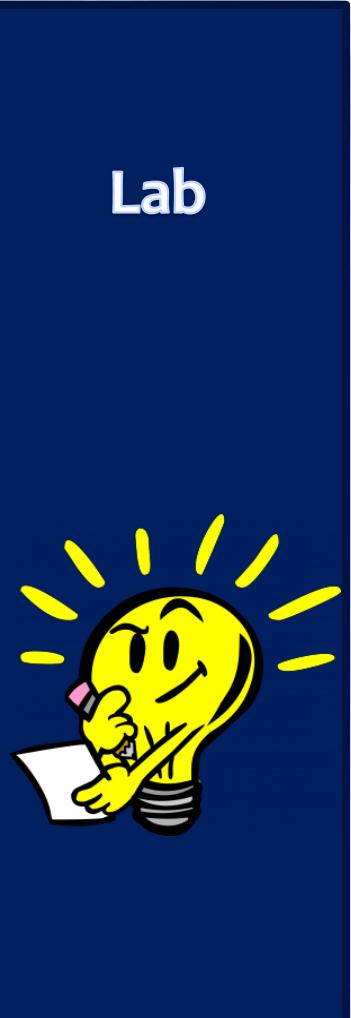
    x = 22;
    y = 5 * x * x + 3 * x + 2;
    printf("y(%f) = %f\r\n", x, y);

    x = 3.5;
    y = 5 * x * x + 3 * x + 2;
    printf("y(%f) = %f\r\n", x, y);

    x = 15;
    y = 5 * x * x + 3 * x + 2;
    printf("y(%f) = %f\r\n", x, y);
}
```

the program works correctly, however it appears that code size contains many repeated sections, each section for each value. If the number of values increases code size increases.

Alternative solve it using Arrays and Loops



Lab

<https://www.facebook.com/groups/embedded.system.KS/>



```
#include "stdio.h"

void main()
{
    float x[5] = {5, 16, 22, 3.5, 15};
    float y;
    int i;

    for(i=0;i<5;i++)
    {
        y = 5 * x[i] * x[i] + 3 * x[i] + 2;
        printf("y(%f) = %f\r\n", x[i], y);
    }
}
```

Lab

Solution



2D Arrays

2D array is suitable to hold tabular information for example, it can be used to store 4 students degrees in 6 subjects or to represent ($m \times n$) matrix.

To define an array variable containing 100 (**float**) values:

```
float matrix[3][3];
```

2D Array Variable Name

```
float matrix[3][3];
```

Item data type

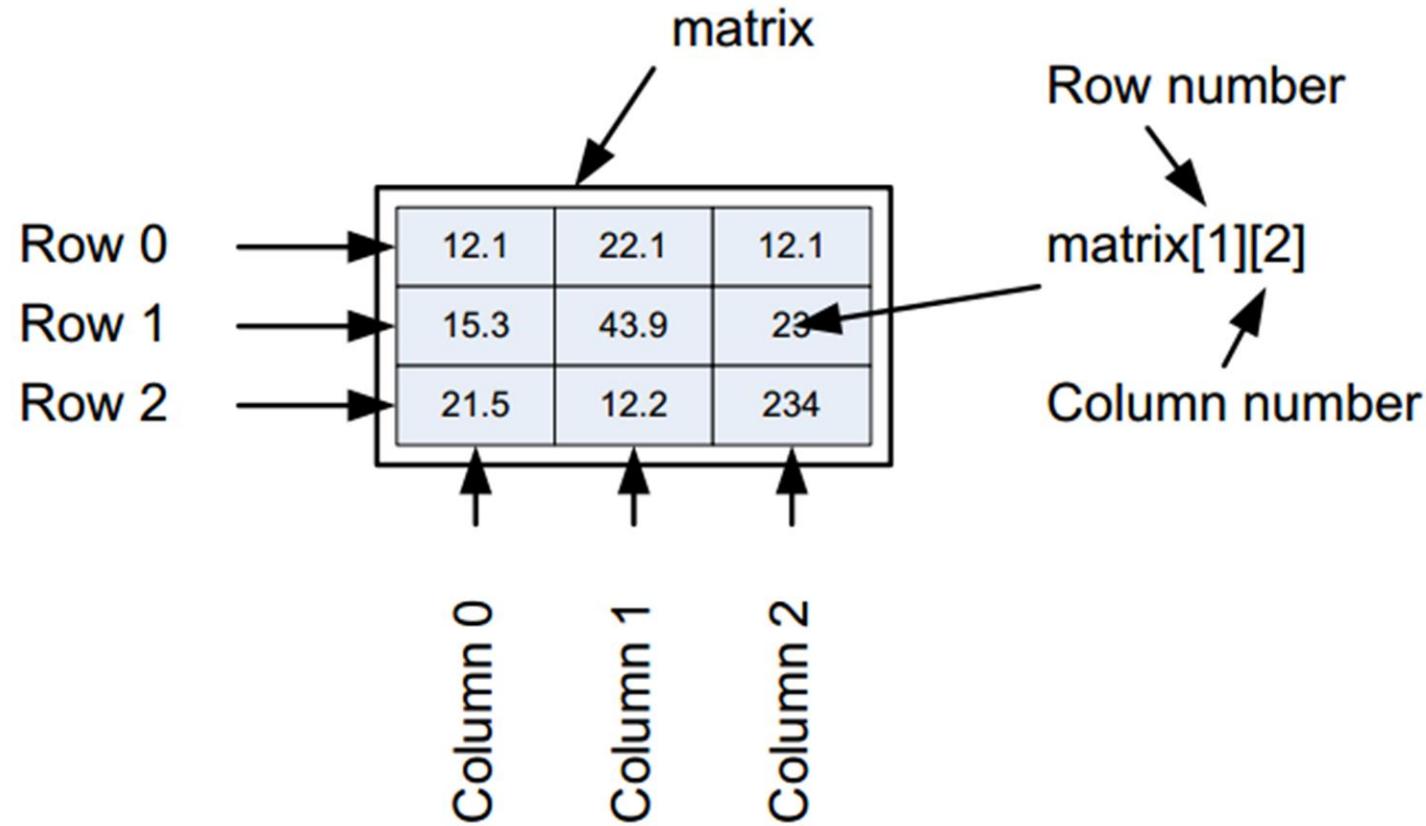
Number of Rows

Number of Columns



2D Arrays

Following figure shows how to represent the 2D array in computer memory:



Following code section shows how to use 2D arrays:

```
//define and initialize the 2D array
int degrees[4][6] = {
    {36, 28, 76, 47, 82, 33},
    {75, 49, 38, 98, 59, 83},
    {82, 65, 10, 21, 86, 22},
    {25, 63, 65, 76, 37, 21}};

int x = 9;
int y;

//assign x value to the item (2, 3)
degrees[2][3] = x;

//assign item (2,3) to y
y = degrees[2][3];

//scan item (1, 4) from user
scanf("%f", &degrees[1][4]);

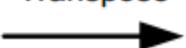
//print item (1, 4) from user
printf("The provided item value is %f", degrees[1][4]);
```

k.com/groups/embedded.system.KS/



LAB: Calculate and Print the Transpose of 3x3 Matrix

m11	m12	m13
m21	m22	m23
m31	m32	m33

Transpose


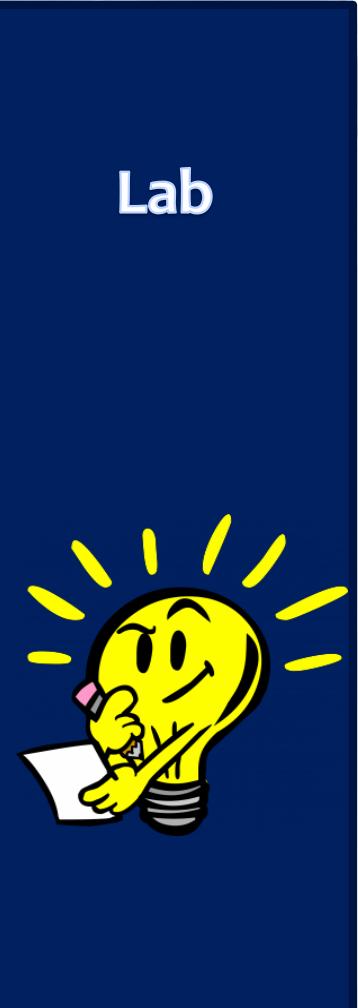
m11	m21	m31
m12	m22	m32
m13	m23	m33

```

Problems Tasks Console Properties
<terminated> (exit value: 0) session2.exe [C/C++ Application]
Enter the item(0, 0) : 0
Enter the item(0, 1) : 1
Enter the item(0, 2) : 2
Enter the item(1, 0) : 3
Enter the item(1, 1) : 4
Enter the item(1, 2) : 5
Enter the item(2, 0) : 6
Enter the item(2, 1) : 7
Enter the item(2, 2) : 8
the matrix is
0.00 1.00 2.00
3.00 4.00 5.00
6.00 7.00 8.00

the Transpose matrix is
0.00 3.00 6.00
1.00 4.00 7.00
2.00 5.00 8.00

```



Lab



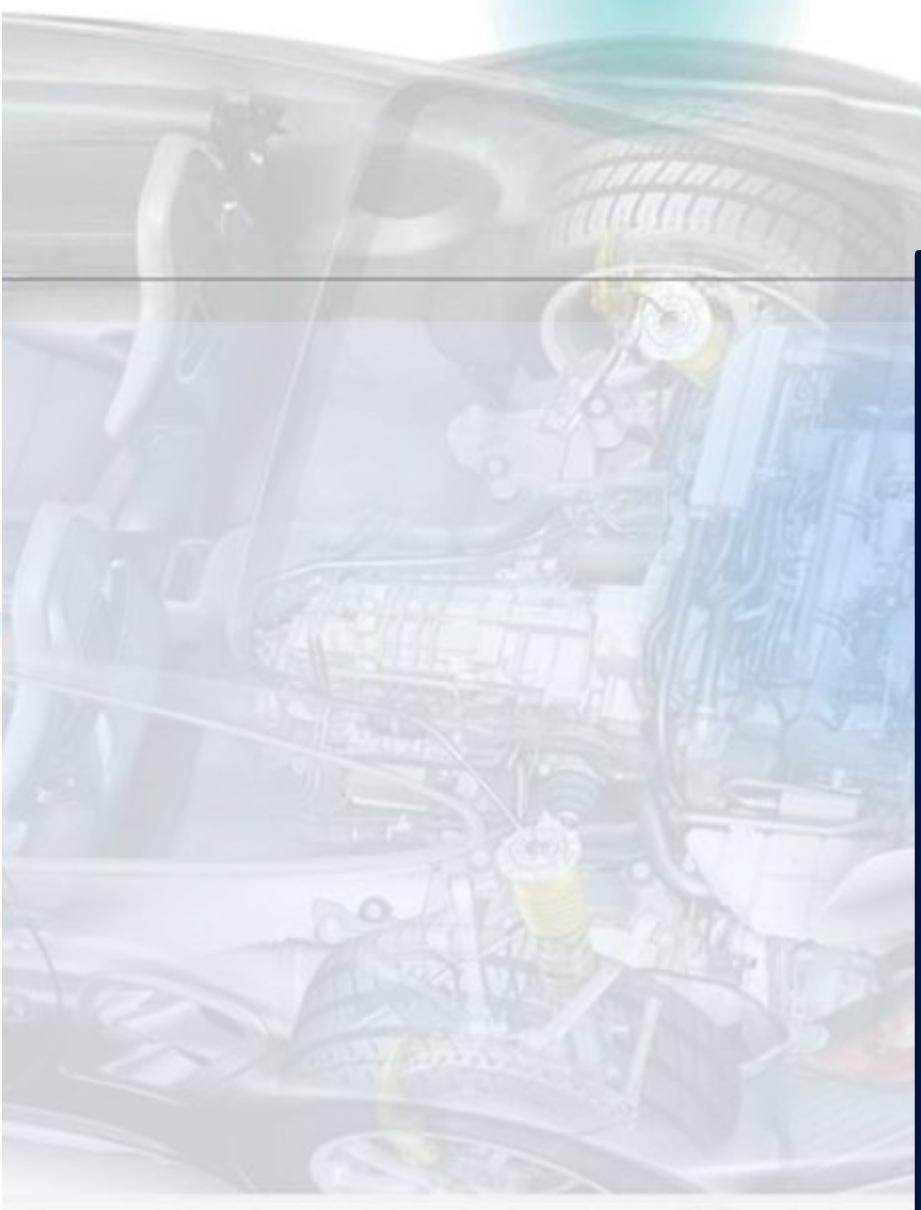


18

```

1 * 
2 * main.c
3 *
4 * Created on: Mar 23, 2017
5 * Author: Keroles
6 */
7 #include <stdio.h>
8
9 int main()
10 {
11     float a[3][3];
12     int r, c;
13     float t[3][3];
14
15
16     for(r=0; r<3; r++)
17     {
18         for(c=0; c<3; c++)
19         {
20             printf("Enter the item(%d, %d) : ", r, c);
21             fflush(stdout); fflush(stdin);
22             scanf("%f", &a[r][c]);
23         }
24     }
25     printf ("the matrix is \n");
26     for(r=0; r<3; r++)
27     {
28         for(c=0; c<3; c++)
29         {
30             printf("%2.2f\t", a[r][c]);
31         }
32         printf("\r\n");
33     }
34     printf ("the Transpose matrix is \n");
35     for(r=0; r<3; r++)
36     {
37         for(c=0; c<3; c++)
38         {
39             t[r][c] = a[c][r];
40         }
41     }
42     for(r=0; r<3; r++)
43     {
44         for(c=0; c<3; c++)
45         {
46             printf("%2.2f\t", t[r][c]);
47         }
48         printf("\r\n");
49     }
50     return 0 ;
51 }
52
53

```



Lab

Solution



<https://www.facebook.com/groups/134078713713111/>

Strings

- ▶ String is a set of several consecutive characters; each character can be represented in C language by a (char) data type.
- ▶ This means that string value can be represented by an **array of (char)**

```
char Text[] = {'h', 'e', 'l', 'l', 'o', 0};
```

Above code shows how to store “hello” string letters in array.

The last letter is sited with zero value (null termination), this value is important to tell the computer that the string is terminated before this item.

```
char text1[] = {"hello"};
char text2[] = "hello";
```



Printing String Value

- ▶ “text” variable contains 100 (char) value, only first 5 places is used to hold the word “hello” and
- ▶ the sixth place is used to hold the null termination.
- ▶ In `printf`, “%s” is used to inform the program that it will print a string value.
- ▶ **Important:** `printf` uses the null termination to end the printing operation. If the null termination is not used, the program continues printing the following memory contents until it reaches a zero.

```
.c main.c ✘
 3  *
 4  *   Created on: Mar 23, 2017
 5  *           Author: Keroles
 6  */
 7 #include <stdio.h>
 8
 9 int main()
10 {
11     char text[100] = {'h', 'e', 'l', 'l', 'o', 0};
12     printf("%s\r\n", text);
13 }
14
15
```



Name	Type	Value
text	char [100]	0x61fecc
(*)= text[0]	char	12 '\f'
(*)= text[1]	char	-1 '\'
(*)= text[2]	char	97 'a'
(*)= text[3]	char	0 '\0'
(*)= text[4]	char	-94 't'
(*)= text[5]	char	105 'i'
(*)= text[6]	char	-109 ''"
(*)= text[7]	char	116 't'
(*)= text[8]	char	8 '\b'
(*)= text[9]	char	0 '\0'
(*)= text[10]	char	0 '\0'
(*)= text[11]	char	0 '\0'
(*)= text[12]	char	115 's'
(*)= text[13]	char	115 's'
(*)= text[14]	char	-110 ''"
(*)= text[15]	char	116 't'
(*)= text[16]	char	106 'j'
(*)= text[17]	char	115 's'
(*)= text[18]	char	-110 ''"
(*)= text[19]	char	116 't'

main.c

```

1  /*
2  * main.c
3  *
4  * Created on: Mar 23, 2017
5  * Author: Keroles
6  */
7 #include <stdio.h>
8
9 int main()
10 {
11     char text[100] = {'h', 'e', 'l', 'l', 'o', 0};
12     printf("%s\r\n", text);
13     return 0 ;
14 }
15

```

<https://www.facebook.com/groups/embedded.system.KS/>



FOLLOW US 



 Press here

 #LEARN_IN_DEPTH

 #Be_professional_in_embedded_system

 22

Name	Type	Value
text	char [100]	0x61fecc
(x)= text[0]	char	104 'h'
(x)= text[1]	char	101 'e'
(x)= text[2]	char	108 'l'
(x)= text[3]	char	108 'l'
(x)= text[4]	char	111 'o'
(x)= text[5]	char	0 '\0'
(x)= text[6]	char	0 '\0'
(x)= text[7]	char	0 '\0'
(x)= text[8]	char	0 '\0'
(x)= text[9]	char	0 '\0'
(x)= text[10]	char	0 '\0'
(x)= text[11]	char	0 '\0'
(x)= text[12]	char	0 '\0'
(x)= text[13]	char	0 '\0'
(x)= text[14]	char	0 '\0'
(x)= text[15]	char	0 '\0'
(x)= text[16]	char	0 '\0'
(x)= text[17]	char	0 '\0'
(x)= text[18]	char	0 '\0'
(x)= text[19]	char	0 '\0'

.c main.c

```

1  /*
2   * main.c
3   *
4   * Created on: Mar 23, 2017
5   * Author: Keroles
6   */
7 #include <stdio.h>
8
9 int main()
{
  char text[100] = {"h", "e", "l", "l", "o", 0};
12 printf("%s\r\n", text);
13 return 0 ;
14}
15

```

<https://www.facebook.com/groups/embedded.system.KS/>



A screenshot of a terminal window. The tabs at the top are "Problems", "Tasks", and "Con". The main area shows the output of a program: "<terminated> (exit value: 0) session" followed by "helloooooo" and a blank line. The terminal has a light blue background with white text.

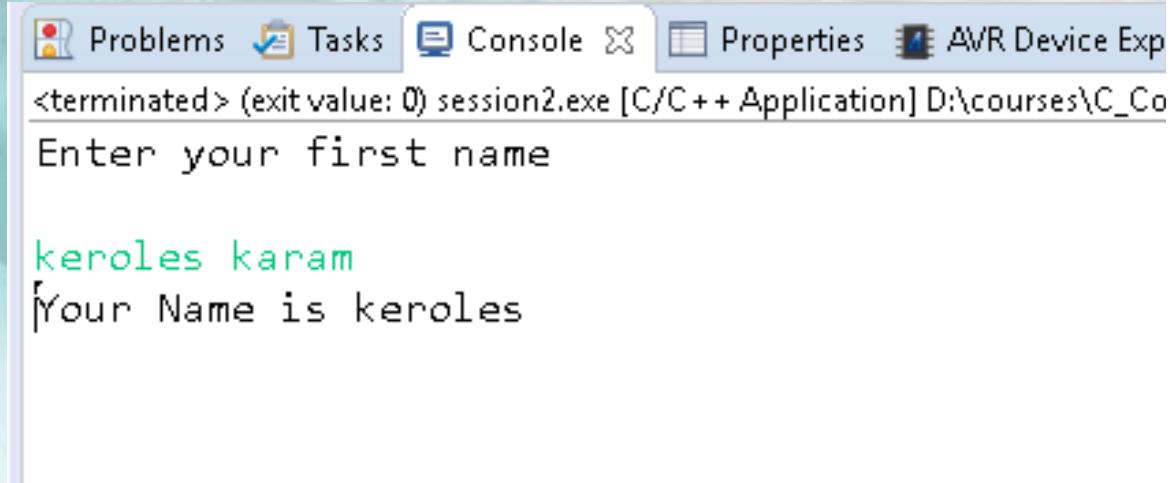
A screenshot of a code editor showing a file named "main.c". The code is as follows:

```
3 *  
4 * Created on: Mar 23, 2017  
5 * Author: Keroles  
6 */  
7 #include <stdio.h>  
8  
9 int main()  
10 {  
11     char text[100] = {'h', 'e', 'l', 'l', 'o'};  
12     printf("%s\r\n", text);  
13 }  
14  
15
```



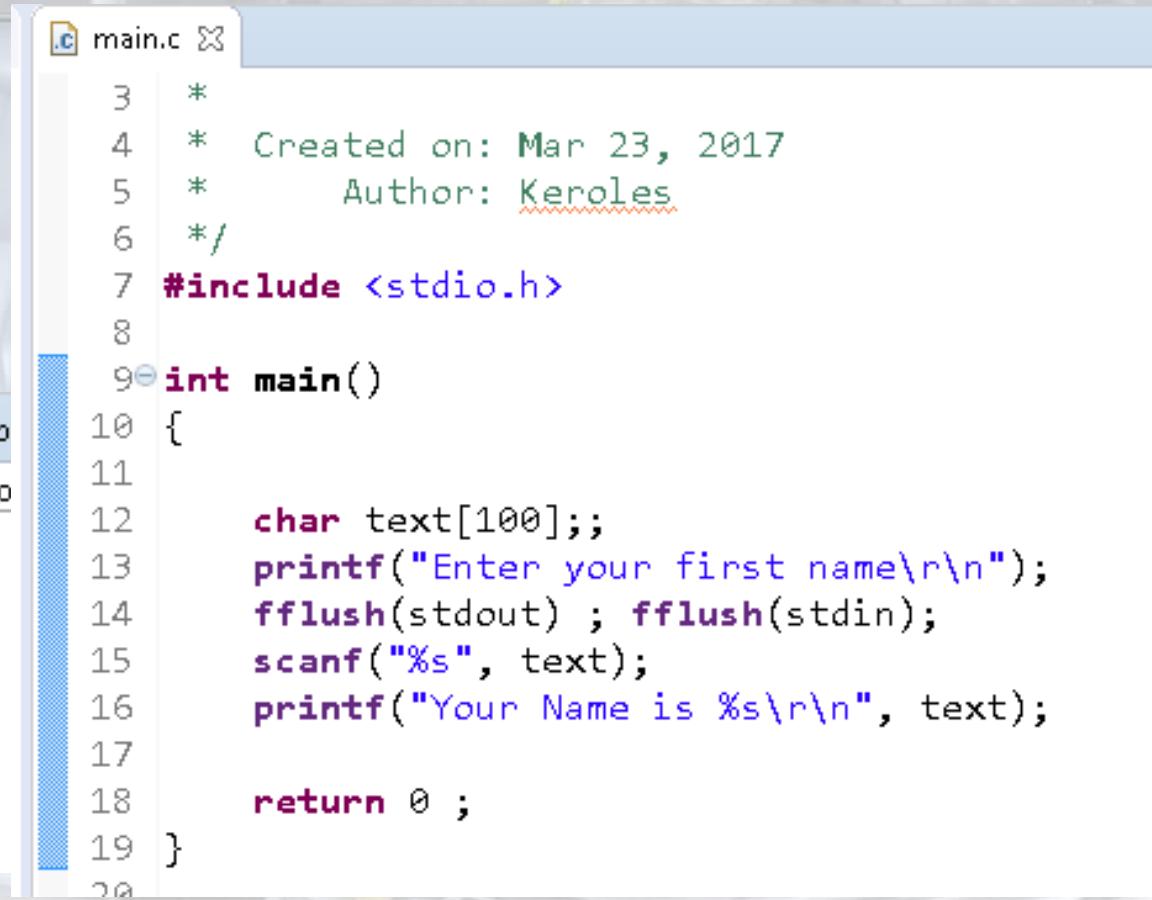
Scanning String Value

- ▶ **scanf** is used with “%s” to scan string input from keyboard. However there is a problem?
- ▶ **scanf** takes only the first word in the input text and leave the rest.



```

Problems Tasks Console ✘ Properties AVR Device Exp
<terminated> (exit value: 0) session2.exe [C/C++ Application] D:\courses\C_Co
Enter your first name
keroles karam
Your Name is keroles
  
```

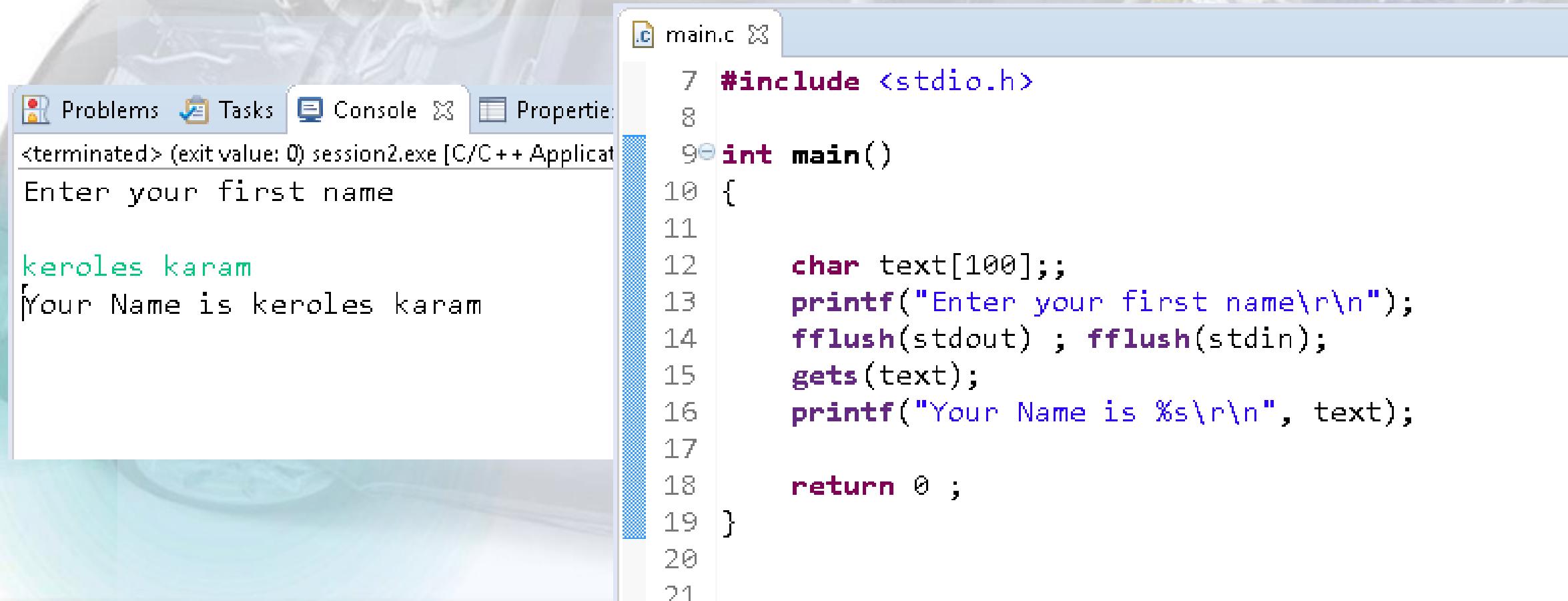


```

main.c ✘
 3  /*
 4  *   Created on: Mar 23, 2017
 5  *       Author: Keroles
 6  */
 7 #include <stdio.h>
 8
 9 int main()
10 {
11
12     char text[100];
13     printf("Enter your first name\r\n");
14     fflush(stdout) ; fflush(stdin);
15     scanf("%s", text);
16     printf("Your Name is %s\r\n", text);
17
18
19 }
20
  
```



C language provides a solution for above problem using gets function



The screenshot shows a C/C++ IDE interface. On the left, there are tabs for 'Problems', 'Tasks', 'Console' (which is active), and 'Properties'. The console output shows:

```
<terminated> (exit value: 0) session2.exe [C/C++ Application]
Enter your first name
keroles karam
Your Name is keroles karam
```

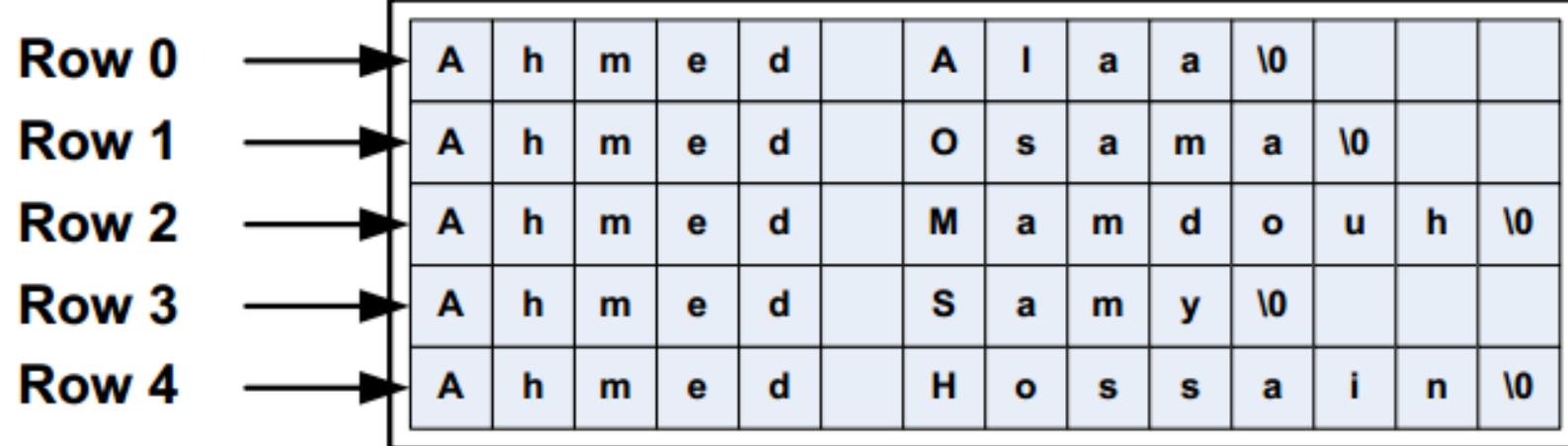
The main.c file contains the following code:

```
7 #include <stdio.h>
8
9 int main()
10 {
11
12     char text[100];
13     printf("Enter your first name\r\n");
14     fflush(stdout) ; fflush(stdin);
15     gets(text);
16     printf("Your Name is %s\r\n", text);
17
18     return 0 ;
19 }
20
21
```



Array of Strings

```
char names[5][14] = {"Ahmed Alaa", "Ahmed Osama", "Ahmed MAMDouh",
                     "Ahmed Samy", "Ahmed Hossain"};
```

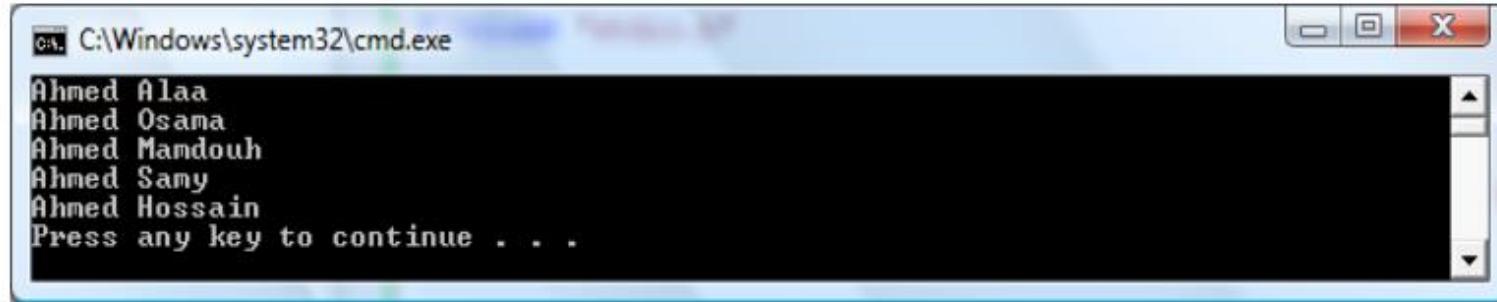


Printing Array of Strings

```
#include "stdio.h"

void main()
{
    char names[5][14] = {"Ahmed Alaa", "Ahmed Osama", "Ahmed
Mamdouh", "Ahmed Samy", "Ahmed Hossain"};
    int i;

    for(i=0; i<5; i++)
        printf("%s\r\n", names[i]);
}
```



A screenshot of a Windows command prompt window titled 'cmd C:\Windows\system32\cmd.exe'. The window displays five lines of text: 'Ahmed Alaa', 'Ahmed Osama', 'Ahmed Mandouh', 'Ahmed Samy', and 'Ahmed Hossain'. Below these lines, the text 'Press any key to continue . . .' is visible at the bottom of the window.

m/groups/embedded.system.KS/

Copy String to String

```
#include "stdio.h"

void main()
{
    char a[20] = "Alaa Ezzat";
    char b[20];
    int i = 0;

    while(a[i] != 0)
    {
        b[i] = a[i];
        i++;
    }
    b[i] = 0; //Add null termination to the end of B

    printf("%s\r\n", b);
}
```

There is another solution to above problem using **strcpy** function. **strcpy** takes both the destination and the source strings and performs the coping operation internally.

[v.facebook.com/groups/embedded.system.KS/](https://www.facebook.com/groups/embedded.system.KS/)

strcpy

```
#include "stdio.h"
#include "string.h"

void main()
{
    char a[20] = "Alaa Ezzat";
    char b[20];
    strcpy(b, a);
    printf("%s\r\n", b);
}
```

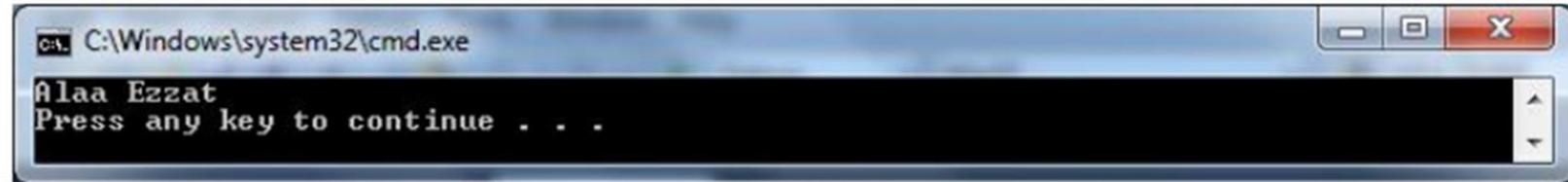


Adding String to String

String addition means concatenating the second string characters at the end of the first string.
 This can be made using **strcat** function.

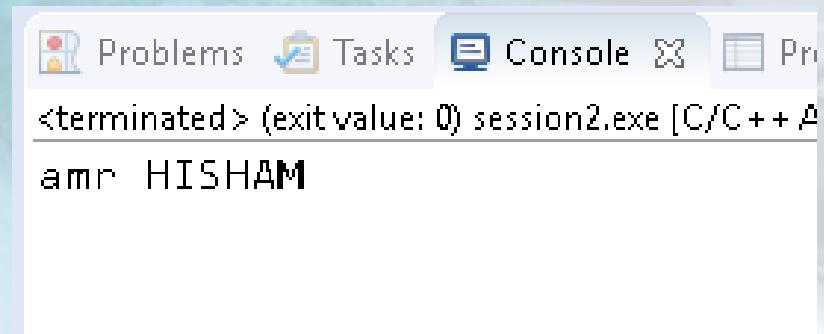
```
#include "stdio.h"
#include "string.h"

void main()
{
    char a[20] = "Alaa";
    char b[20] = "Ezzat";
    strcat(a, " ");
    strcat(a, b);
    printf("%s\r\n", a);
}
```

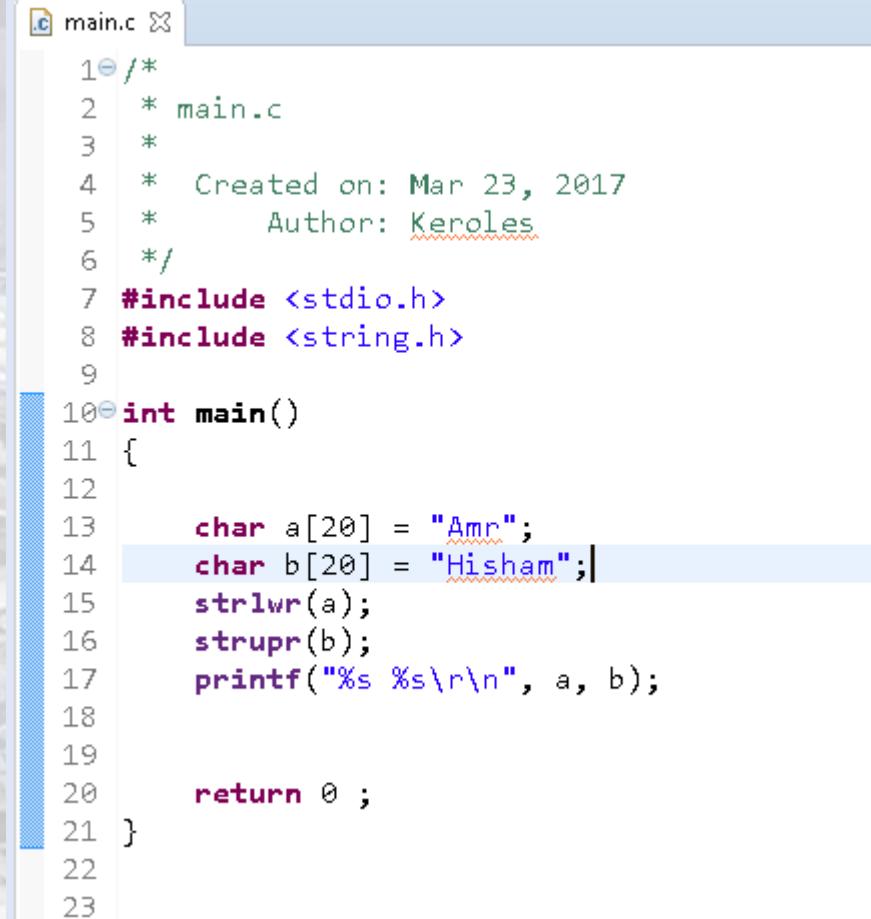


Changing String Case

- ▶ **strlwr** function changes all string letters to the lower case.
Ex: “AhMed” ↘ “ahmed”
- ▶ **strupr** function change all string letters to the upper case.
Ex: “aHmeD” ↘ “AHMED”



```
Problems Tasks Console Problems
<terminated> (exit value: 0) session2.exe [C/C++ A
amr HISHAM
```



```
main.c ✘
1  /*
2  * main.c
3  *
4  * Created on: Mar 23, 2017
5  * Author: Keroles
6  */
7 #include <stdio.h>
8 #include <string.h>
9
10 int main()
11 {
12
13     char a[20] = "Amr";
14     char b[20] = "Hisham";
15     strlwr(a);
16    strupr(b);
17     printf("%s %s\r\n", a, b);
18
19
20
21
22
23 }
```

<https://www.facebook.com/groups/embedded.system.KS/>



Finding the String Length

strlen function calculates the given string length, it simply counts the number of letters until the null termination.

```
#include "stdio.h"
#include "string.h"

void main()
{
    char name[] = "Ahmed Said";
    printf("Name: %s, Length: %d\r\n", name, strlen(name));
}
```



Comparing Two Strings

- ▶ **strcmp** function compares two strings and produces one of three results:
 - ▶ if the two string are identical it gives 0
 - ▶ if the first string is lower in the alphabetic order it gives -1
 - ▶ if the second string is higher in the alphabetic order it gives 1
 - “ahmed” and “ahmed” ↴ 0
 - “ahmed” and “amgad” ↴ -1 because the second letter „h“ is less than „m“
 - “maged” and “aya” ↴ 1 because the first letter „m“ is greater than „a“

strcmp differentiate between capital and small letters which means “MAGED” is less than “ahmed”, because the ASCII code of the capital letters is lower than the ASCII code of the small letters. To solve this problem you can change the case of both strings to the same case then use strcmp function. Alternatively you can use **stricmp** function which performs the comparison independent on the string case



```
#include "stdio.h"
#include "string.h"

void main()
{
    char names[5][14] = {"Alaa", "Osama", "Mamdouh", "Samy",
    "Hossain"};
    char name[14];
    int i;

    printf("Enter your name : ");
    scanf("%s", name);

    for(i=0;i<5;i++)
    {
        if(strcmp(name, names[i])==0)
        {
            printf("Congratulation,
                    your name is in the list");
            break;
        }
    }

    if(i==5)
        printf("We are sorry, your name is not listed");
}
```



Converting String to Integer Value

35

```
#include "stdio.h"
#include "string.h"

void main()
{
    char text[20] = "1025";
    int x = 200;
    int y;

    y = x + text; //wrong
}
```

The variable text contains 4 ASCII letters „1“, „0“, „2“, „5“. The statement ($x + \text{text}$) is completely wrong, because text is string and not a number, computer cannot understand string contents directly.

atoi function helps computer to convert string to a number of type (**int**) if it is applicable otherwise it gives zero, for example:

“1025” → 1025

“120KG” → 120

“The Cost is 30” → 0

<https://www.facebook.com/groups/embedded.system.KS/>



Converting String to Integer Value

```
#include "stdio.h"
#include "stdlib.h"

void main()
{
    char text[20];
    int x;
    int y;
    int z;

    printf("Enter x : ");
    gets(text);
    x = atoi(text);

    printf("Enter y : ");
    gets(text);
    y = atoi(text);

    z = x + y;
    printf("x + y = %d\r\n", z);
}
```

<com/groups/embedded.system.KS/>



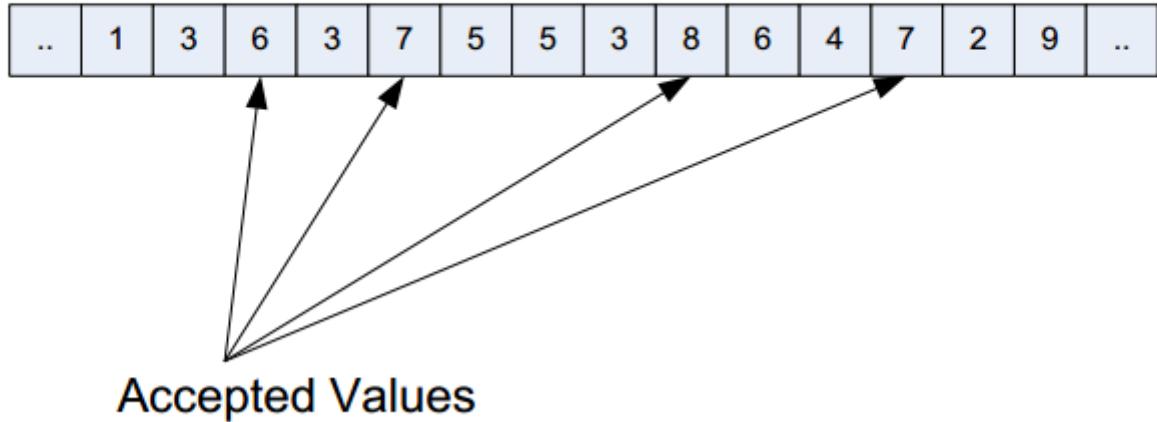
Converting String to Real Value

- ▶ **atof** function helps computer to convert string to a number of type (float) if it is applicable otherwise it gives zero, for example:
“10.25” > 10.25
“1.2KG” > 1.2
“The Cost is 3.5” > 0



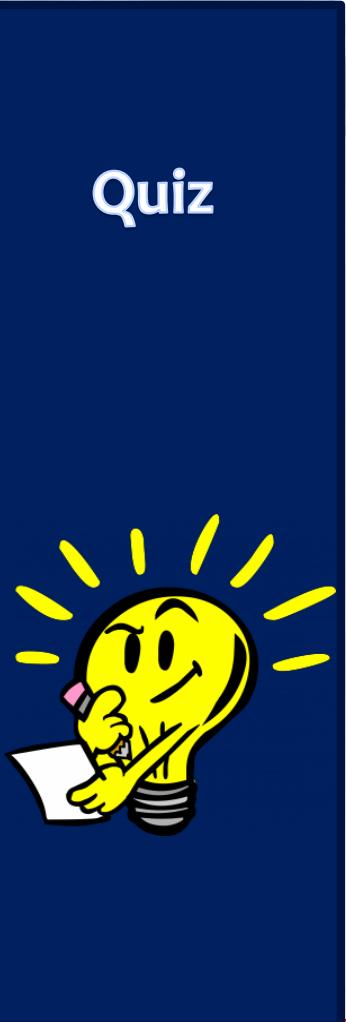
Define a 1D array of 1000 element and fill it with a random values (using rand() function) between 1 and 1000, then searches and counts the number of values that satisfy the following condition:

The previous and the next values is lower than the center value



```
.c main.c
7 #include <stdio.h>
8 #include <string.h>
9
10 int main()
11 {
12
13     int x[1000];
14     int i ;
15     for (i=0; i<1000;i++)
16         x[i]=rand();
17
18     return 0 ;
19 }
```

Variables		
Name	Type	Value
x	int [1000]	0x61ef8c
[0..99]	int [100]	0x61ef8c
(*) x[0]	int	41
(*) x[1]	int	18467
(*) x[2]	int	6334
(*) x[3]	int	26500
(*) x[4]	int	19169
(*) x[5]	int	15724
(*) x[6]	int	11478
(*) x[7]	int	29358
(*) x[8]	int	26962
(*) x[9]	int	24464
(*) x[10]	int	5705
(*) x[11]	int	28145
(*) x[12]	int	23281
(*) x[13]	int	16827
(*) x[14]	int	10512288
(*) x[15]	int	0



<https://www.facebook.com/groups/embedded.system.KS/>



eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Thank you ☺

KEROLES SHENOUDA

