

# **LAB 1**

NAME : محمد عبدالسلام عبدالمطلب

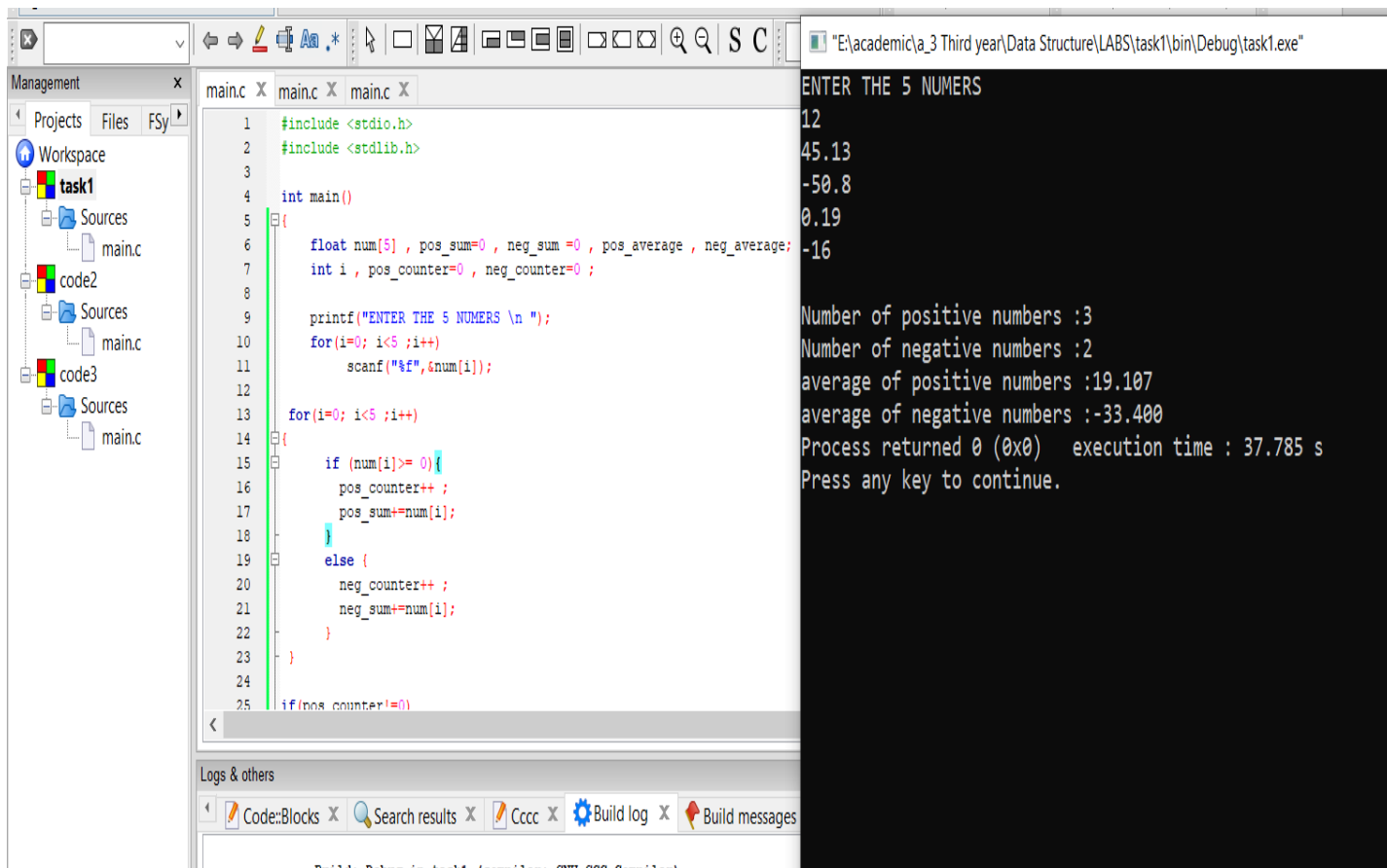
ID : 18011535

**COMMUNICATION DEPARTMENT**

**( 3<sup>RD</sup> YEAR )**

## Code 1:

- 1- If user Enter all positive or all negative average will be printed zero
- 2- I use two variables neg\_sum and pos\_sum to calculate the average by  $avr = \text{sum}/\text{num}$



The screenshot shows a C program in a code editor with a project explorer on the left and a console window on the right. The code calculates the average of positive and negative numbers from a user input of 5 numbers.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      float num[5], pos_sum=0, neg_sum=0, pos_average, neg_average;
7      int i, pos_counter=0, neg_counter=0;
8
9      printf("ENTER THE 5 NUMERS \n ");
10     for(i=0; i<5; i++)
11         scanf("%f", &num[i]);
12
13     for(i=0; i<5; i++)
14     {
15         if (num[i] >= 0) {
16             pos_counter++;
17             pos_sum+=num[i];
18         }
19         else {
20             neg_counter++;
21             neg_sum+=num[i];
22         }
23     }
24
25     if(pos_counter!=0)
```

The console window shows the following output:

```
ENTER THE 5 NUMERS
12
45.13
-50.8
0.19
-16

Number of positive numbers :3
Number of negative numbers :2
average of positive numbers :19.107
average of negative numbers : -33.400
Process returned 0 (0x0)   execution time : 37.785 s
Press any key to continue.
```

## Code 2:

- 1- I use a fixed array sales\_people to count salary in range.
- 2- I will take any number until -1.

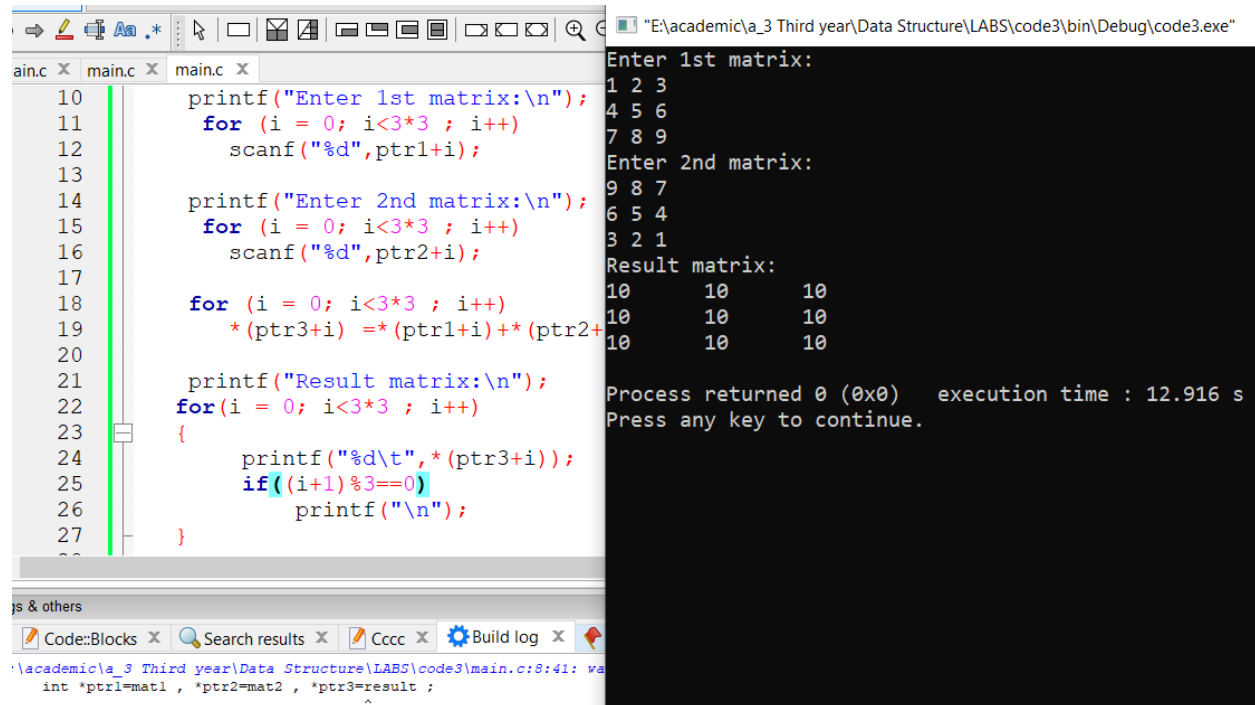
The screenshot displays the Code::Blocks IDE with a C program open in the editor. The program is designed to count the number of employees within specific salary ranges. The code uses an array named 'sales\_people' to store the counts for each range. The salary ranges are defined by the 'gross' variable in the output, which is calculated as 'gross = (-1 to end)'. The counts are as follows:

Salary Range (gross)	Count
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

The total number of employees reported is 6. The program returned 0 and took 18.620 seconds to execute. The IDE interface shows the project 'code2' and the file 'main.c' in the workspace. The 'Logs & others' panel at the bottom shows the build process details.

## Code 3:

I use pointer only.



The screenshot shows the Code::Blocks IDE with a C program in `main.c` and its execution output. The program uses pointers to add two 3x3 matrices. The first matrix is entered as 1 2 3, 4 5 6, 7 8 9. The second matrix is entered as 9 8 7, 6 5 4, 3 2 1. The resulting matrix is 10 10 10, 10 10 10, 10 10 10. The execution time is 12.916 s.

```
10 printf("Enter 1st matrix:\n");
11 for (i = 0; i<3*3 ; i++)
12     scanf("%d",ptr1+i);
13
14 printf("Enter 2nd matrix:\n");
15 for (i = 0; i<3*3 ; i++)
16     scanf("%d",ptr2+i);
17
18 for (i = 0; i<3*3 ; i++)
19     *(ptr3+i) =*(ptr1+i)+*(ptr2+i);
20
21 printf("Result matrix:\n");
22 for(i = 0; i<3*3 ; i++)
23 {
24     printf("%d\t",*(ptr3+i));
25     if((i+1)%3==0)
26         printf("\n");
27 }
```

Enter 1st matrix:  
1 2 3  
4 5 6  
7 8 9  
Enter 2nd matrix:  
9 8 7  
6 5 4  
3 2 1  
Result matrix:  
10 10 10  
10 10 10  
10 10 10  
Process returned 0 (0x0) execution time : 12.916 s  
Press any key to continue.

Code::Blocks Search results Cccc Build log  
.\academic\3 Third year\Data Structure\LABS\code3\main.c:8:41: va  
int \*ptr1=mat1 , \*ptr2=mat2 , \*ptr3=result ;  
^