



Mega Naukri

A Career Platform for You

WWW.MEGANAUKRI.COM

FOLLOW OUR FACEBOOK PAGE- WWW.FACEBOOK.COM/MEGANAUKRI

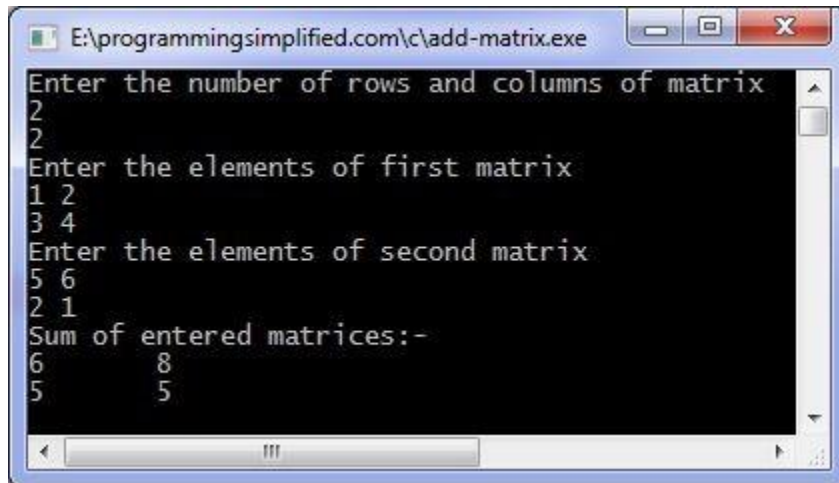
FOLLOW US ON TELEGRAM- [HTTP://TELEGRAM.ME/OFFCAMPUSJOBS_FRESHERS](http://TELEGRAM.ME/OFFCAMPUSJOBS_FRESHERS)

FOLLOW US ON LINKEDIN - [HTTPS://WWW.LINKEDIN.COM/IN/MEGANAUKRI](https://WWW.LINKEDIN.COM/IN/MEGANAUKRI)

1. Addition Of Two Matrices In C:

```
1. #include <stdio.h>
2.
3. int main()
4. {
5.     int m, n, c, d, first[10][10], second[10][10], sum[10][10];
6.
7.     printf("Enter the number of rows and columns of matrix\n");
8.     scanf("%d%d", &m, &n);
9.     printf("Enter the elements of first matrix\n");
10.
11.     for (c = 0; c < m; c++)
12.         for (d = 0; d < n; d++)
13.             scanf("%d", &first[c][d]);
14.
15.     printf("Enter the elements of second matrix\n");
16.
17.     for (c = 0; c < m; c++)
18.         for (d = 0; d < n; d++)
19.             scanf("%d", &second[c][d]);
20.
21.     printf("Sum of entered matrices:-\n");
22.
23.     for (c = 0; c < m; c++) {
24.         for (d = 0; d < n; d++) {
25.             sum[c][d] = first[c][d] + second[c][d];
26.             printf("%d\t", sum[c][d]);
27.         }
28.         printf("\n");
29.     }
30.
31.     return 0;
32. }
```

Output:



The screenshot shows a Windows command prompt window titled "E:\programmingsimplified.com\c\add-matrix.exe". The text inside the window is as follows:

```
Enter the number of rows and columns of matrix
2
2
Enter the elements of first matrix
1 2
3 4
Enter the elements of second matrix
5 6
2 1
Sum of entered matrices:-
6      8
5      5
```

2. Program to find the average of n ($n < 10$) numbers using arrays

```
#include <stdio.h>
int main()
{
    int marks[10], i, n, sum = 0, average;
    printf("Enter n: ");
    scanf("%d", &n);
    for(i=0; i<n; ++i)
    {
        printf("Enter number%d: ",i+1);
        scanf("%d", &marks[i]);
        sum += marks[i];
    }
    average = sum/n;

    printf("Average = %d", average);

    return 0;
}
```

Output:

```
Enter n: 5
Enter number1: 45
Enter number2: 35
Enter number3: 38
Enter number4: 31
Enter number5: 49
Average = 39
```

3. C program To Implement Linked List

```
1. #include <stdio.h>
2. #include <stdlib.h>
3.
4. struct node {
5.     int data;
6.     struct node *next;
7. };
8.
9. struct node *start = NULL;
10. void insert_at_begin(int);
11. void insert_at_end(int);
12. void traverse();
13. void delete_from_begin();
14. void delete_from_end();
15. int count = 0;
16.
17. int main () {
18.     int input, data;
19.
20.     for (;;) {
21.         printf("1. Insert an element at beginning of linked list.\n");
22.         printf("2. Insert an element at end of linked list.\n");
23.         printf("3. Traverse linked list.\n");
24.         printf("4. Delete element from beginning.\n");
```

```
25.     printf("5. Delete element from end.\n");
26.     printf("6. Exit\n");
27.
28.     scanf("%d", &input);
29.
30.     if (input == 1) {
31.         printf("Enter value of element\n");
32.         scanf("%d", &data);
33.         insert_at_begin(data);
34.     }
35.     else if (input == 2) {
36.         printf("Enter value of element\n");
37.         scanf("%d", &data);
38.         insert_at_end(data);
39.     }
40.     else if (input == 3)
41.         traverse();
42.     else if (input == 4)
43.         delete_from_begin();
44.     else if (input == 5)
45.         delete_from_end();
46.     else if (input == 6)
47.         break;
48.     else
49.         printf("Please enter valid input.\n");
50. }
51.
52. return 0;
53. }
54.
55. void insert_at_begin(int x) {
56.     struct node *t;
57.
58.     t = (struct node*)malloc(sizeof(struct node));
59.     count++;
60.
```

```
61.     if (start == NULL) {
62.         start = t;
63.         start->data = x;
64.         start->next = NULL;
65.         return;
66.     }
67.
68.     t->data = x;
69.     t->next = start;
70.     start = t;
71. }
72.
73. void insert_at_end(int x) {
74.     struct node *t, *temp;
75.
76.     t = (struct node*)malloc(sizeof(struct node));
77.     count++;
78.
79.     if (start == NULL) {
80.         start = t;
81.         start->data = x;
82.         start->next = NULL;
83.         return;
84.     }
85.
86.     temp = start;
87.
88.     while (temp->next != NULL)
89.         temp = temp->next;
90.
91.     temp->next = t;
92.     t->data = x;
93.     t->next = NULL;
94. }
95.
96. void traverse() {
```

```
97.     struct node *t;
98.
99.     t = start;
100.
101.     if (t == NULL) {
102.         printf("Linked list is empty.\n");
103.         return;
104.     }
105.
106.     printf("There are %d elements in linked list.\n", count);
107.
108.     while (t->next != NULL) {
109.         printf("%d\n", t->data);
110.         t = t->next;
111.     }
112.     printf("%d\n", t->data);
113. }
114.
115. void delete_from_begin() {
116.     struct node *t;
117.     int n;
118.
119.     if (start == NULL) {
120.         printf("Linked list is already empty.\n");
121.         return;
122.     }
123.
124.     n = start->data;
125.     t = start->next;
126.     free(start);
127.     start = t;
128.     count--;
129.
130.     printf("%d deleted from beginning successfully.\n", n);
131. }
132.
```

```

133. void delete_from_end() {
134.     struct node *t, *u;
135.     int n;
136.
137.     if (start == NULL) {
138.         printf("Linked list is already empty.\n");
139.         return;
140.     }
141.
142.     count--;
143.
144.     if (start->next == NULL) {
145.         n = start->data;
146.         free(start);
147.         start = NULL;
148.         printf("%d deleted from end successfully.\n", n);
149.         return;
150.     }
151.
152.     t = start;
153.
154.     while (t->next != NULL) {
155.         u = t;
156.         t = t->next;
157.     }
158.
159.     n = t->data;
160.     u->next = NULL;
161.     free(t);
162.
163.     printf("%d deleted from end successfully.\n", n);
164. }

```

4. Operations On Linked List


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

struct node
{
    int data;
    struct node *next;
};

void display(struct node* head)
{
    struct node *temp = head;
    printf("\n\nList elements are - \n");
    while(temp != NULL)
    {
        printf("%d --->",temp->data);
        temp = temp->next;
    }
}
```

[Buy full course of TCS NQT here](https://meganaukri.com/tcs-nqt-premium-preparation-material)

<https://meganaukri.com/tcs-nqt-premium-preparation-material>

Expect 60%-70% similar question in TCS NQT Exam

This Product Includes all the questions asked in TCS NQT

- Coding Question
- Numerical Ability
- Programming logic
- Verbal Ability
- Reasoning ability
- Sample Papers
- Quantitative Aptitude

Complimentary Bonus

600+ Technical Programming and Coding Questions

TCS HR interview handwritten notes

