



# **ASSIGNMENT 1**

## **AI & ML**

**Dr. Layla Abo Hadid**

---

**Asmaa Gamal Abdel-Halem Mabrouk Nagy**  
**Communication & Electronics**  
**15010473**

**أسماء جمال عبد الحليم مبروك ناجي**

---

**Lab 1**  
**Pointers & Arrays**



# Table of contents

1. Assumptions & Other Details About Pointers & Arrays.
2. The code
3. Sample runs

## Assumptions & Other Details About Points & Arrays.

### Arrays in C:

An array in C/C++ or be it in any programming language is a collection of similar data items stored at contiguous memory locations and elements can be accessed randomly using indices of an array. They can be used to store collection of primitive data types such as int, float, double, char, etc of any particular type. To add to it, an array in C/C++ can store derived data types such as the structures, pointers etc. Given below is the picture representation of an array.

40	55	63	17	22	68	89	97	89
0	1	2	3	4	5	6	7	8

<- Array Indices

**Array Length = 9**

**First Index = 0**

**Last Index = 8**

### Pointers in C/C++ with Examples

Pointers are symbolic representation of addresses. They enable programs to simulate call-by-reference as well as to create and manipulate dynamic data structures. It's general declaration in C/C++ has the format:

Syntax:

```
datatype *var_name;
```

```
int *ptr; //ptr can point to an address which holds int data
```

# The Code

## 1. Problem 1 – Arithmetic Operations:

```
# Asmaa Gamal
#Assignment 1
# Electronics & Communications department
1#include <stdio.h>
2#include <stdlib.h>
3
4
5#define ARRSIZE 5
6int main()
7{
8    int counter_pos=0 ,counter_neg=0;
9    float sum_pos =0 ,sum_neg=0;
10
11    float positive[ARRSIZE]={0,0,0,0,0};
12    float negative[ARRSIZE]={0,0,0,0,0};
13
14    float nums[ARRSIZE];
15
16    printf("Enter 5 numbers:\n");
17    for (int i=0;i<ARRSIZE; i++)
18    {
19
20        scanf("%f",&nums[i]);
21
22        if(nums[i]>0)
23        {
24            positive[i]=nums[i];
25            sum_pos+=positive[i];
26            counter_pos++;
27
28        }else if(nums[i]==0){
29            printf("u have just entered a zero!\n");
30
31        }else{
32            negative[i]= nums[i];
33            sum_neg+=negative[i];
34            counter_neg++;
35        }
36    }
37
38    printf("Number of positive numbers:%d \n",counter_pos);
39    printf("Number of negative numbers:%d \n",counter_neg);
40
41    printf("Average of positive numbers:%f \n", sum_pos/counter_pos);
42    printf("Average of negative numbers:%f \n", sum_neg/counter_neg);
43
44    return 0;
45 }
```

## 2. Problem 2 - Sales People:

```
# Asmaa Gamal
#Assignment 1
# Electronics & Communications department
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define SIZE 3
5 void add(int*matrix_1,int*matrix_2, int*sum);
6 void print_sum (int* sum);
7 int main()
8 {
9 int matrix_1[SIZE][SIZE], matrix_2[SIZE][SIZE];
10 int sum[SIZE][SIZE]={0};
11
12 printf("Enter ur input matrix_1 of size 3x3: \n");
13 for(int i=0; i<3; i++)
14 {
15 for(int j=0;j<3;j++) {
16 scanf("%d", &matrix_1[i][j]);
17 }
18 printf("\n");
19 }
20
21
22 printf("Enter ur input matrix_2 of size 3x3:\n");
23 for(int i=0; i<3; i++)
24 {
25 for(int j=0;j<3;j++) {
26 scanf("%d", &matrix_2[i][j]);
27 }
28 printf("\n");
29 }
30
31
32 add(matrix_1,matrix_2,sum);
33 printf("sum of both matrices: \n");
34 print_sum(sum);
35
36 return 0;
37 }
38
39
40 void add (int*matrix_1,int*matrix_2, int*sum)
41 {
42 for(int i=0; i<3; i++)
43 {
44 for(int j=0;j<3;j++)
45 {
46 *((sum+i)+j)=*((matrix_1+i)+j) + *((matrix_2+i)+j);
47 }
48 printf("\n");
49 }
50
51 }
52
53 void print_sum (int* sum){
54 for(int i=0; i<3; i++)
55 {
56 for(int j=0;j<3;j++)
57 {
58 printf( "%d ", *((sum+i)+j) );
59 }
60
61 printf("\n");
62 }
63 }
```

### 3. Problem 3 - Add two matrix using pointers:

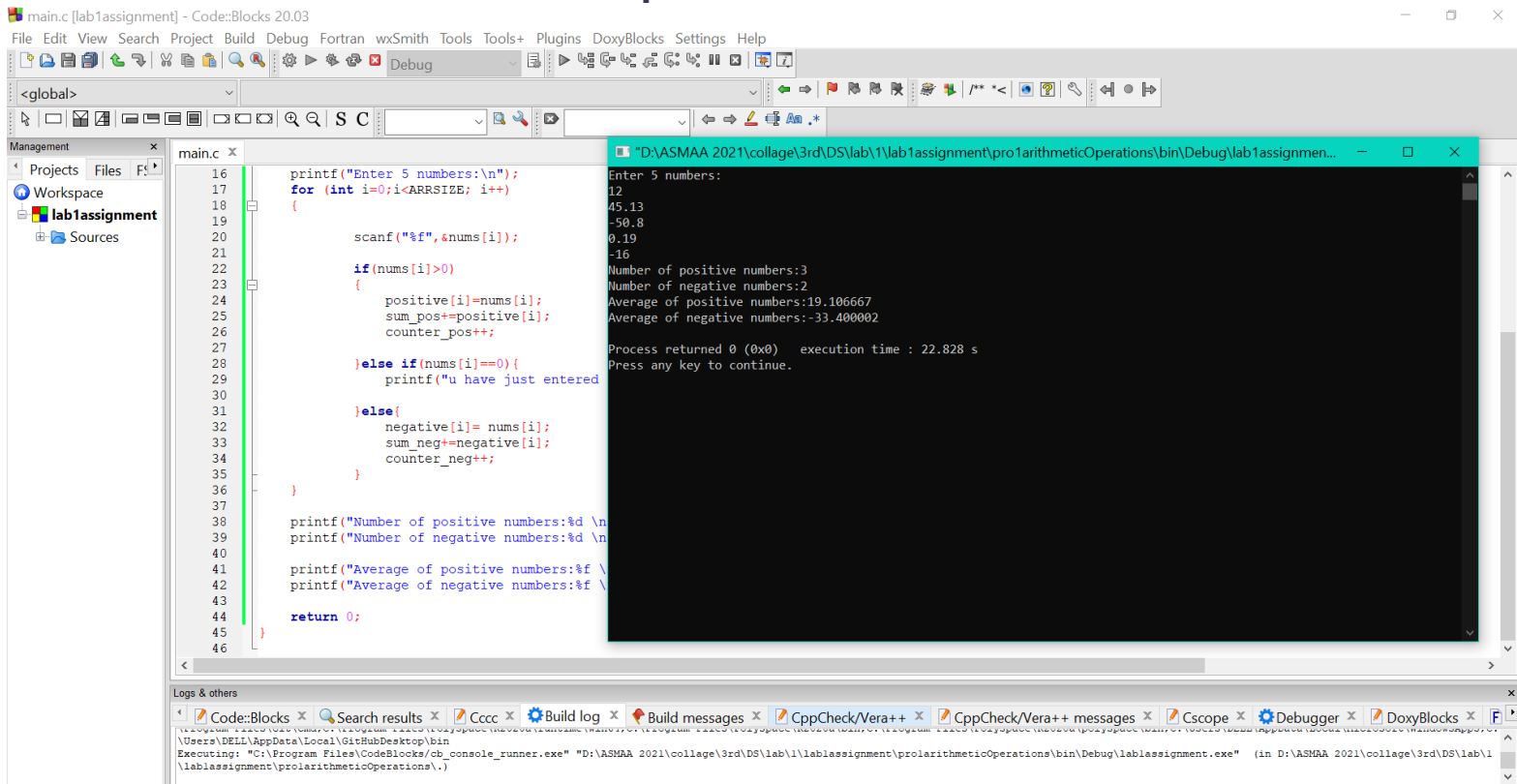
```
# Asmaa Gamal
#Assignment 1
# Electronics & Communications department
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 #define EMPLOYERS 10^6
6 int main()
7 {
8     int out_salary=0;
9     int in_gross[EMPLOYERS];
10
11     int i,
range2=0, range3=0, range4=0, range5=0, range6=0, range7=0, range8=0, range9=0,
range10=0;
12     for(i=0; i<=EMPLOYERS ; i++)
13     {
14         printf("Enter Employee gross sale (-1 to end) :");
15         scanf("%d",&in_gross[i]);
16         if (in_gross[i]==-1)
17         {
18             break;
19         }
20     }
21
22     out_salary=200+(0.09* in_gross[i]);
23     printf("Employee salary is:%d\n",out_salary);
24     if(out_salary>= 200 && out_salary<=299 )
25     {
26         range2++;
27     }
28     }else if(out_salary>= 300 && out_salary<=399 )
29     {
30         range3++;
31     }
32     }else if(out_salary>= 400 && out_salary<=499 )
33     {
34         range4++;
35     }
36     }else if(out_salary>= 500 && out_salary<=599 )
37     {
38         range5++;
39     }
40     }else if(out_salary>= 600 && out_salary<=699 )
41     {
42         range6++;
43     }
44     }else if(out_salary>= 700 && out_salary<=799 )
45     {
46         range7++;
47     }
48     }else if(out_salary>= 800 && out_salary<=899 )
```

```
49 {
50 range8++;
51
52 }else if(out_salary>= 900 && out_salary<=1000 )
53 {
54 range9++;
55
56 }else if(out_salary>= 1000 )
57 {
58 range10++;
59
60 }
61
62
63 }
64
65 printf("Total %d Employees Reported \n",i);
66 printf("Employees in the range: \n");
67 printf("200 299: %d\n",range2);
68 printf("300 399: %d\n",range3);
69 printf("400 499: %d\n",range4);
70 printf("500 599: %d\n",range5);
71 printf("600 699: %d\n",range6);
72 printf("700 799: %d\n",range7);
73 printf("800 899: %d\n",range8);
74 printf("900 999: %d\n",range9);
75 printf("Over 1000: %d\n",range10);
76
77
78
79 return 0;
80 }
```

# Screenshots Of Some Runs

## 1. Problem 1 – Arithmetic Operations:



The screenshot shows the Code::Blocks IDE with the source code for 'main.c' and a terminal window. The source code calculates the sum and average of positive and negative numbers from an array of 5 elements. The terminal window shows the program's output, including the input numbers, the count of positive and negative numbers, and their respective averages. The program returns 0 and the execution time is 22.828 s.

```
main.c [lab1assignment] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
main.c x
16
17 printf("Enter 5 numbers:\n");
18 for (int i=0; i<ARRSIZE; i++)
19 {
20     scanf("%f", &nums[i]);
21
22     if (nums[i]>0)
23     {
24         positive[i]=nums[i];
25         sum_pos+=positive[i];
26         counter_pos++;
27
28     } else if (nums[i]==0) {
29         printf("u have just entered
30
31     } else {
32         negative[i]= nums[i];
33         sum_neg+=negative[i];
34         counter_neg++;
35     }
36
37
38 printf("Number of positive numbers:%d \n");
39 printf("Number of negative numbers:%d \n");
40
41 printf("Average of positive numbers:%f \n");
42 printf("Average of negative numbers:%f \n");
43
44 return 0;
45
46
"D:\ASMAA 2021\collage\3rd\DS\lab\1\lab1assignment\pro1arithmeticOperations\bin\Debug\lab1assignment.exe"
Enter 5 numbers:
12
45.13
-50.8
0.19
-16
Number of positive numbers:3
Number of negative numbers:2
Average of positive numbers:19.106667
Average of negative numbers:-33.400002
Process returned 0 (0x0) execution time : 22.828 s
Press any key to continue.
```

"D:\ASMAA 2021\collage\3rd\DS\lab\1\lab1assignment\pro1

Enter 5 numbers:

12

45.13

-50.8

0.19

-16

Number of positive numbers:3

Number of negative numbers:2

Average of positive numbers:19.106667

Average of negative numbers:-33.400002

Process returned 0 (0x0) execution time : 22.828 s

Press any key to continue.

## 2. Problem 2 - Sales People:

```
main.c [lab1assignment2] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global> main() : int
D:\ASMAA 2021\collage\3rd\DS\lab\1\lab1assignment2\bin\Debug\lab1assignment2.exe
Enter Employee gross sale (-1 to end) :3000
Employee salary is:470
Enter Employee gross sale (-1 to end) :1000
Employee salary is:290
Enter Employee gross sale (-1 to end) :10000
Employee salary is:1100
Enter Employee gross sale (-1 to end) :8000
Employee salary is:920
Enter Employee gross sale (-1 to end) :200
Employee salary is:218
Enter Employee gross sale (-1 to end) :7000
Employee salary is:830
Enter Employee gross sale (-1 to end) :-1
Total 6 Employees Reported
printf("Total %d Employees in the range:
printf("Employees in the range:
200 299: 2
printf("200 299: %d\n", 2)
300 399: 0
printf("300 399: %d\n", 0)
400 499: 1
printf("400 499: %d\n", 1)
500 599: 0
printf("500 599: %d\n", 0)
600 699: 0
printf("600 699: %d\n", 0)
700 799: 0
printf("700 799: %d\n", 0)
800 899: 1
printf("800 899: %d\n", 1)
900 999: 1
printf("900 999: %d\n", 1)
Over 1000: 1
printf("Over 1000: %d\n", 1)
Process returned 0 (0x0) execution time : 33.815 s
Press any key to continue.
return 0;
```

```
"D:\ASMAA 2021\collage\3rd\DS\lab\1\lab1assignment2\bin\Debug\lab1assignment2.exe"
Enter Employee gross sale (-1 to end) :3000
Employee salary is:470
Enter Employee gross sale (-1 to end) :1000
Employee salary is:290
Enter Employee gross sale (-1 to end) :10000
Employee salary is:1100
Enter Employee gross sale (-1 to end) :8000
Employee salary is:920
Enter Employee gross sale (-1 to end) :200
Employee salary is:218
Enter Employee gross sale (-1 to end) :7000
Employee salary is:830
Enter Employee gross sale (-1 to end) :-1
Total 6 Employees Reported
Employees in the range:
200 299: 2
300 399: 0
400 499: 1
500 599: 0
600 699: 0
700 799: 0
800 899: 1
900 999: 1
Over 1000: 1
Process returned 0 (0x0) execution time : 33.815 s
Press any key to continue.
```



### 3. Problem 3 - Add two matrix using pointers:

main.c [lab1assignment3] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Debug

<global> print\_sum(int\* sum) : void

Management

Projects Files

Workspace

lab1assignment3

Sources

main.c

```
13 for(int i=0; i<3; i++)
14 {
15     for(int j=0; j<3; j++) {
16         scanf("%d", &matrix_1[i][j]);
17     }
18     printf("\n");
19 }
20
21
22 printf("Enter ur input matrix_2 of size 3x3:\n");
23 for(int i=0; i<3; i++)
24 {
25     for(int j=0; j<3; j++) {
26         scanf("%d", &matrix_2[i][j]);
27     }
28     printf("\n");
29 }
30
31
32 add(matrix_1, matrix_2, sum);
33 printf("sum of both matrices: \n");
34 print_sum(sum);
35
36 return 0;
37
38
39
40 void add (int*matrix_1, int*matrix_2, int*sum)
41 {
42     for(int i=0; i<3; i++)
```

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

"D:\ASMAA 2021\collage\3rd\DS\lab\1\lab1assignment3\bin\...

Enter ur input matrix\_1 of size 3x3:

1

2

3

4

5

6

7

8

9

Enter ur input matrix\_2 of size 3x3:

9

8

7

6

5

4

3

2

1

sum of both matrices:

10 10 10

10 10 10

10 10 10

Process returned 0 (0x0) execution time : 42.406 s

Press any key to continue.

DoxyBlocks

Microsoft Windows Apps; C:

lab1assignment3\.)

Show comments in the diagram

C/C++

Windows (CR+LF)

"D:\ASMAA 2021\collage\3rd\DS\lab\1\lab1assignment3\bin\...

Enter ur input matrix\_1 of size 3x3:

1

2

3

4

5

6

7

8

9

Enter ur input matrix\_2 of size 3x3:

9

8

7

6

5

4

3

2

1

sum of both matrices:

10 10 10

10 10 10

10 10 10

Process returned 0 (0x0) execution time : 42.406 s

Press any key to continue.