

VARUN BHARGAVA – 241010282

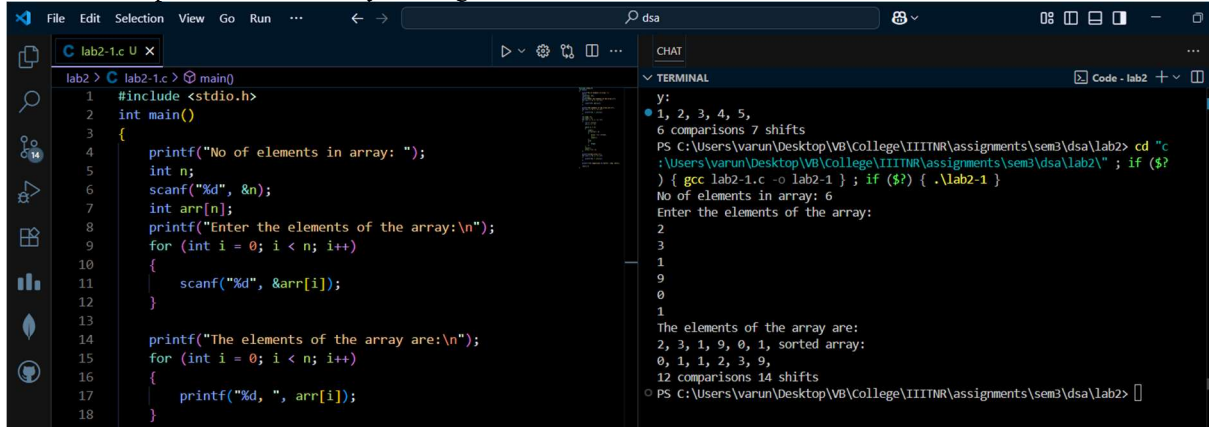
DATA STRUCTURES TASK-2

Task 1: Insertion Sort:

(<https://github.com/varunnnb/dsa-sem3-iiitnr/blob/main/lab2/lab2-1.c>)

Write a program to perform the following operations using Insertion Sort:

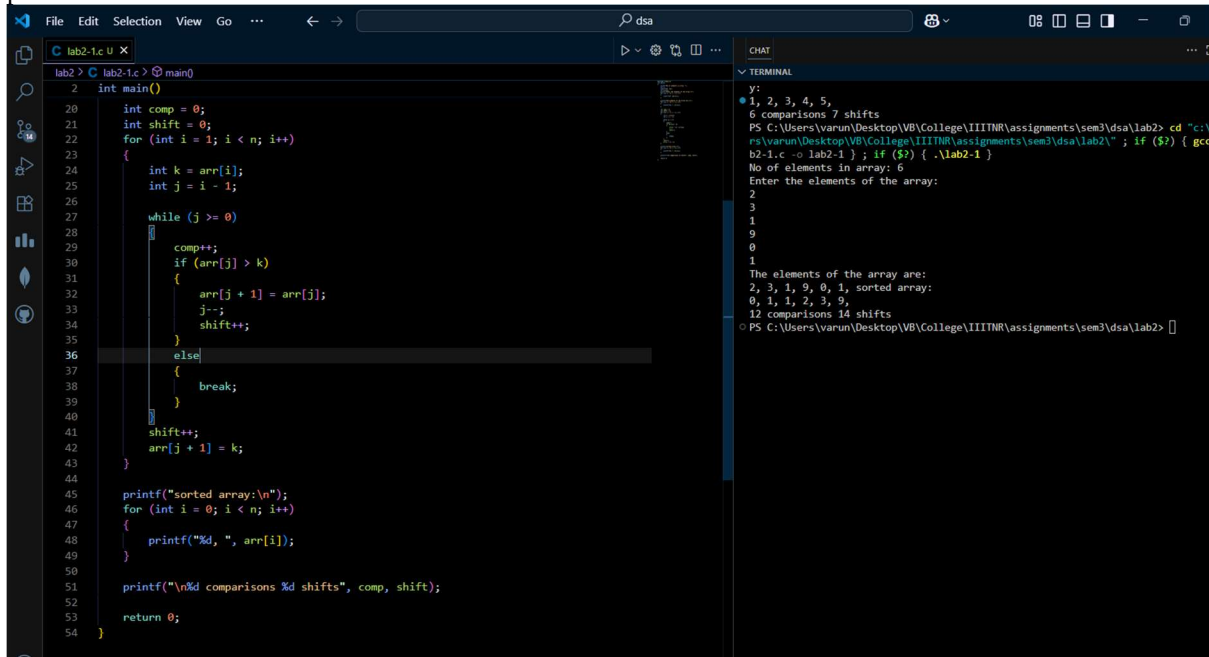
- Take user input to create an array of integers.



```
1 #include <stdio.h>
2 int main()
3 {
4     printf("No of elements in array: ");
5     int n;
6     scanf("%d", &n);
7     int arr[n];
8     printf("Enter the elements of the array:\n");
9     for (int i = 0; i < n; i++)
10    {
11        scanf("%d", &arr[i]);
12    }
13
14    printf("The elements of the array are:\n");
15    for (int i = 0; i < n; i++)
16    {
17        printf("%d ", arr[i]);
18    }
19 }
```

Y:
1, 2, 3, 4, 5,
6 comparisons 7 shifts
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2> cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2" ; if (\$?) { gcc lab2-1.c -o lab2-1 } ; if (\$?) { .\lab2-1 }
No of elements in array: 6
Enter the elements of the array:
2
3
1
9
0
1
The elements of the array are:
2, 3, 1, 9, 0, 1, sorted array:
0, 1, 1, 2, 3, 9,
12 comparisons 14 shifts
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2>

- Sort the array using Insertion Sort in ascending order.
- Display the sorted array.
- Track and display the number of comparisons and shifts performed during the sorting process.



```
20 int comp = 0;
21 int shift = 0;
22 for (int i = 1; i < n; i++)
23 {
24     int k = arr[i];
25     int j = i - 1;
26
27     while (j >= 0)
28     {
29         comp++;
30         if (arr[j] > k)
31         {
32             arr[j + 1] = arr[j];
33             j--;
34             shift++;
35         }
36         else
37         {
38             break;
39         }
40     }
41     shift++;
42     arr[j + 1] = k;
43 }
44
45 printf("sorted array:\n");
46 for (int i = 0; i < n; i++)
47 {
48     printf("%d ", arr[i]);
49 }
50
51 printf("\n%d comparisons %d shifts", comp, shift);
52
53 return 0;
54 }
```

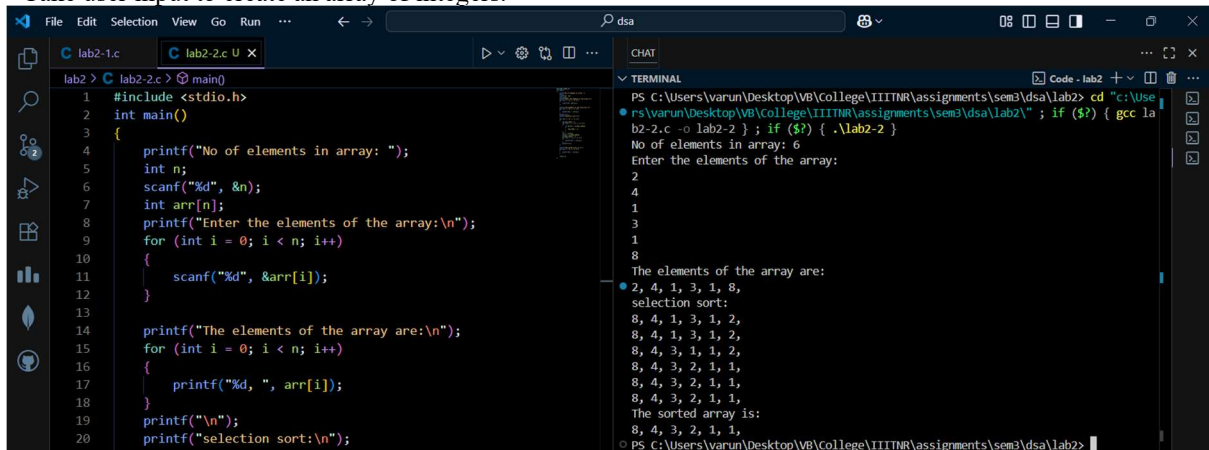
Y:
1, 2, 3, 4, 5,
6 comparisons 7 shifts
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2> cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2" ; if (\$?) { gcc lab2-1.c -o lab2-1 } ; if (\$?) { .\lab2-1 }
No of elements in array: 6
Enter the elements of the array:
2
3
1
9
0
1
The elements of the array are:
2, 3, 1, 9, 0, 1, sorted array:
0, 1, 1, 2, 3, 9,
12 comparisons 14 shifts
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2>

Task 2: Selection Sort:

(<https://github.com/varunnnb/dsa-sem3-iiitnr/blob/main/lab2/lab2-2.c>)

Write a program to perform the following operations using Selection Sort:

- Take user input to create an array of integers.

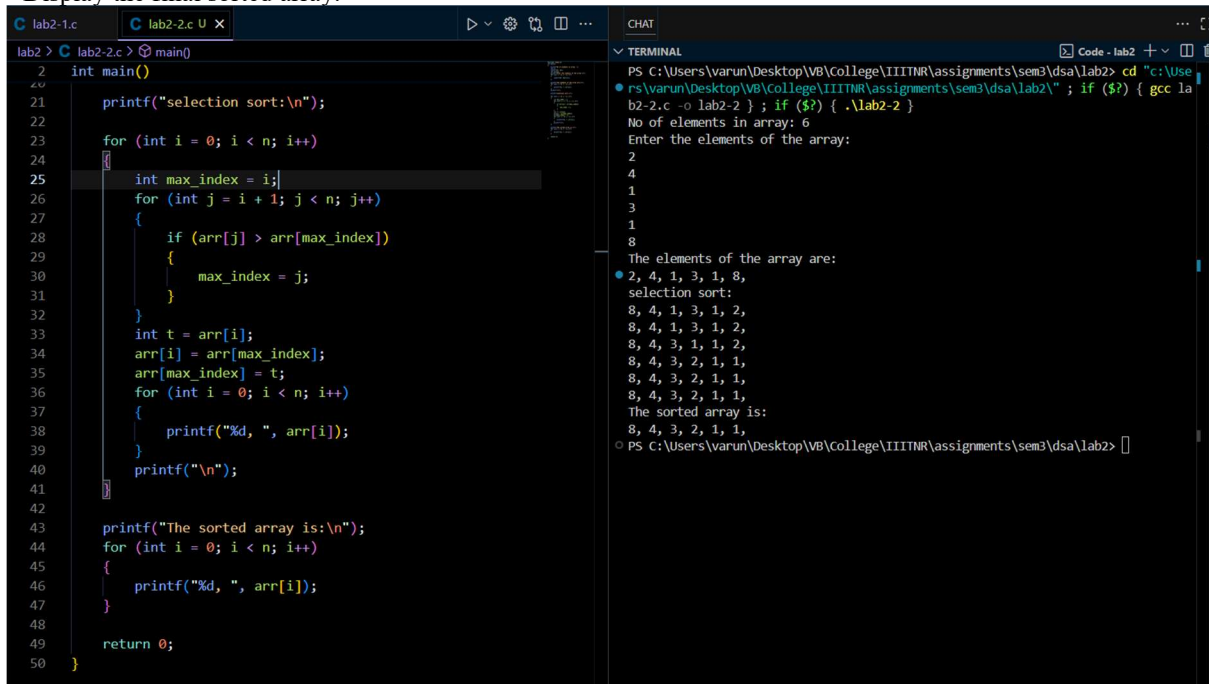


```
lab2 > C lab2-2.c > main()
1 #include <stdio.h>
2 int main()
3 {
4     printf("No of elements in array: ");
5     int n;
6     scanf("%d", &n);
7     int arr[n];
8     printf("Enter the elements of the array:\n");
9     for (int i = 0; i < n; i++)
10     {
11         scanf("%d", &arr[i]);
12     }
13
14     printf("The elements of the array are:\n");
15     for (int i = 0; i < n; i++)
16     {
17         printf("%d", arr[i]);
18     }
19     printf("\n");
20     printf("selection sort:\n");
21 }
```

Terminal output:

```
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2> cd "c:\Use
rs\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2" ; if ($?) { gcc la
b2-2.c -o lab2-2 } ; if ($?) { .\lab2-2 }
No of elements in array: 6
Enter the elements of the array:
2
4
1
3
1
8
The elements of the array are:
2, 4, 1, 3, 1, 8,
selection sort:
8, 4, 1, 3, 1, 2,
8, 4, 1, 3, 1, 2,
8, 4, 3, 1, 1, 2,
8, 4, 3, 2, 1, 1,
8, 4, 3, 2, 1, 1,
8, 4, 3, 2, 1, 1,
The sorted array is:
8, 4, 3, 2, 1, 1,
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2>
```

- Sort the array using Selection Sort in descending order.
- Print the array after each pass to show the intermediate steps of sorting.
- Display the final sorted array.



```
lab2 > C lab2-2.c > main()
21 int main()
22 {
23     printf("selection sort:\n");
24     for (int i = 0; i < n; i++)
25     {
26         int max_index = i;
27         for (int j = i + 1; j < n; j++)
28         {
29             if (arr[j] > arr[max_index])
30             {
31                 max_index = j;
32             }
33             int t = arr[i];
34             arr[i] = arr[max_index];
35             arr[max_index] = t;
36             for (int i = 0; i < n; i++)
37             {
38                 printf("%d", arr[i]);
39             }
40             printf("\n");
41         }
42
43         printf("The sorted array is:\n");
44         for (int i = 0; i < n; i++)
45         {
46             printf("%d", arr[i]);
47         }
48     }
49     return 0;
50 }
```

Terminal output:

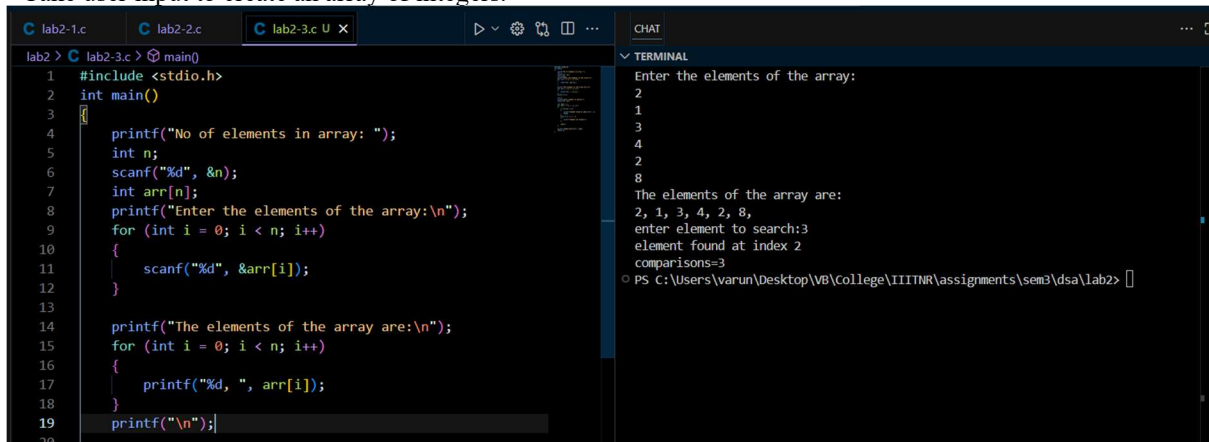
```
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2> cd "c:\Use
rs\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2" ; if ($?) { gcc la
b2-2.c -o lab2-2 } ; if ($?) { .\lab2-2 }
No of elements in array: 6
Enter the elements of the array:
2
4
1
3
1
8
The elements of the array are:
2, 4, 1, 3, 1, 8,
selection sort:
8, 4, 1, 3, 1, 2,
8, 4, 1, 3, 1, 2,
8, 4, 3, 1, 1, 2,
8, 4, 3, 2, 1, 1,
8, 4, 3, 2, 1, 1,
8, 4, 3, 2, 1, 1,
The sorted array is:
8, 4, 3, 2, 1, 1,
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2>
```

Task 3: Linear Search:

(<https://github.com/varunnnb/dsa-sem3-iiitnr/blob/main/lab2/lab2-3.c>)

Write a program to perform the following operations using Linear Search:

- Take user input to create an array of integers.



The screenshot shows a C program in a code editor. The program prompts the user for the number of elements in the array, reads the values, and prints them out. The terminal output shows the user entering 5 elements: 2, 1, 3, 4, 2, and the program printing them back.

```
lab2 > C lab2-3.c > main()
1  #include <stdio.h>
2  int main()
3
4  printf("No of elements in array: ");
5  int n;
6  scanf("%d", &n);
7  int arr[n];
8  printf("Enter the elements of the array:\n");
9  for (int i = 0; i < n; i++)
10 {
11     scanf("%d", &arr[i]);
12 }
13
14 printf("The elements of the array are:\n");
15 for (int i = 0; i < n; i++)
16 {
17     printf("%d, ", arr[i]);
18 }
19 printf("\n");
20
```

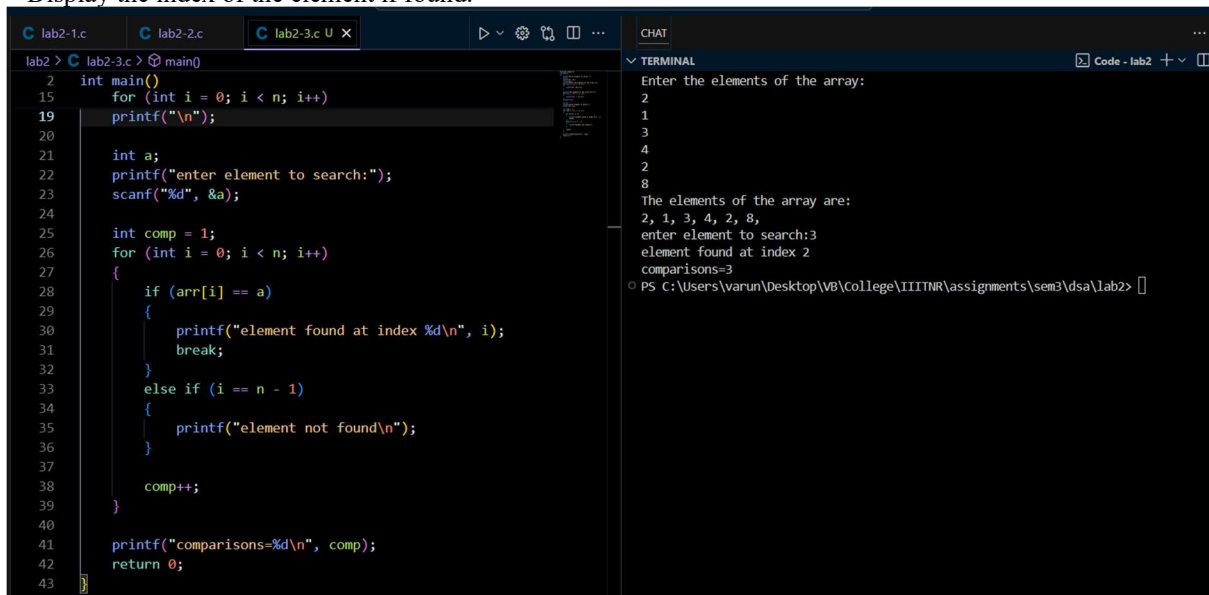
TERMINAL

```
Enter the elements of the array:
2
1
3
4
2
8
The elements of the array are:
2, 1, 3, 4, 2, 8,
enter element to search:3
element found at index 2
comparisons=3
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2>
```

- Take input for the element to be searched in the array.

- Perform Linear Search:

– Display the index of the element if found.



The screenshot shows a C program in a code editor. The program prompts the user for an element to search, then iterates through the array to find it. If found, it prints the index; otherwise, it prints "element not found". The terminal output shows the user searching for 3, which is found at index 2.

```
lab2 > C lab2-3.c > main()
15 for (int i = 0; i < n; i++)
19 printf("\n");
20
21 int a;
22 printf("enter element to search:");
23 scanf("%d", &a);
24
25 int comp = 1;
26 for (int i = 0; i < n; i++)
27 {
28     if (arr[i] == a)
29     {
30         printf("element found at index %d\n", i);
31         break;
32     }
33     else if (i == n - 1)
34     {
35         printf("element not found\n");
36     }
37     comp++;
38 }
39
40 printf("comparisons=%d\n", comp);
41 return 0;
42
43
```

TERMINAL

```
Enter the elements of the array:
2
1
3
4
2
8
The elements of the array are:
2, 1, 3, 4, 2, 8,
enter element to search:3
element found at index 2
comparisons=3
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2>
```

- If not found, display Element not found in the array.
- Display the number of comparisons made during the search.

The image shows a code editor with a C program for linear search and its terminal output. The code is as follows:

```
2 int main()
19 {
20     printf("\n");
21
22     int a;
23     printf("enter element to search:");
24     scanf("%d", &a);
25
26     int comp = 1;
27     for (int i = 0; i < n; i++)
28     {
29         if (arr[i] == a)
30         {
31             printf("element found at index %d\n", i);
32             break;
33         }
34         else if (i == n - 1)
35         {
36             printf("element not found\n");
37         }
38         comp++;
39     }
40
41     printf("comparisons=%d\n", comp);
42     return 0;
43 }
```

The terminal output shows the execution of the program:

```
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2> cd "c
:Users\varun\Desktop\VB\College\IIITNR\assig
:Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2" ; if ($?) { gcc lab2-3.c -o lab2-3 } ; if ($?) { .\lab2-3 }
No of elements in array: 6
Enter the elements of the array:
3
1
3
5
2
6
The elements of the array are:
3, 1, 3, 5, 2, 6,
enter element to search:4
element not found
comparisons=7
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2>
```

Task 4: Binary Search:

(<https://github.com/varunnnb/dsa-sem3-iiitnr/blob/main/lab2/lab2-4.c>)

Write a program to perform the following operations using Binary Search:

- Take user input to create a sorted array of integers.

```
lab2 > C lab2-4.c > main()
1  #include <stdio.h>
2  int main()
3  {
4      printf("No of elements in array: ");
5      int n;
6      scanf("%d", &n);
7      int arr[n];
8      printf("Enter the elements of the array:\n");
9      for (int i = 0; i < n; i++)
10     {
11         scanf("%d", &arr[i]);
12     }
13
14     printf("The elements of the array are:\n");
15     for (int i = 0; i < n; i++)
16     {
17         printf("%d, ", arr[i]);
18     }
19     printf("\n");
20
21     int sorted = 1;
22     for (int i = 0; i < n; i++)
23     {
24         if (arr[i] > arr[i + 1])
25         {
26             sorted = 0;
27             break;
28         }
29     }
```

```
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2> cd "c:\Use
rs\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2" ; if ($?) { gcc la
b2-4.c -o lab2-4 } ; if ($?) { .\lab2-4 }
No of elements in array: 6
Enter the elements of the array:
1
3
2
6
4
9
The elements of the array are:
1, 3, 2, 6, 4, 9,
The sorted array is:
1, 2, 3, 4, 6, 9, enter element to search:2
element found.
steps taken: 3
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2>
```

```
lab2 > C lab2-4.c > main()
2  int main()
31
32     if (sorted == 0)
33     {
34         for (int i = 0; i < n; i++)
35         {
36             int min_index = i;
37             for (int j = i + 1; j < n; j++)
38             {
39                 if (arr[j] < arr[min_index])
40                 {
41                     min_index = j;
42                 }
43             }
44             int t = arr[i];
45             arr[i] = arr[min_index];
46             arr[min_index] = t;
47         }
48
49         printf("The sorted array is:\n");
50         for (int i = 0; i < n; i++)
51         {
52             printf("%d, ", arr[i]);
53         }
54     }
55
56     else
57     {
58         printf("array is already sorted.\n");
59     }
```

```
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2> cd "c:\Use
rs\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2" ; if ($?) { gcc la
b2-4.c -o lab2-4 } ; if ($?) { .\lab2-4 }
No of elements in array: 6
Enter the elements of the array:
1
3
2
6
4
9
The elements of the array are:
1, 3, 2, 6, 4, 9,
The sorted array is:
1, 2, 3, 4, 6, 9, enter element to search:2
element found.
steps taken: 3
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2>
```

- Take input for the element to be searched in the array.
- Perform Binary Search:
- Display the index of the element if found.

```

lab2-4.c main()
2 int main()
61 printf("enter element to search:");
62 int a;
63 scanf("%d", &a);
64
65 int low = 0;
66 int high = n - 1;
67 int found = 0;
68 int steps = 0;
69
70 while (low <= high)
71 {
72     int mid = (low + high) / 2;
73     steps++;
74     if (arr[mid] == a)
75     {
76         found = 1;
77         break;
78     }
79     else if (arr[mid] < a)
80     {
81         low = mid + 1;
82     }
83     else
84     {
85         high = mid - 1;
86     }
87 }
88
89 if (found)
90 {
91     printf("element found.\n", low);
92 }
93 else

```

TERMINAL

```

PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2> cd "c:\Use
rs\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2" ; if ($?) { gcc la
b2-4.c -o lab2-4 } ; if ($?) { .\lab2-4 }
No of elements in array: 6
Enter the elements of the array:
1
3
2
6
4
9
The elements of the array are:
1, 3, 2, 6, 4, 9,
The sorted array is:
1, 2, 3, 4, 6, 9, enter element to search:2
element found.
steps taken: 3
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2>

```

- If not found, display an appropriate message.
- Display the number of steps/iterations taken to find the element.

```

lab2-4.c main()
2 int main()
70 while (low <= high)
71 {
72     int mid = (low + high) / 2;
73     steps++;
74     if (arr[mid] == a)
75     {
76         found = 1;
77         break;
78     }
79     else if (arr[mid] < a)
80     {
81         low = mid + 1;
82     }
83     else
84     {
85         high = mid - 1;
86     }
87 }
88
89 if (found)
90 {
91     printf("element found.\n", low);
92 }
93 else
94 {
95     printf("element not found.\n", a);
96 }
97
98 printf("steps taken: %d\n", steps);
99 return 0;
100 }

```

TERMINAL

```

PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2> cd "c:\Use
rs\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2" ; if ($?) { gcc la
b2-4.c -o lab2-4 } ; if ($?) { .\lab2-4 }
No of elements in array: 6
Enter the elements of the array:
1
3
2
6
4
9
The elements of the array are:
1, 3, 2, 6, 4, 9,
The sorted array is:
1, 2, 3, 4, 6, 9, enter element to search:2
element found.
steps taken: 3
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2>

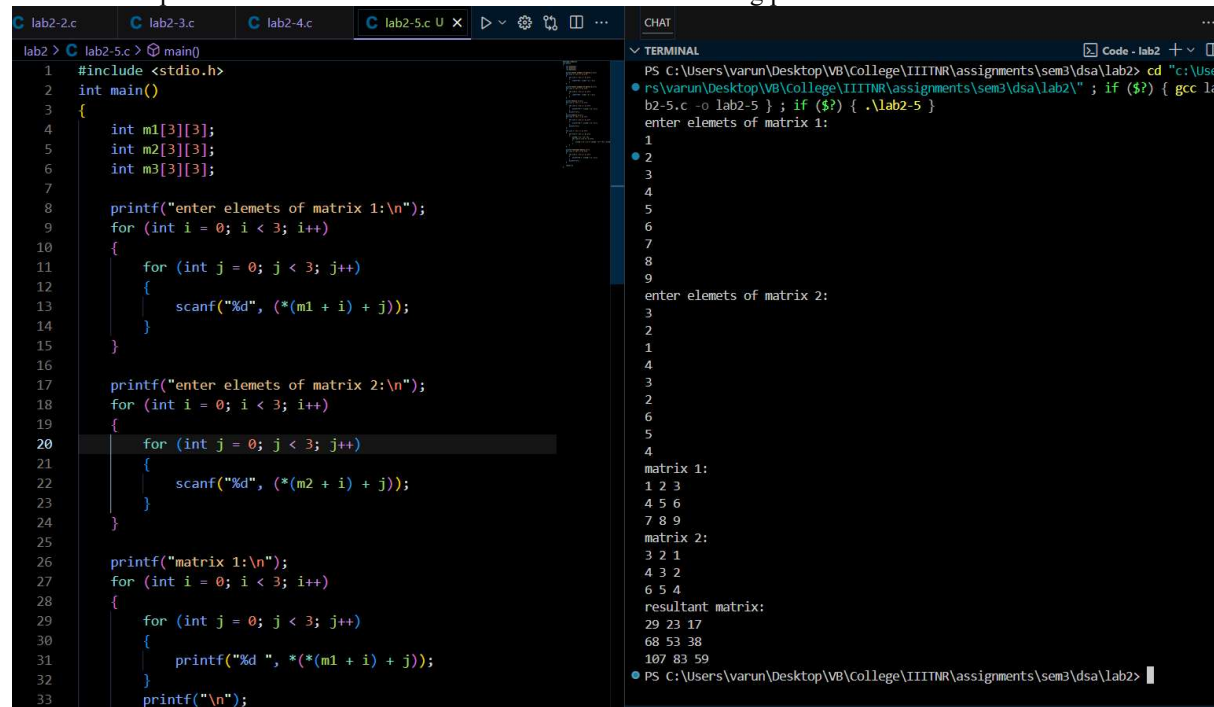
```


Task 5: Matrix Multiplication Using Pointers:

(<https://github.com/varunnrb/dsa-sem3-iiitnr/blob/main/lab2/lab2-5.c>)

Write a program to perform the following operations related to matrix multiplication:

- Declare three 2D matrices A[3][3], B[3][3], and C[3][3].
- Take user input to enter all elements of Matrix A and Matrix B using pointers.

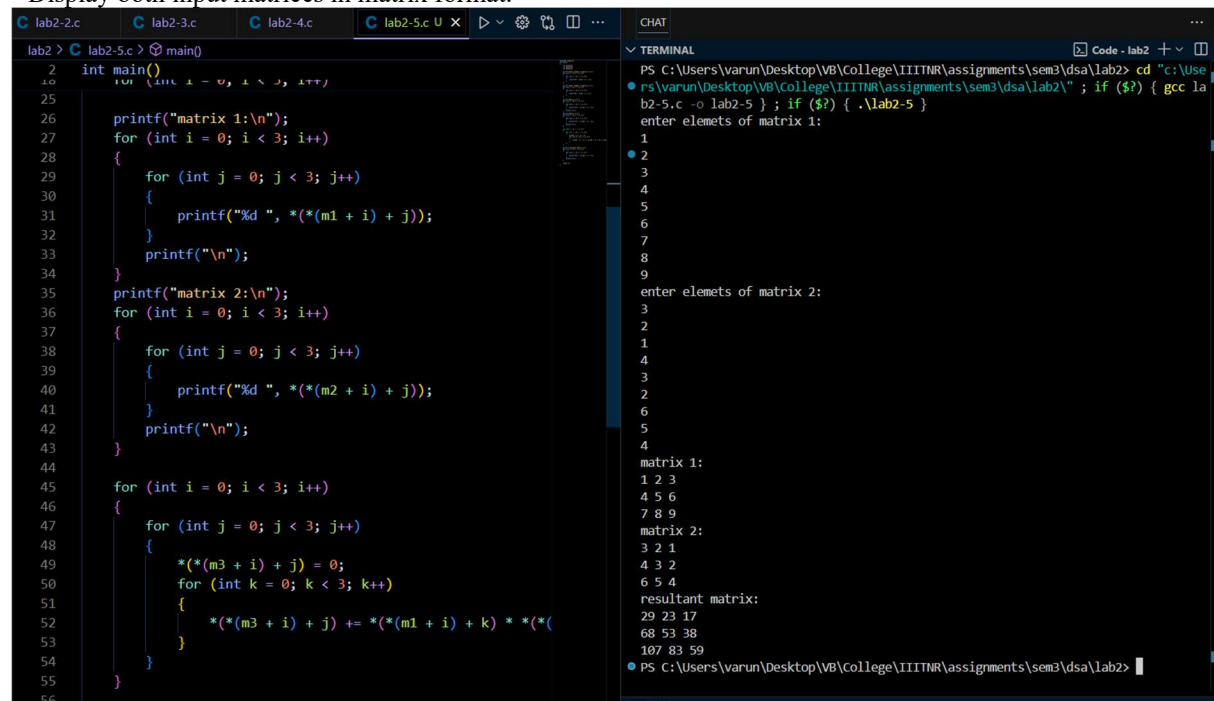


```
1 #include <stdio.h>
2 int main()
3 {
4     int m1[3][3];
5     int m2[3][3];
6     int m3[3][3];
7
8     printf("enter elemets of matrix 1:\n");
9     for (int i = 0; i < 3; i++)
10     {
11         for (int j = 0; j < 3; j++)
12         {
13             scanf("%d", (*(m1 + i) + j));
14         }
15     }
16
17     printf("enter elemets of matrix 2:\n");
18     for (int i = 0; i < 3; i++)
19     {
20         for (int j = 0; j < 3; j++)
21         {
22             scanf("%d", (*(m2 + i) + j));
23         }
24     }
25
26     printf("matrix 1:\n");
27     for (int i = 0; i < 3; i++)
28     {
29         for (int j = 0; j < 3; j++)
30         {
31             printf("%d ", (*(m1 + i) + j));
32         }
33         printf("\n");
34     }
```

TERMINAL

```
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2> cd "C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2" & if ($?) { gcc lab2-5.c -o lab2-5 } ; if ($?) { .\lab2-5 }
enter elemets of matrix 1:
1
2
3
4
5
6
7
8
9
enter elemets of matrix 2:
3
2
1
4
3
2
6
5
4
matrix 1:
1 2 3
4 5 6
7 8 9
matrix 2:
3 2 1
4 3 2
6 5 4
resultant matrix:
29 23 17
68 53 38
107 83 59
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2>
```

- Display both input matrices in matrix format.



```
2 int main()
3 {
4     for (int i = 0; i < 3; i++)
5     {
6         for (int j = 0; j < 3; j++)
7         {
8             scanf("%d", (*(m1 + i) + j));
9         }
10     }
11
12     printf("matrix 1:\n");
13     for (int i = 0; i < 3; i++)
14     {
15         for (int j = 0; j < 3; j++)
16         {
17             printf("%d ", (*(m1 + i) + j));
18         }
19         printf("\n");
20     }
21
22     printf("matrix 2:\n");
23     for (int i = 0; i < 3; i++)
24     {
25         for (int j = 0; j < 3; j++)
26         {
27             scanf("%d", (*(m2 + i) + j));
28         }
29     }
30
31     printf("matrix 2:\n");
32     for (int i = 0; i < 3; i++)
33     {
34         for (int j = 0; j < 3; j++)
35         {
36             printf("%d ", (*(m2 + i) + j));
37         }
38         printf("\n");
39     }
40
41     for (int i = 0; i < 3; i++)
42     {
43         for (int j = 0; j < 3; j++)
44         {
45             *(m3 + i) + j = 0;
46             for (int k = 0; k < 3; k++)
47             {
48                 *(m3 + i) + j += (*(m1 + i) + k) * (*(m2 + i) + k);
49             }
50         }
51     }
52
53     printf("resultant matrix:\n");
54     for (int i = 0; i < 3; i++)
55     {
56         for (int j = 0; j < 3; j++)
57         {
58             printf("%d ", *(m3 + i) + j);
59         }
60         printf("\n");
61     }
```

TERMINAL

```
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2> cd "C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2" & if ($?) { gcc lab2-5.c -o lab2-5 } ; if ($?) { .\lab2-5 }
enter elemets of matrix 1:
1
2
3
4
5
6
7
8
9
enter elemets of matrix 2:
3
2
1
4
3
2
6
5
4
matrix 1:
1 2 3
4 5 6
7 8 9
matrix 2:
3 2 1
4 3 2
6 5 4
resultant matrix:
29 23 17
68 53 38
107 83 59
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2>
```

- Perform matrix multiplication using pointer arithmetic and store the result in Matrix C.
- Display the resultant matrix C in matrix format.

The screenshot shows a C++ IDE with a file named `lab2-5.c` open. The code implements a 3x3 matrix multiplication using pointer arithmetic. The `main` function initializes a 3x3 matrix `m3` to zero, then reads elements for matrix 1 and matrix 2. It then performs the multiplication using nested loops and pointer arithmetic to calculate the resultant matrix `m3`, which is finally printed in a 3x3 matrix format.

```

2  int main()
36  {
40      for (int i = 0; i < 3; i++)
42          printf("\n");
43      }
44
45      for (int i = 0; i < 3; i++)
46      {
47          for (int j = 0; j < 3; j++)
48          {
49              (*(m3 + i) + j) = 0;
50              for (int k = 0; k < 3; k++)
51              {
52                  (*(m3 + i) + j) += (*(m1 + i) + k) * (*(m2 + i) + k);
53              }
54          }
55      }
56
57      printf("resultant matrix:\n");
58      for (int i = 0; i < 3; i++)
59      {
60          for (int j = 0; j < 3; j++)
61          {
62              printf("%d ", (*(m3 + i) + j));
63          }
64          printf("\n");
65      }
66
67      return 0;
68  }

```

The terminal output shows the execution of the program. It prompts for the elements of matrix 1 and matrix 2, and then displays the resultant matrix:

```

PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2> cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2" ; if ($?) { gcc lab2-5.c -o lab2-5 } ; if ($?) { .\lab2-5 }
enter elemets of matrix 1:
1
2
3
4
5
6
7
8
9
enter elemets of matrix 2:
3
2
1
4
5
6
7
8
9
matrix 1:
1 2 3
4 5 6
7 8 9
matrix 2:
3 2 1
4 3 2
6 5 4
resultant matrix:
29 23 17
68 53 38
107 83 59
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab2>

```