

## EX:1

```
1 #include <stdio.h>
2
3+ int main() {
4     int n, rev = 0, rem;
5     printf("Enter a 32-bit integer: ");
6     scanf("%d", &n);
7
8+     while(n != 0) {
9         rem = n % 10;
10        rev = rev * 10 + rem;
11        n /= 10;
12    }
13
14    printf("Reversed number = %d\n", rev);
15    return 0;
16 }
```

```
Enter a 32-bit integer: 1234
Reversed number = 4321
```

```
== Code Execution Successful ==
```

## EX: 2

```
#include <stdio.h>
#include <ctype.h>

int main() {
    char str[100];
    int i, valid = 1;

    printf("Enter a string: ");
    gets(str);

    for(i = 0; str[i] != '\0'; i++) {
        if(!isalpha(str[i])) {
            valid = 0;
            break;
        }
    }
}
```

```
Enter a 32-bit integer: 1234
Reversed number = 4321
```

```
== Code Execution Successful ==
```

```
9     gets(str);
10
11+    for(i = 0; str[i] != '\0'; i++) {
12+        if(!isalpha(str[i])) {
13+            valid = 0;
14+            break;
15        }
16    }
17
18    if(valid)
19        printf("Valid String (only alphabets)\n");
20    else
21        printf("Invalid String\n");
22    return 0;
23 }
24 |
```

```
Enter a 32-bit integer: 1234
Reversed number = 4321
```

```
== Code Execution Successful ==
```

## EX:6 A

```
1 #include <stdio.h>
2
3* int main() {
4     int n, key, i, found = 0;
5
6     printf("Enter number of elements: ");
7     scanf("%d", &n);
8
9     int arr[n];
10    printf("Enter %d elements:\n", n);
11    for(i = 0; i < n; i++)
12        scanf("%d", &arr[i]);
13
14    printf("Enter element to search: ");
15    scanf("%d", &key);
16
```

```
Enter number of elements: 3
Enter 3 elements:
10 20 40
Enter element to search: 20
Element 20 found at position 2 (index 1)

== Code Execution Successful ==
```

```
16
17*     for(i = 0; i < n; i++) {
18*         if(arr[i] == key) {
19*             printf("Element %d found at position %d (index %d)\n", key, i
+ 1, i);
20*             found = 1;
21*             break;
22*         }
23     }
24
25     if(!found)
26         printf("Element %d not found in array\n", key);
27
28     return 0;
29 }
30 |
```

```
Enter number of elements: 3
Enter 3 elements:
10 20 40
Enter element to search: 20
Element 20 found at position 2 (index 1)

== Code Execution Successful ==
```

## EX 6 B

```
main.c
1 #include <stdio.h>
2
3* int main() {
4     int n, key, low, high, mid, found = 0;
5
6     printf("Enter number of elements: ");
7     scanf("%d", &n);
8
9     int arr[n];
10    printf("Enter %d elements in sorted order:\n", n);
11    for(int i = 0; i < n; i++)
12        scanf("%d", &arr[i]);
13
14    printf("Enter element to search: ");
15    scanf("%d", &key);
16
```

```
Enter number of elements: 3
Enter 3 elements in sorted order:
10 20 30
Enter element to search: 30
Element 30 found at position 3 (index 2)

== Code Execution Successful ==
```

```
low = 0;
high = n - 1;

while(low <= high) {
    mid = (low + high) / 2;

    if(arr[mid] == key) {
        printf("Element %d found at position %d (index %d)\n", key,
+ mid + 1, mid);
        found = 1;
        break;
    }
    else if(arr[mid] < key)
        low = mid + 1;
    else
        high = mid - 1;
```

```
Enter number of elements: 3
Enter 3 elements in sorted order:
10 20 30
Enter element to search: 30
Element 30 found at position 3 (index 2)

== Code Execution Successful ==
```

```
26         break;
27     }
28     else if(arr[mid] < key)
29         low = mid + 1;
30     else
31         high = mid - 1;
32 }
33
34 if(!found)
35     printf("Element %d not found in array\n", key);
36
37 return 0;
38 }
```

Enter number of elements: 3  
Enter 3 elements in sorted order:  
10 20 30  
Enter element to search: 30  
Element 30 found at position 3 (index 2)  
== Code Execution Successful ==

## EX:7

```
main.c | Run | Output
```

```
1 #include <stdio.h>
2
3 int main() {
4     int arr[5] = {10, 20, 30, 40, 50}, key, i, found = 0;
5     printf("Enter element to find: ");
6     scanf("%d", &key);
7
8     for(i = 0; i < 5; i++) {
9         if(arr[i] == key) {
10             printf("Element found at index %d\n", i);
11             found = 1;
12             break;
13         }
14     }
15     if(!found)
16         printf("Element not found\n");
17     return 0;
18 }
```

Enter element to find: 20  
Element found at index 1  
== Code Execution Successful ==

## EX:8

```
main.c | Run | Output
```

```
1 #include <stdio.h>
2
3 int main() {
4     int arr[5] = {2, 5, 8, 11, 14};
5     printf("Even numbers: ");
6     for(int i=0; i<5; i++)
7         if(arr[i] % 2 == 0)
8             printf("%d ", arr[i]);
9
10    printf("\nOdd numbers: ");
11    for(int i=0; i<5; i++)
12        if(arr[i] % 2 != 0)
13            printf("%d ", arr[i]);
14
15    return 0;
16 }
```

Even numbers: 2 8 14  
Odd numbers: 5 11  
== Code Execution Successful ==

## EX:9

```
1 #include <stdio.h>
2
3 int main() {
4     int n, a=0, b=1, c, sum=1;
5     printf("Enter terms: ");
6     scanf("%d", &n);
7
8     for(int i=2; i<n; i++) {
9         c = a + b;
10        sum += c;
11        a = b;
12        b = c;
13    }
14    printf("Sum of Fibonacci series = %d\n", sum);
15    return 0;
16 }
```

```
Enter terms: 5
Sum of Fibonacci series = 7

==== Code Execution Successful ===
```

## EX:10

```
main.c
1 #in[main.c] <stdio.h>
2
3 int main() {
4     int n, fact = 1;
5     printf("Enter number: ");
6     scanf("%d", &n);
7
8     for(int i=1; i<=n; i++)
9         fact *= i;
10
11    printf("Factorial = %d\n", fact);
12    return 0;
13 }
14
15
```

```
Enter number: 5
Factorial = 120

==== Code Execution Successful ===
```

## EX:17

```
1 #include <stdio.h>
2
3 int main() {
4     int arr[5] = {10, 20, 30, 40, 50};
5     int key, choice, i, found = 0;
6     int low, high, mid;
7
8     printf("Array elements: 10 20 30 40 50\n");
9     printf("\nChoose search method:\n");
10    printf("1. Linear Search\n");
11    printf("2. Binary Search\n");
12    printf("Enter your choice: ");
13    scanf("%d", &choice);
14
15    printf("Enter number to search: ");
16    scanf("%d", &key);
```

```
Array elements: 10 20 30 40 50
Choose search method:
1. Linear Search
2. Binary Search
Enter your choice: 1
Enter number to search: 10
Element found at position 1 (index 0)

==== Code Execution Successful ===
```

```

17
18    switch(choice) {
19        case 1: // Linear Search
20            for(i = 0; i < 5; i++) {
21                if(arr[i] == key) {
22                    printf("Element found at position %d (index %d)\n", i
23                                + 1, i);
24                    found = 1;
25                    break;
26                }
27            if(!found)
28                printf("Element not found\n");
29            break;
30
31        case 2: // Binary Search

```

main.c

```

32     low = 0;
33     high = 4;
34     found = 0;
35
36     while(low <= high) {
37         mid = (low + high) / 2;
38
39         if(arr[mid] == key) {
40             printf("Element found at position %d (index %d)\n",
41                     mid + 1, mid);
42             found = 1;
43             break;
44         } else if(arr[mid] < key)
45             low = mid + 1;
46         else
47             high = mid - 1;
48
49         if(!found)
50             printf("Element not found\n");
51         break;
52     default:
53         printf("Invalid choice! Please enter 1 or 2.\n");
54     }
55
56     return 0;
57 }
58

```

Output

```

Array elements: 10 20 30 40 50
Choose search method:
1. Linear Search
2. Binary Search
Enter your choice: 1
Enter number to search: 10
Element found at position 1 (index 0)

== Code Execution Successful ==

```

```

Array elements: 10 20 30 40 50
Choose search method:
1. Linear Search
2. Binary Search
Enter your choice: 1
Enter number to search: 10
Element found at position 1 (index 0)

== Code Execution Successful ==

```

```

Array elements: 10 20 30 40 50
Choose search method:
1. Linear Search
2. Binary Search
Enter your choice: 1
Enter number to search: 10
Element found at position 1 (index 0)

== Code Execution Successful ==

```

## EX:31

```

1 #include <stdio.h>
2 #include <string.h>
3
4 int main() {
5     char str[100];
6     int i, j, len;
7
8     printf("Enter a string: ");
9     fgets(str, sizeof(str), stdin); // Safe input function
10
11    // Remove newline character added by fgets
12    str[strcspn(str, "\n")] = '\0';
13
14    len = strlen(str);
15    printf("\nRepeated characters and their indexes:\n");
16

```

Enter a string: HELLO

Repeated characters and their indexes:  
Character 'L' repeated at index 2 and 3

== Code Execution Successful ==

```

16
17+     for(i = 0; i < len; i++) {
18+         for(j = i + 1; j < len; j++) {
19+             if(str[i] == str[j] && str[i] != ' ') {
20+                 printf("Character '%c' repeated at index %d and %d\n",
21+                         str[i], i, j);
22+             }
23     }
24
25     return 0;
26 }
27
28
29

```

Enter a string: HELLO  
 Repeated characters and their indexes:  
 Character 'L' repeated at index 2 and 3  
 === Code Execution Successful ===

## EX:32

```

1 #include <stdio.h>
2
3 int main() {
4     int arr[] = {1, 2, 2, 3, 3, 3, 4};
5     int n = 7, count, maxCount = 0, freqNum;
6
7     for(int i=0; i<n; i++) {
8         count = 1;
9         for(int j=i+1; j<n; j++) {
10            if(arr[i] == arr[j])
11                count++;
12        }
13        if(count > maxCount) {
14            maxCount = count;
15            freqNum = arr[i];
16        }
17    }
18
19    printf("Most frequent number: %d (count = %d)\n", freqNum, maxCount);
20    return 0;
21 }

```

Most frequent number: 3 (count = 3)  
 === Code Execution Successful ===

Most frequent number: 3 (count = 3)  
 === Code Execution Successful ===

## EX:42

main.c

Run Output

```

1 #include <stdio.h>
2
3 int main() {
4     int a[2][2] = {{1, 2}, {3, 4}};
5     int rowSum, colSum;
6
7     for(int i=0; i<2; i++) {
8         rowSum = 0;
9         for(int j=0; j<2; j++)
10            rowSum += a[i][j];
11         printf("Sum of row %d = %d\n", i, rowSum);
12     }
13
14     for(int j=0; j<2; j++) {
15         colSum = 0;
16         for(int i=0; i<2; i++)

```

Sum of row 0 = 3  
 Sum of row 1 = 7  
 Sum of column 0 = 4  
 Sum of column 1 = 6  
 === Code Execution Successful ===

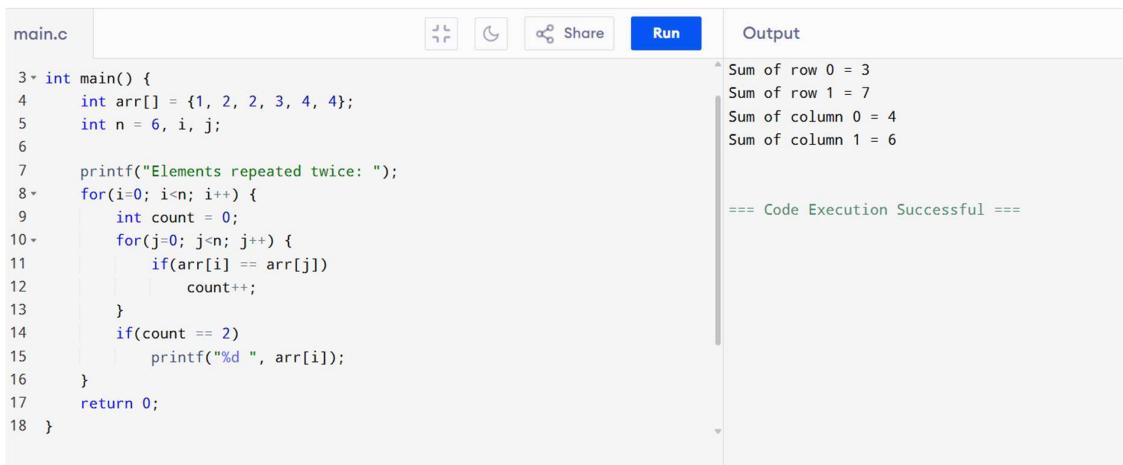
```

11     printf("Sum of row %d = %d\n", i, rowSum);
12 }
13
14 for(int j=0; j<2; j++) {
15     colSum = 0;
16     for(int i=0; i<2; i++)
17         colSum += arr[i][j];
18     printf("Sum of column %d = %d\n", j, colSum);
19 }
20
21 return 0;
22 }
23

```

Sum of row 0 = 5  
Sum of row 1 = 7  
Sum of column 0 = 4  
Sum of column 1 = 6  
  
==== Code Execution Successful ===

## EX:43



The screenshot shows a code editor interface with a tab labeled "main.c". The code is as follows:

```

3 int main() {
4     int arr[] = {1, 2, 2, 3, 4, 4};
5     int n = 6, i, j;
6
7     printf("Elements repeated twice: ");
8     for(i=0; i<n; i++) {
9         int count = 0;
10        for(j=0; j<n; j++) {
11            if(arr[i] == arr[j])
12                count++;
13        }
14        if(count == 2)
15            printf("%d ", arr[i]);
16    }
17    return 0;
18 }

```

The output window on the right shows the execution results:

Sum of row 0 = 3  
Sum of row 1 = 7  
Sum of column 0 = 4  
Sum of column 1 = 6  
  
==== Code Execution Successful ===

## EX:48

```

2
3 int main() {
4     int arr[3][3] = {
5         {1, 9, 3},
6         {4, 2, 8},
7         {7, 6, 5}
8     };
9     int max = arr[0][0];
10
11    for(int i=0; i<3; i++)
12        for(int j=0; j<3; j++)
13            if(arr[i][j] > max)
14                max = arr[i][j];
15
16    printf("Largest element in matrix = %d\n", max);
17    return 0;

```

Largest element in matrix = 9  
  
==== Code Execution Successful ===

## EX:

```
1 #include <stdio.h>
2
3 int main() {
4     int a[2][2], b[2][2], c[2][2];
5     int i, j, k, choice;
6
7     printf("Enter elements of Matrix A (2x2):\n");
8     for(i=0;i<2;i++)
9         for(j=0;j<2;j++)
10            scanf("%d",&a[i][j]);
11
12    printf("Enter elements of Matrix B (2x2):\n");
13    for(i=0;i<2;i++)
14        for(j=0;j<2;j++)
15            scanf("%d",&b[i][j]);
16
17    printf("\n1.Addition\n2.Subtraction\n3.Multiplication\n4
18 .Traversal\nEnter choice: ");
19    scanf("%d",&choice);
20
21    if(choice==1){
22        printf("\nAddition:\n");
23        for(i=0;i<2;i++){
24            for(j=0;j<2;j++){
25                c[i][j]=a[i][j]+b[i][j];
26                printf("%d ",c[i][j]);
27            }
28            printf("\n");
29        }
30    } else if(choice==2){
31        printf("\nSubtraction:");
32
33        for(i=0;i<2;i++){
34            for(j=0;j<2;j++){
35                c[i][j]=a[i][j]-b[i][j];
36                printf("%d ",c[i][j]);
37            }
38            printf("\n");
39        }
40    } else if(choice==3){
41        printf("\nMultiplication:");
42        for(i=0;i<2;i++){
43            for(j=0;j<2;j++){
44                c[i][j]=0;
45                for(k=0;k<2;k++)
46                    c[i][j]+=a[i][k]*b[k][j];
47                printf("%d ",c[i][j]);
48            }
49            printf("\n");
50        }
51    } else if(choice==4){
52        printf("\nMatrix A:");
53        for(i=0;i<2;i++){
54            for(j=0;j<2;j++)
55                printf("%d ",a[i][j]);
56            printf("\n");
57        }
58
59    printf("\nMatrix B:");
60    for(i=0;i<2;i++){
61        for(j=0;j<2;j++)
62            printf("%d ",b[i][j]);
63    }
64
65    printf("5 4
66 Enter elements of Matrix B (2x2):
67 2 3
68 4 5
69
70 1.Addition
71 2.Subtraction
72 3.Multiplication
73 4.Traversal
74 Enter choice: 2
75
76 Subtraction:
77 -1 -1
78 1 -1
79
80 5 4
81 Enter elements of Matrix B (2x2):
82 2 3
83 4 5
84
85 1.Addition
86 2.Subtraction
87 3.Multiplication
88 4.Traversal
89 Enter choice: 2
90
91 Subtraction:
92 -1 -1
93 1 -1
94
95 5 4
96 Enter elements of Matrix B (2x2):
97 2 3
98 4 5
99
100 1.Addition
101 2.Subtraction
102 3.Multiplication
103 4.Traversal
104 Enter choice: 2
105
106 Subtraction:
107 -1 -1
108 1 -1
109
110 5 4
111 Enter elements of Matrix B (2x2):
112 2 3
113 4 5
114
115 1.Addition
116 2.Subtraction
117 3.Multiplication
118 4.Traversal
119 Enter choice: 2
120
121 Subtraction:
122 -1 -1
123 1 -1
124
125 5 4
126 Enter elements of Matrix B (2x2):
127 2 3
128 4 5
129
130 1.Addition
131 2.Subtraction
132 3.Multiplication
133 4.Traversal
134 Enter choice: 2
135
136 Subtraction:
137 -1 -1
138 1 -1
139
140 5 4
141 Enter elements of Matrix B (2x2):
142 2 3
143 4 5
144
145 1.Addition
146 2.Subtraction
147 3.Multiplication
148 4.Traversal
149 Enter choice: 2
150
151 Subtraction:
152 -1 -1
153 1 -1
154
155 5 4
156 Enter elements of Matrix B (2x2):
157 2 3
158 4 5
159
160 1.Addition
161 2.Subtraction
162 3.Multiplication
163 4.Traversal
164 Enter choice: 2
165
166 Subtraction:
167 -1 -1
168 1 -1
169
170 5 4
171 Enter elements of Matrix B (2x2):
172 2 3
173 4 5
174
175 1.Addition
176 2.Subtraction
177 3.Multiplication
178 4.Traversal
179 Enter choice: 2
180
181 Subtraction:
182 -1 -1
183 1 -1
184
185 5 4
186 Enter elements of Matrix B (2x2):
187 2 3
188 4 5
189
190 1.Addition
191 2.Subtraction
192 3.Multiplication
193 4.Traversal
194 Enter choice: 2
195
196 Subtraction:
197 -1 -1
198 1 -1
199
200 5 4
201 Enter elements of Matrix B (2x2):
202 2 3
203 4 5
204
205 1.Addition
206 2.Subtraction
207 3.Multiplication
208 4.Traversal
209 Enter choice: 2
210
211 Subtraction:
212 -1 -1
213 1 -1
214
215 5 4
216 Enter elements of Matrix B (2x2):
217 2 3
218 4 5
219
220 1.Addition
221 2.Subtraction
222 3.Multiplication
223 4.Traversal
224 Enter choice: 2
225
226 Subtraction:
227 -1 -1
228 1 -1
229
230 5 4
231 Enter elements of Matrix B (2x2):
232 2 3
233 4 5
234
235 1.Addition
236 2.Subtraction
237 3.Multiplication
238 4.Traversal
239 Enter choice: 2
240
241 Subtraction:
242 -1 -1
243 1 -1
244
245 5 4
246 Enter elements of Matrix B (2x2):
247 2 3
248 4 5
249
250 1.Addition
251 2.Subtraction
252 3.Multiplication
253 4.Traversal
254 Enter choice: 2
255
256 Subtraction:
257 -1 -1
258 1 -1
259
260 5 4
261 Enter elements of Matrix B (2x2):
262 2 3
263 4 5
264
265 1.Addition
266 2.Subtraction
267 3.Multiplication
268 4.Traversal
269 Enter choice: 2
270
271 Subtraction:
272 -1 -1
273 1 -1
274
275 5 4
276 Enter elements of Matrix B (2x2):
277 2 3
278 4 5
279
280 1.Addition
281 2.Subtraction
282 3.Multiplication
283 4.Traversal
284 Enter choice: 2
285
286 Subtraction:
287 -1 -1
288 1 -1
289
290 5 4
291 Enter elements of Matrix B (2x2):
292 2 3
293 4 5
294
295 1.Addition
296 2.Subtraction
297 3.Multiplication
298 4.Traversal
299 Enter choice: 2
300
301 Subtraction:
302 -1 -1
303 1 -1
304
305 5 4
306 Enter elements of Matrix B (2x2):
307 2 3
308 4 5
309
310 1.Addition
311 2.Subtraction
312 3.Multiplication
313 4.Traversal
314 Enter choice: 2
315
316 Subtraction:
317 -1 -1
318 1 -1
319
320 5 4
321 Enter elements of Matrix B (2x2):
322 2 3
323 4 5
324
325 1.Addition
326 2.Subtraction
327 3.Multiplication
328 4.Traversal
329 Enter choice: 2
330
331 Subtraction:
332 -1 -1
333 1 -1
334
335 5 4
336 Enter elements of Matrix B (2x2):
337 2 3
338 4 5
339
340 1.Addition
341 2.Subtraction
342 3.Multiplication
343 4.Traversal
344 Enter choice: 2
345
346 Subtraction:
347 -1 -1
348 1 -1
349
350 5 4
351 Enter elements of Matrix B (2x2):
352 2 3
353 4 5
354
355 1.Addition
356 2.Subtraction
357 3.Multiplication
358 4.Traversal
359 Enter choice: 2
360
361 Subtraction:
362 -1 -1
363 1 -1
364
365 5 4
366 Enter elements of Matrix B (2x2):
367 2 3
368 4 5
369
370 1.Addition
371 2.Subtraction
372 3.Multiplication
373 4.Traversal
374 Enter choice: 2
375
376 Subtraction:
377 -1 -1
378 1 -1
379
380 5 4
381 Enter elements of Matrix B (2x2):
382 2 3
383 4 5
384
385 1.Addition
386 2.Subtraction
387 3.Multiplication
388 4.Traversal
389 Enter choice: 2
390
391 Subtraction:
392 -1 -1
393 1 -1
394
395 5 4
396 Enter elements of Matrix B (2x2):
397 2 3
398 4 5
399
400 1.Addition
401 2.Subtraction
402 3.Multiplication
403 4.Traversal
404 Enter choice: 2
405
406 Subtraction:
407 -1 -1
408 1 -1
409
410 5 4
411 Enter elements of Matrix B (2x2):
412 2 3
413 4 5
414
415 1.Addition
416 2.Subtraction
417 3.Multiplication
418 4.Traversal
419 Enter choice: 2
420
421 Subtraction:
422 -1 -1
423 1 -1
424
425 5 4
426 Enter elements of Matrix B (2x2):
427 2 3
428 4 5
429
430 1.Addition
431 2.Subtraction
432 3.Multiplication
433 4.Traversal
434 Enter choice: 2
435
436 Subtraction:
437 -1 -1
438 1 -1
439
440 5 4
441 Enter elements of Matrix B (2x2):
442 2 3
443 4 5
444
445 1.Addition
446 2.Subtraction
447 3.Multiplication
448 4.Traversal
449 Enter choice: 2
450
451 Subtraction:
452 -1 -1
453 1 -1
454
455 5 4
456 Enter elements of Matrix B (2x2):
457 2 3
458 4 5
459
460 1.Addition
461 2.Subtraction
462 3.Multiplication
463 4.Traversal
464 Enter choice: 2
465
466 Subtraction:
467 -1 -1
468 1 -1
469
470 5 4
471 Enter elements of Matrix B (2x2):
472 2 3
473 4 5
474
475 1.Addition
476 2.Subtraction
477 3.Multiplication
478 4.Traversal
479 Enter choice: 2
480
481 Subtraction:
482 -1 -1
483 1 -1
484
485 5 4
486 Enter elements of Matrix B (2x2):
487 2 3
488 4 5
489
490 1.Addition
491 2.Subtraction
492 3.Multiplication
493 4.Traversal
494 Enter choice: 2
495
496 Subtraction:
497 -1 -1
498 1 -1
499
500 5 4
501 Enter elements of Matrix B (2x2):
502 2 3
503 4 5
504
505 1.Addition
506 2.Subtraction
507 3.Multiplication
508 4.Traversal
509 Enter choice: 2
510
511 Subtraction:
512 -1 -1
513 1 -1
514
515 5 4
516 Enter elements of Matrix B (2x2):
517 2 3
518 4 5
519
520 1.Addition
521 2.Subtraction
522 3.Multiplication
523 4.Traversal
524 Enter choice: 2
525
526 Subtraction:
527 -1 -1
528 1 -1
529
530 5 4
531 Enter elements of Matrix B (2x2):
532 2 3
533 4 5
534
535 1.Addition
536 2.Subtraction
537 3.Multiplication
538 4.Traversal
539 Enter choice: 2
540
541 Subtraction:
542 -1 -1
543 1 -1
544
545 5 4
546 Enter elements of Matrix B (2x2):
547 2 3
548 4 5
549
550 1.Addition
551 2.Subtraction
552 3.Multiplication
553 4.Traversal
554 Enter choice: 2
555
556 Subtraction:
557 -1 -1
558 1 -1
559
560 5 4
561 Enter elements of Matrix B (2x2):
562 2 3
563 4 5
564
565 1.Addition
566 2.Subtraction
567 3.Multiplication
568 4.Traversal
569 Enter choice: 2
570
571 Subtraction:
572 -1 -1
573 1 -1
574
575 5 4
576 Enter elements of Matrix B (2x2):
577 2 3
578 4 5
579
580 1.Addition
581 2.Subtraction
582 3.Multiplication
583 4.Traversal
584 Enter choice: 2
585
586 Subtraction:
587 -1 -1
588 1 -1
589
590 5 4
591 Enter elements of Matrix B (2x2):
592 2 3
593 4 5
594
595 1.Addition
596 2.Subtraction
597 3.Multiplication
598 4.Traversal
599 Enter choice: 2
600
601 Subtraction:
602 -1 -1
603 1 -1
604
605 5 4
606 Enter elements of Matrix B (2x2):
607 2 3
608 4 5
609
610 1.Addition
611 2.Subtraction
612 3.Multiplication
613 4.Traversal
614 Enter choice: 2
615
616 Subtraction:
617 -1 -1
618 1 -1
619
620 5 4
621 Enter elements of Matrix B (2x2):
622 2 3
623 4 5
624
625 1.Addition
626 2.Subtraction
627 3.Multiplication
628 4.Traversal
629 Enter choice: 2
630
631 Subtraction:
632 -1 -1
633 1 -1
634
635 5 4
636 Enter elements of Matrix B (2x2):
637 2 3
638 4 5
639
640 1.Addition
641 2.Subtraction
642 3.Multiplication
643 4.Traversal
644 Enter choice: 2
645
646 Subtraction:
647 -1 -1
648 1 -1
649
650 5 4
651 Enter elements of Matrix B (2x2):
652 2 3
653 4 5
654
655 1.Addition
656 2.Subtraction
657 3.Multiplication
658 4.Traversal
659 Enter choice: 2
660
661 Subtraction:
662 -1 -1
663 1 -1
664
665 5 4
666 Enter elements of Matrix B (2x2):
667 2 3
668 4 5
669
670 1.Addition
671 2.Subtraction
672 3.Multiplication
673 4.Traversal
674 Enter choice: 2
675
676 Subtraction:
677 -1 -1
678 1 -1
679
680 5 4
681 Enter elements of Matrix B (2x2):
682 2 3
683 4 5
684
685 1.Addition
686 2.Subtraction
687 3.Multiplication
688 4.Traversal
689 Enter choice: 2
690
691 Subtraction:
692 -1 -1
693 1 -1
694
695 5 4
696 Enter elements of Matrix B (2x2):
697 2 3
698 4 5
699
700 1.Addition
701 2.Subtraction
702 3.Multiplication
703 4.Traversal
704 Enter choice: 2
705
706 Subtraction:
707 -1 -1
708 1 -1
709
710 5 4
711 Enter elements of Matrix B (2x2):
712 2 3
713 4 5
714
715 1.Addition
716 2.Subtraction
717 3.Multiplication
718 4.Traversal
719 Enter choice: 2
720
721 Subtraction:
722 -1 -1
723 1 -1
724
725 5 4
726 Enter elements of Matrix B (2x2):
727 2 3
728 4 5
729
730 1.Addition
731 2.Subtraction
732 3.Multiplication
733 4.Traversal
734 Enter choice: 2
735
736 Subtraction:
737 -1 -1
738 1 -1
739
740 5 4
741 Enter elements of Matrix B (2x2):
742 2 3
743 4 5
744
745 1.Addition
746 2.Subtraction
747 3.Multiplication
748 4.Traversal
749 Enter choice: 2
750
751 Subtraction:
752 -1 -1
753 1 -1
754
755 5 4
756 Enter elements of Matrix B (2x2):
757 2 3
758 4 5
759
760 1.Addition
761 2.Subtraction
762 3.Multiplication
763 4.Traversal
764 Enter choice: 2
765
766 Subtraction:
767 -1 -1
768 1 -1
769
770 5 4
771 Enter elements of Matrix B (2x2):
772 2 3
773 4 5
774
775 1.Addition
776 2.Subtraction
777 3.Multiplication
778 4.Traversal
779 Enter choice: 2
780
781 Subtraction:
782 -1 -1
783 1 -1
784
785 5 4
786 Enter elements of Matrix B (2x2):
787 2 3
788 4 5
789
790 1.Addition
791 2.Subtraction
792 3.Multiplication
793 4.Traversal
794 Enter choice: 2
795
796 Subtraction:
797 -1 -1
798 1 -1
799
800 5 4
801 Enter elements of Matrix B (2x2):
802 2 3
803 4 5
804
805 1.Addition
806 2.Subtraction
807 3.Multiplication
808 4.Traversal
809 Enter choice: 2
810
811 Subtraction:
812 -1 -1
813 1 -1
814
815 5 4
816 Enter elements of Matrix B (2x2):
817 2 3
818 4 5
819
820 1.Addition
821 2.Subtraction
822 3.Multiplication
823 4.Traversal
824 Enter choice: 2
825
826 Subtraction:
827 -1 -1
828 1 -1
829
830 5 4
831 Enter elements of Matrix B (2x2):
832 2 3
833 4 5
834
835 1.Addition
836 2.Subtraction
837 3.Multiplication
838 4.Traversal
839 Enter choice: 2
840
841 Subtraction:
842 -1 -1
843 1 -1
844
845 5 4
846 Enter elements of Matrix B (2x2):
847 2 3
848 4 5
849
850 1.Addition
851 2.Subtraction
852 3.Multiplication
853 4.Traversal
854 Enter choice: 2
855
856 Subtraction:
857 -1 -1
858 1 -1
859
860 5 4
861 Enter elements of Matrix B (2x2):
862 2 3
863 4 5
864
865 1.Addition
866 2.Subtraction
867 3.Multiplication
868 4.Traversal
869 Enter choice: 2
870
871 Subtraction:
872 -1 -1
873 1 -1
874
875 5 4
876 Enter elements of Matrix B (2x2):
877 2 3
878 4 5
879
880 1.Addition
881 2.Subtraction
882 3.Multiplication
883 4.Traversal
884 Enter choice: 2
885
886 Subtraction:
887 -1 -1
888 1 -1
889
890 5 4
891 Enter elements of Matrix B (2x2):
892 2 3
893 4 5
894
895 1.Addition
896 2.Subtraction
897 3.Multiplication
898 4.Traversal
899 Enter choice: 2
900
901 Subtraction:
902 -1 -1
903 1 -1
904
905 5 4
906 Enter elements of Matrix B (2x2):
907 2 3
908 4 5
909
910 1.Addition
911 2.Subtraction
912 3.Multiplication
913 4.Traversal
914 Enter choice: 2
915
916 Subtraction:
917 -1 -1
918 1 -1
919
920 5 4
921 Enter elements of Matrix B (2x2):
922 2 3
923 4 5
924
925 1.Addition
926 2.Subtraction
927 3.Multiplication
928 4.Traversal
929 Enter choice: 2
930
931 Subtraction:
932 -1 -1
933 1 -1
934
935 5 4
936 Enter elements of Matrix B (2x2):
937 2 3
938 4 5
939
940 1.Addition
941 2.Subtraction
942 3.Multiplication
943 4.Traversal
944 Enter choice: 2
945
946 Subtraction:
947 -1 -1
948 1 -1
949
950 5 4
951 Enter elements of Matrix B (2x2):
952 2 3
953 4 5
954
955 1.Addition
956 2.Subtraction
957 3.Multiplication
958 4.Traversal
959 Enter choice: 2
960
961 Subtraction:
962 -1 -1
963 1 -1
964
965 5 4
966 Enter elements of Matrix B (2x2):
967 2 3
968 4 5
969
970 1.Addition
971 2.Subtraction
972 3.Multiplication
973 4.Traversal
974 Enter choice: 2
975
976 Subtraction:
977 -1 -1
978 1 -1
979
980 5 4
981 Enter elements of Matrix B (2x2):
982 2 3
983 4 5
984
985 1.Addition
986 2.Subtraction
987 3.Multiplication
988 4.Traversal
989 Enter choice: 2
990
991 Subtraction:
992 -1 -1
993 1 -1
994
995 5 4
996 Enter elements of Matrix B (2x2):
997 2 3
998 4 5
999
1000 1.Addition
1001 2.Subtraction
1002 3.Multiplication
1003 4.Traversal
1004 Enter choice: 2
1005
1006 Subtraction:
1007 -1 -1
1008 1 -1
1009
1010 5 4
1011 Enter elements of Matrix B (2x2):
1012 2 3
1013 4 5
1014
1015 1.Addition
1016 2.Subtraction
1017 3.Multiplication
1018 4.Traversal
1019 Enter choice: 2
1020
1021 Subtraction:
1022 -1 -1
1023 1 -1
1024
1025 5 4
1026 Enter elements of Matrix B (2x2):
1027 2 3
1028 4 5
1029
1030 1.Addition
1031 2.Subtraction
1032 3.Multiplication
1033 4.Traversal
1034 Enter choice: 2
1035
1036 Subtraction:
1037 -1 -1
1038 1 -1
1039
1040 5 4
1041 Enter elements of Matrix B (2x2):
1042 2 3
1043 4 5
1044
1045 1.Addition
1046 2.Subtraction
1047 3.Multiplication
1048 4.Traversal
1049 Enter choice: 2
1050
1051 Subtraction:
1052 -1 -1
1053 1 -1
1054
1055 5 4
1056 Enter elements of Matrix B (2x2):
1057 2 3
1058 4 5
1059
1060 1.Addition
1061 2.Subtraction
1062 3.Multiplication
1063 4.Traversal
1064 Enter choice: 2
1065
1066 Subtraction:
1067 -1 -1
1068 1 -1
1069
1070 5 4
1071 Enter elements of Matrix B (2x2):
1072 2 3
1073 4 5
1074
1075 1.Addition
1076 2.Subtraction
1077 3.Multiplication
1078 4.Traversal
1079 Enter choice: 2
1080
1081 Subtraction:
1082 -1 -1
1083 1 -1
1084
1085 5 4
1086 Enter elements of Matrix B (2x2):
1087 2 3
1088 4 5
1089
1090 1.Addition
1091 2.Subtraction
1092 3.Multiplication
1093 4.Traversal
1094 Enter choice: 2
1095
1096 Subtraction:
1097 -1 -1
1098 1 -1
1099
1100 5 4
1101 Enter elements of Matrix B (2x2):
1102 2 3
1103 4 5
1104
1105 1.Addition
1106 2.Subtraction
1107 3.Multiplication
1108 4.Traversal
1109 Enter choice: 2
1110
1111 Subtraction:
1112 -1 -1
1113 1 -1
1114
1115 5 4
1116 Enter elements of Matrix B (2x2):
1117 2 3
1118 4 5
1119
1120 1.Addition
1121 2.Subtraction
1122 3.Multiplication
1123 4.Traversal
1124 Enter choice: 2
1125
1126 Subtraction:
1127 -1 -1
1128 1 -1
1129
1130 5 4
1131 Enter elements of Matrix B (2x2):
1132 2 3
1133 4 5
1134
1135 1.Addition
1136 2.Subtraction
1137 3.Multiplication
1138 4.Traversal
1139 Enter choice: 2
1140
1141 Subtraction:
1142 -1 -1
1143 1 -1
1144
1145 5 4
1146 Enter elements of Matrix B (2x2):
1147 2 3
1148 4 5
1149
1150 1.Addition
1151 2.Subtraction
1152 3.Multiplication
1153 4.Traversal
1154 Enter choice: 2
1155
1156 Subtraction:
1157 -1 -1
1158 1 -1
1159
1160 5 4
1161 Enter elements of Matrix B (2x2):
1162 2 3
1163 4 5
1164
1165 1.Addition
1166 2.Subtraction
1167 3.Multiplication
1168 4.Traversal
1169 Enter choice: 2
1170
1171 Subtraction:
1172 -1 -1
1173 1 -1
1174
1175 5 4
1176 Enter elements of Matrix B (2x2):
1177 2 3
1178 4 5
1179
1180 1.Addition
1181 2.Subtraction
1182 3.Multiplication
1183 4.Traversal
1184 Enter choice: 2
1185
1186 Subtraction:
1187 -1 -1
1188 1 -1
1189
1190 5 4
1191 Enter elements of Matrix B (2x2):
1192 2 3
1193 4 5
1194
1195 1.Addition
1196 2.Subtraction
1197 3.Multiplication
1198 4.Traversal
1199 Enter choice: 2
1200
1201 Subtraction:
1202 -1 -1
1203 1 -1
1204
1205 5 4
1206 Enter elements of Matrix B (2x2):
1207 2 3
1208 4 5
1209
1210 1.Addition
1211 2.Subtraction
1212 3.Multiplication
1213 4.Traversal
1214 Enter choice: 2
1215
1216 Subtraction:
1217 -1 -1
1218 1 -1
1219
1220 5 4
1221 Enter elements of Matrix B (2x2):
1222 2 3
1223 4 5
1224
1225 1.Addition
1226 2.Subtraction
1227 3.Multiplication
1228 4.Traversal
1229 Enter choice: 2
1230
1231 Subtraction:
1232 -1 -1
1233 1 -1
1234
1235 5 4
1236 Enter elements of Matrix B (2x2):
1237 2 3
1238 4 5
1239
1240 1.Addition
1241 2.Subtraction
1242 3.Multiplication
1243 4.Traversal
1244 Enter choice: 2
1245
1246 Subtraction:
1247 -1 -1
1248 1 -1
1249
1250 5 4
1251 Enter elements of Matrix B (2x2):
1252 2 3
1253 4 5
1254
1255 1.Addition
1256 2.Subtraction
1257 3.Multiplication
1258 4.Traversal
1259 Enter choice: 2
1260
1261 Subtraction:
1262 -1 -1
1263 1 -1
1264
1265 5 4
1266 Enter elements of Matrix B (2x2):
1267 2 3
1268 4 5
1269
1270 1.Addition
1271 2.Subtraction
1272 3.Multiplication
1273 4.Traversal
1274 Enter choice: 2
1275
1276 Subtraction:
1277 -1 -1
1278 1 -1
1279
1280 5 4
1281 Enter elements of Matrix B (2x2):
1282 2 3
1283 4 5
1284
1285 1.Addition
1286 2.Subtraction
1287 3.Multiplication
1288 4.Traversal
1289 Enter choice: 2
1290
1291 Subtraction:
1292 -1 -1
1293 1 -1
1294
1295 5 4
1296 Enter elements of Matrix B (2x2):
1297 2 3
1298 4 5
1299
1300 1.Addition
1301 2.Subtraction
1302 3.Multiplication
1303 4.Traversal
1304 Enter choice: 2
1305
1306 Subtraction:
1307 -1 -1
1308 1 -1
1309
1310 5 4
1311 Enter elements of Matrix B (2x2):
1312 2 3
1313 4 5
1314
1315 1.Addition
1316 2.Subtraction
1317 3.Multiplication
1318 4.Traversal
1319 Enter choice: 2
1320
1321 Subtraction:
1322 -1 -1
1323 1 -1
1324
1325 5 4
1326 Enter elements of Matrix B (2x2):
1327 2 3
1328 4 5
1329
1330 1.Addition
1331 2.Subtraction
1332 3.Multiplication
1333 4.Traversal
1334 Enter choice: 2
1335
1336 Subtraction:
1337 -1 -1
1338 1 -1
1339
1340 5 4
1341 Enter elements of Matrix B (2x2):
1342 2 3
1343 4 5
1344
1345 1.Addition
1346 2.Subtraction
1347 3.Multiplication
1348 4.Traversal
1349 Enter choice: 2
1350
1351 Subtraction:
1352 -1 -1
1353 1 -1
1354
1355 5 4
1356 Enter elements of Matrix B (2x2):
1357 2 3
1358 4 5
1359
1360 1.Addition
1361 2.Subtraction
1362 3.Multiplication
1363 4.Traversal
1364 Enter choice: 2
1365
1366 Subtraction:
1367 -1 -1
1368 1 -1
1369
1370 5 4
1371 Enter elements of Matrix B (2x2):
1372 2 3
1373 4 5
1374
1375 1.Addition
1376 2.Subtraction
1377 3.Multiplication
1378 4.Traversal
1379 Enter choice: 2
1380
1381 Subtraction:
1382 -1 -1
1383 1 -1
1384
1385 5 4
1386 Enter elements of Matrix B (2x2):
1387 2 3
1388 4 5
1389
1390 1.Addition
1391 2.Subtraction
1392 3.Multiplication
1393 4.Traversal
1394 Enter choice: 2
1395
1396 Subtraction:
1397 -1 -1
1398 1 -1
1399
1400 5 4
1401 Enter elements of Matrix B (2x2):
1402 2 3
1403 4 5
1404
1405 1.Addition
1406 2.Subtraction
1407 3.Multiplication
1408 4.Traversal
1409 Enter choice: 2
1410
1411 Subtraction:
1412 -1 -1
1413 1 -1
1414
1415 5 4
1416 Enter elements of Matrix B (2x2):
1417 2 3
1418 4 5
1419
1420 1.Addition
1421 2.Subtraction
1422 3.Multiplication
1423 4.Traversal
1424 Enter choice: 2
1425
1426 Subtraction:
1427 -1 -1
1428 1 -1
1429
1430 5 4
1431 Enter
```

```
61     for(i=0;i<2;i++){
62         for(j=0;j<2;j++){
63             printf("%d ",b[i][j]);
64             printf("\n");
65         }
66     }
67 else
68     printf("Invalid choice!");
69
70 return 0;
71 }
72
73
74
75
76
```

5 4  
Enter elements of Matrix B (2x2):  
2 3  
4 5  
  
1.Addition  
2.Subtraction  
3.Multiplication  
4.Traversal  
Enter choice: 2  
  
Subtraction:  
-1 -1  
1 -1