ASSIGNMENT 1
AI & ML

Dr. Layla Abo Hadid

Asmaa Gamal Abdel-Halem Mabrouk Nagy
Communication & Electronics
15010473
أسماء جمال عبد الحليم مبروك ناجي

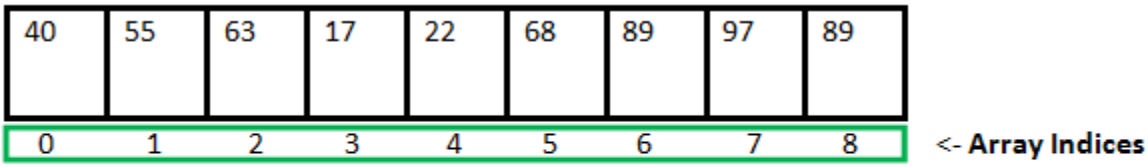
Lab 1
Pointers & Arrays

Table of contents

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.



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
6 int 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:

```c
# Asmaa Gamal
#Assignment 1
# Electronics & Communications department
#include <stdio.h>
#include <stdlib.h>

#define SIZE 3
void add(int*matrix_1,int*matrix_2, int*sum);
void print_sum (int* sum);
int main()
{
int matrix_1[SIZE][SIZE], matrix_2[SIZE][SIZE];
 int sum[SIZE][SIZE]={0};

 printf("Enter ur input matrix_1 of size 3x3: \n");
 for(int i=0; i<3; i++)
 {
 for(int j=0;j<3;j++) {
 scanf("%d", &matrix_1[i][j]);
 }
 printf("\n");
 }


 printf("Enter ur input matrix_2 of size 3x3:\n");
 for(int i=0; i<3; i++)
 {
 for(int j=0;j<3;j++) {
 scanf("%d", &matrix_2[i][j]);
 }
 printf("\n");
 }


 add(matrix_1,matrix_2,sum);
 printf("sum of both matrices: \n");
 print_sum(sum);

 return 0;
 }


 void add (int*matrix_1,int*matrix_2, int*sum)
 {
 for(int i=0; i<3; i++)
 {
 for(int j=0;j<3;j++)
 {
 *((sum+i)+j)=*((matrix_1+i)+j) + *((matrix_2+i)+j);
 }
 printf("\n");
 }

 }

 void print_sum (int* sum){
 for(int i=0; i<3; i++)
 {
 for(int j=0;j<3;j++)
 {
 printf( "%d ", *((sum+i)+j) );
 }

 printf("\n");
 }
 }
```

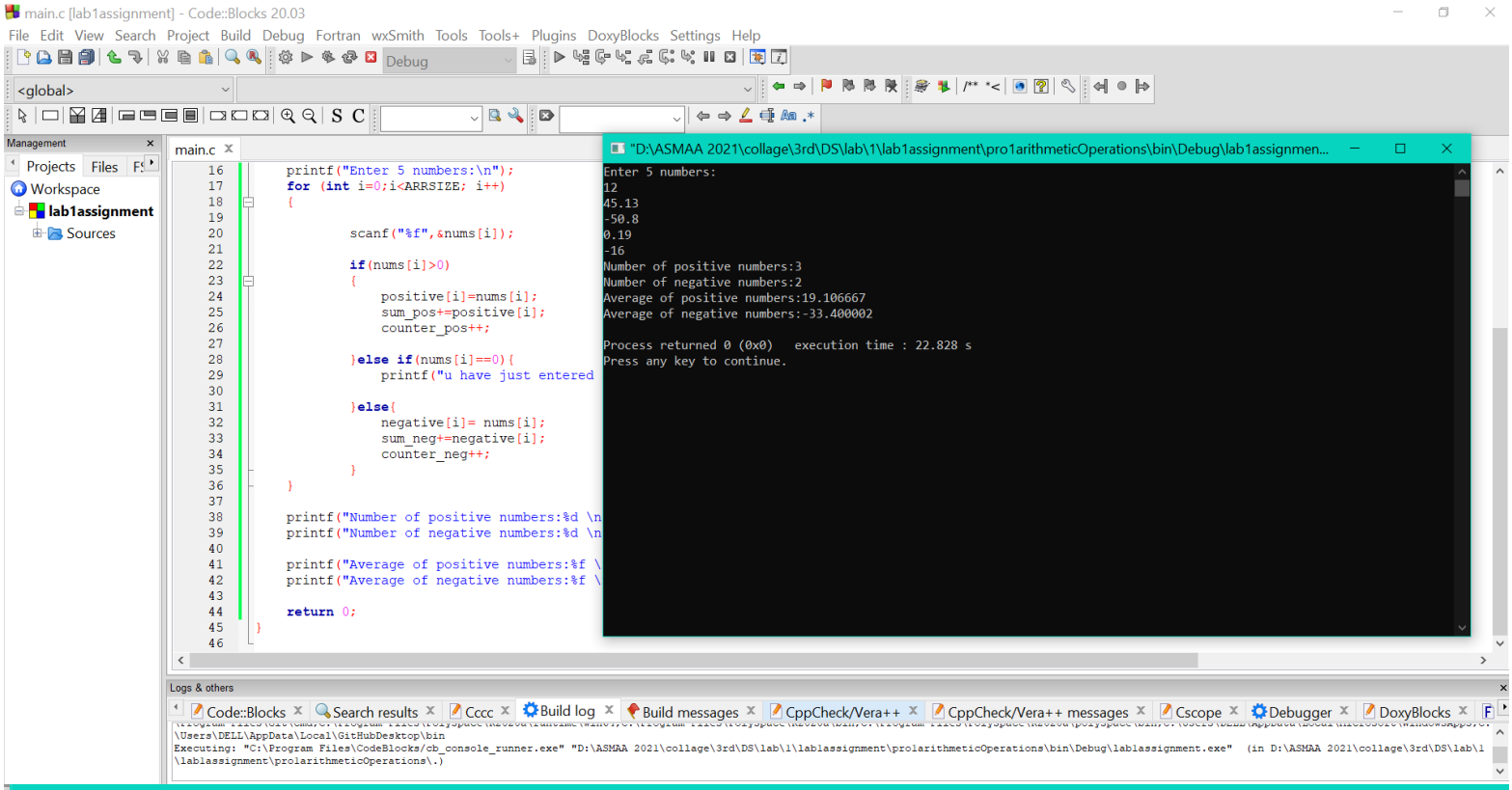
3. Problem 3 - Add two matrix using pointers:

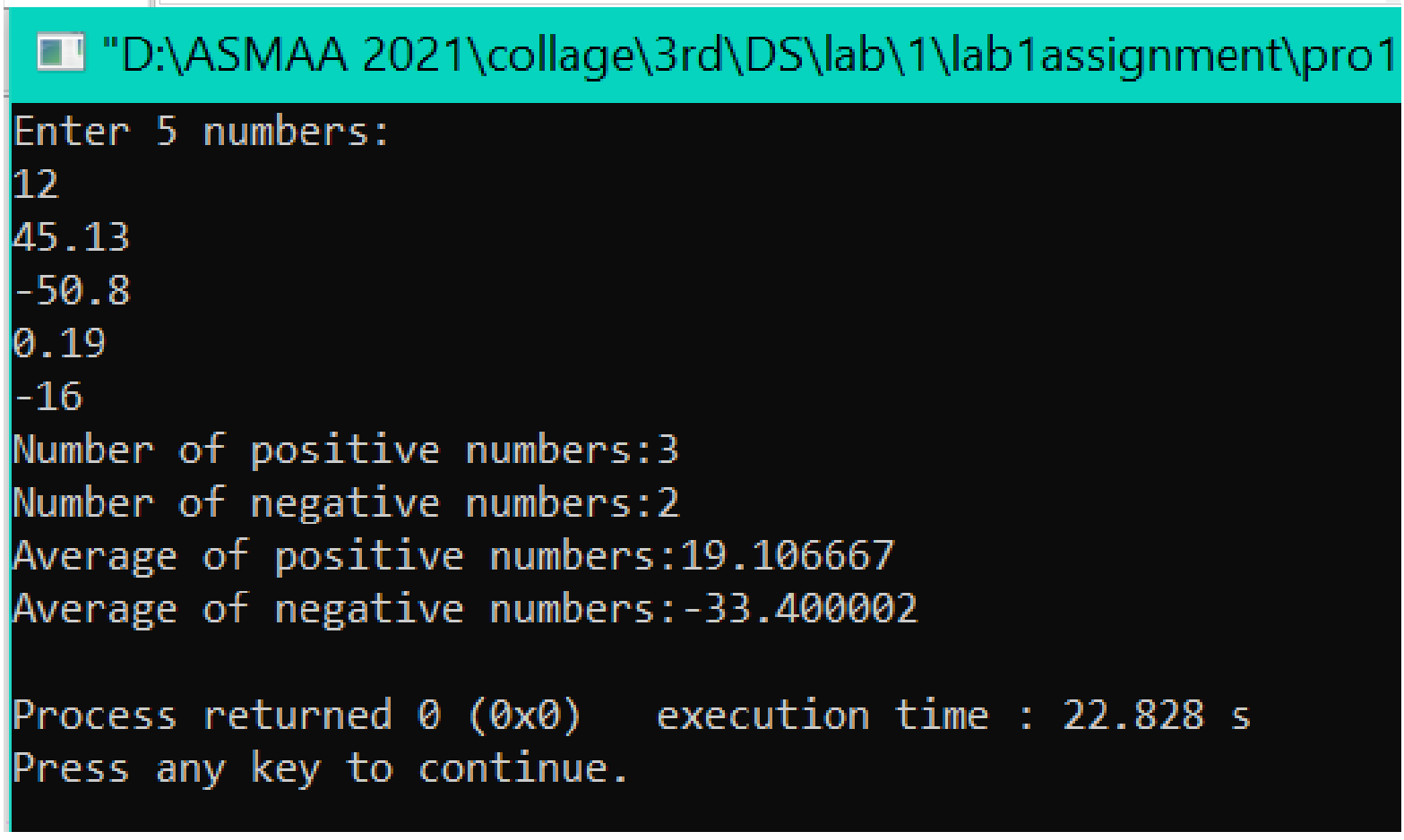
```
# Asmaa Gamal
#Assignment 1
# Electronics & Communications department


#include <stdio.h>
#include <stdlib.h>


#define EMPLOYERS 10^6
int main()
{
int out_salary=0;
int in_gross[EMPLOYERS];

 int i,
range2=0,range3=0,range4=0,range5=0,range6=0,range7=0,range8=0,range9=0,
range10=0;
 for(i=0;i<=EMPLOYERS ; i++)
 {
printf("Enter Employee gross sale (-1 to end) :");
scanf("%d",&in_gross[i]);
if (in_gross[i]==-1)
{
break;

}

out_salary=200+(0.09* in_gross[i]);
printf("Employee salary is:%d\n",out_salary);
if(out_salary>= 200 && out_salary<=299 )
{
range2++;

}else if(out_salary>= 300 && out_salary<=399 )
{
range3++;

}else if(out_salary>= 400 && out_salary<=499 )
{
range4++;

}else if(out_salary>= 500 && out_salary<=599 )
{
range5++;

}else if(out_salary>= 600 && out_salary<=699 )
{
range6++;

}else if(out_salary>= 700 && out_salary<=799 )
{
range7++;

}else if(out_salary>= 800 && out_salary<=899 )
```

```
{
range8++;

}else if(out_salary>= 900 && out_salary<=1000 )
{
range9++;

}else if(out_salary>= 1000 )
{
range10++;

}


}

printf("Total %d Employees Reported \n",i);
printf("Employees in the range: \n");
printf("200 299: %d\n",range2);
printf("300 399: %d\n",range3);
printf("400 499: %d\n",range4);
printf("500 599: %d\n",range5);
printf("600 699: %d\n",range6);
printf("700 799: %d\n",range7);
printf("800 899: %d\n",range8);
printf("900 999: %d\n",range9);
printf("Over 1000: %d\n",range10);


return 0;
}
```

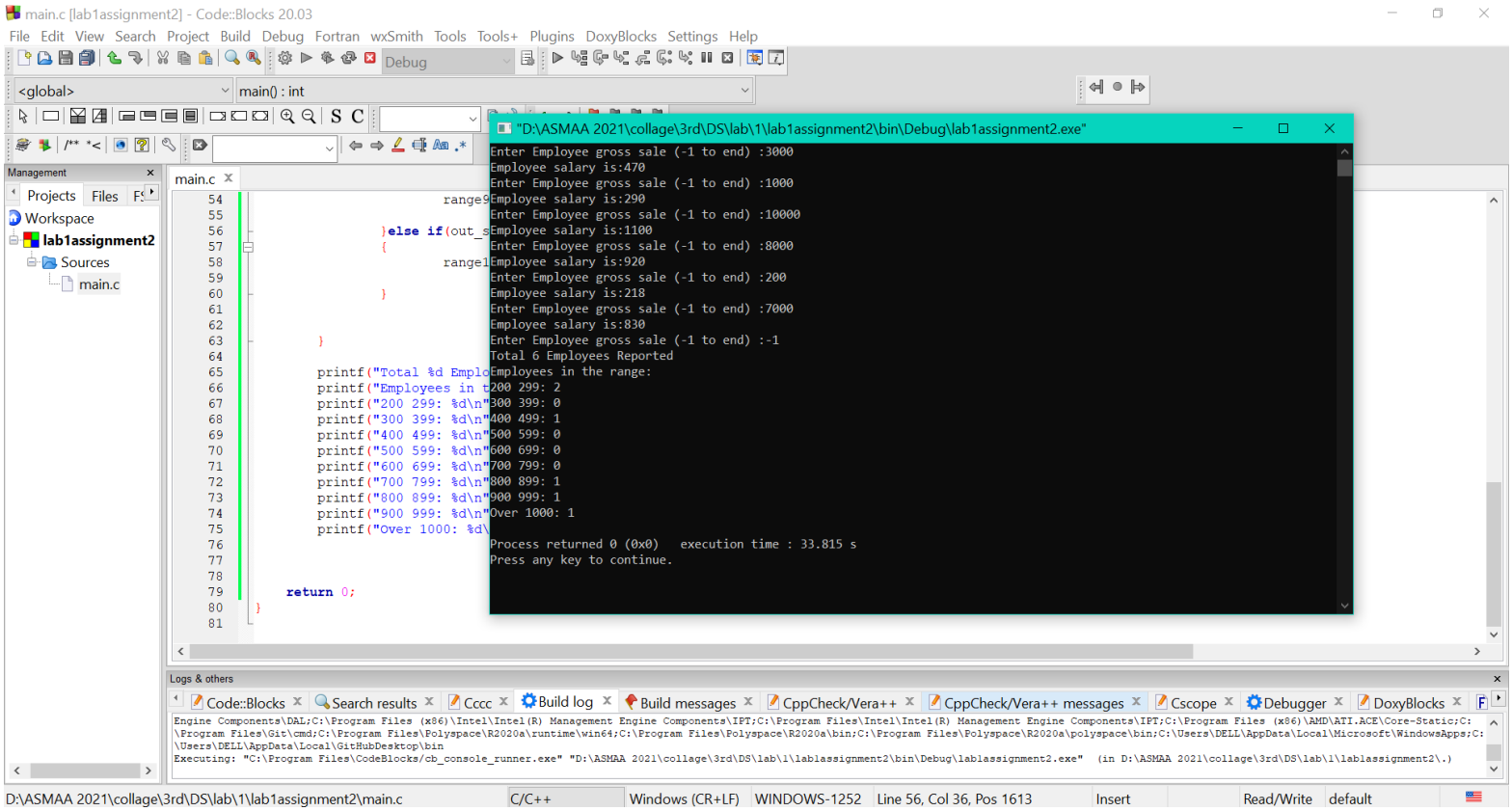
Screenshots Of Some Runs

1. Problem 1 – Arithmetic Operations:





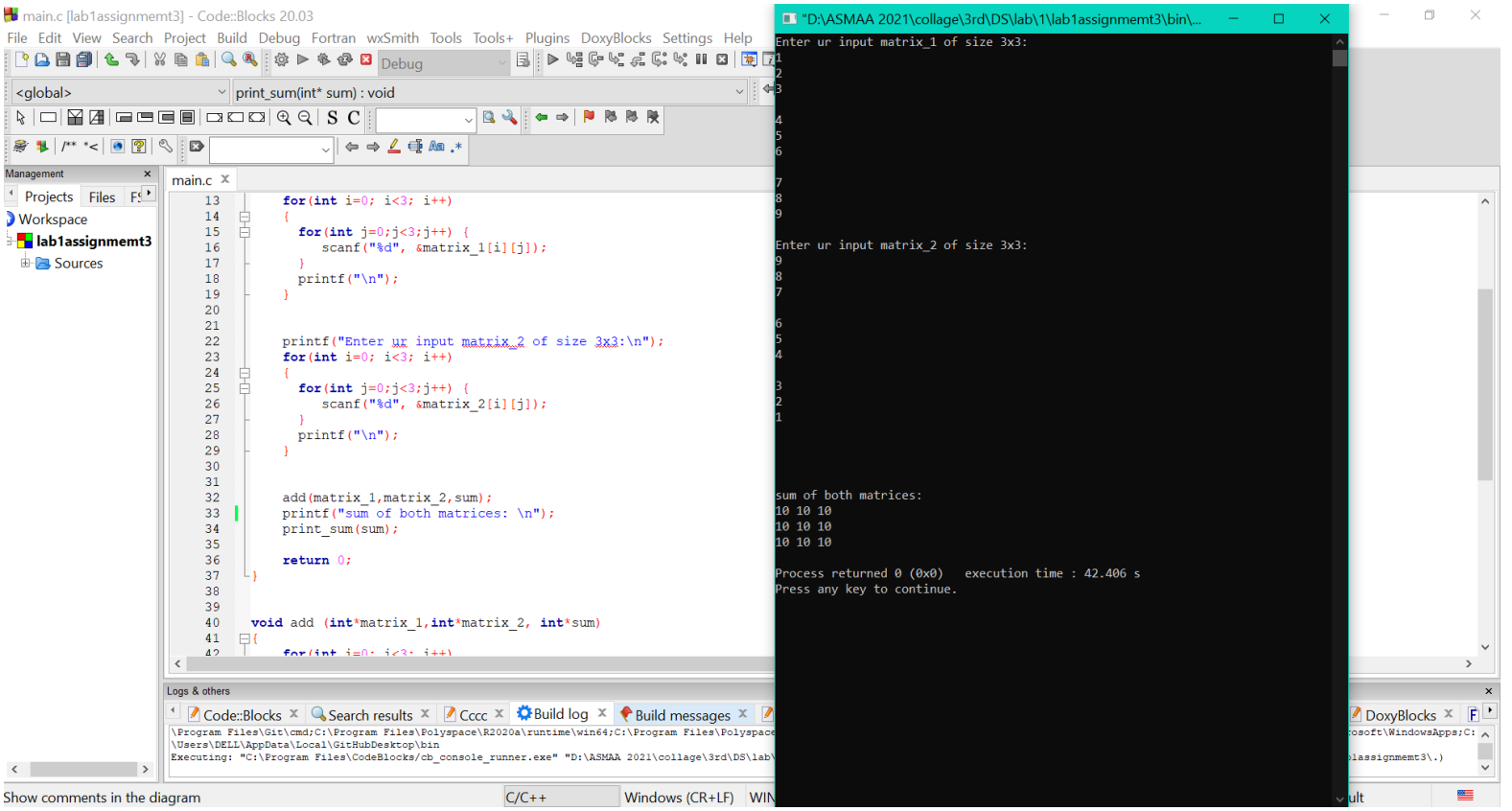
2. Problem 2 - Sales People:



"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:



"D:\ASMAA 2021\collage\3rd\DS\lab\1\lab1assignmemt3\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.