### DS LAB FILE MAIN

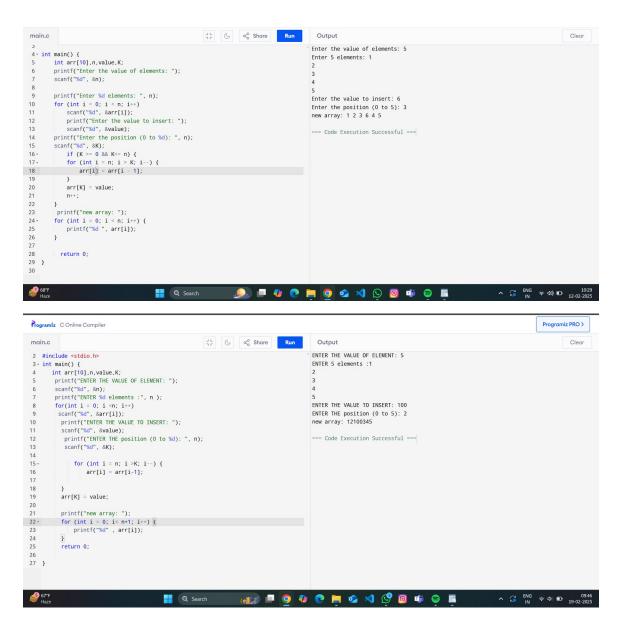
# 1. Array and sum and average

```
[] ← C Share Run
                                                                                                Output
          1 // Online C compiler to run C program online
                                                                                               Enter the value of n: 5
          2 #include <stdio.h>
                                                                                               Enter the marks: 65
         4 * int main() {
5     int marks[10],
                   sum=0, avg,i,n;
5
                  printf("Enter the value of n: ");
scanf("%d" ,&n);
                                                                                               sum is 343
                                                                                               avg is 68
                 printf("Enter the marks: ");
                 for( i=0; i<n; i++)
scanf("%d", &marks[i]);</pre>
0
                                                                                               === Code Execution Successful ===
                  for (i = 0; i<n; i++)
0
                 sum = sum + marks[i];
printf("sum is %d\n" ,sum);
                 avg = sum/n;
printf("avg is %d\n" ,avg);
                  return 0;
```

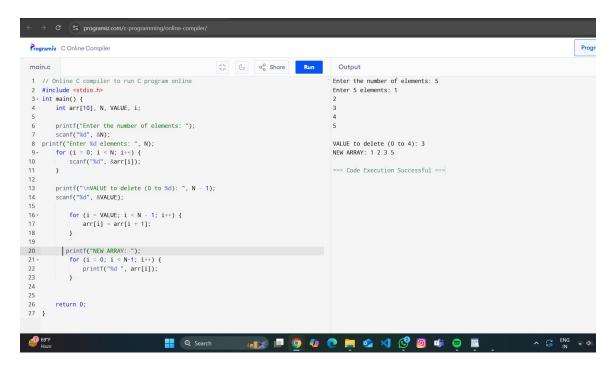
## 2. trasversing

```
Premium Coding
    Programiz
                                                                                             Programiz PRO
    C Online Compiler
                                       Courses by Programiz
                                       [] G Share Run
      1 // Online C compiler to run C program online
                                                                   Enter the value of elements: 5
P
      2 #include <stdio.h>
      4 • int main() {
int arr[10];
5
            printf("Enter the value of elements: ");
                                                                   Transversing a elements: 1
            scanf("%d", &n);
            printf("Enter %d elements: ", n);
           0
0
0
      16
             return 0:
JS
```

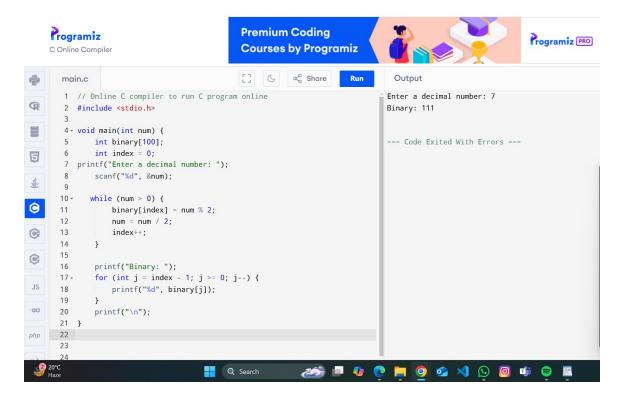
### 2. insertion



## **DELETION**



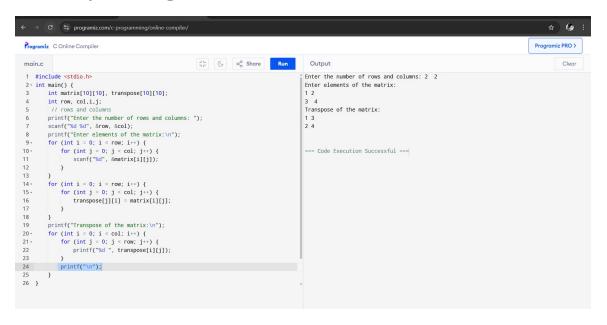
## binary to decimal



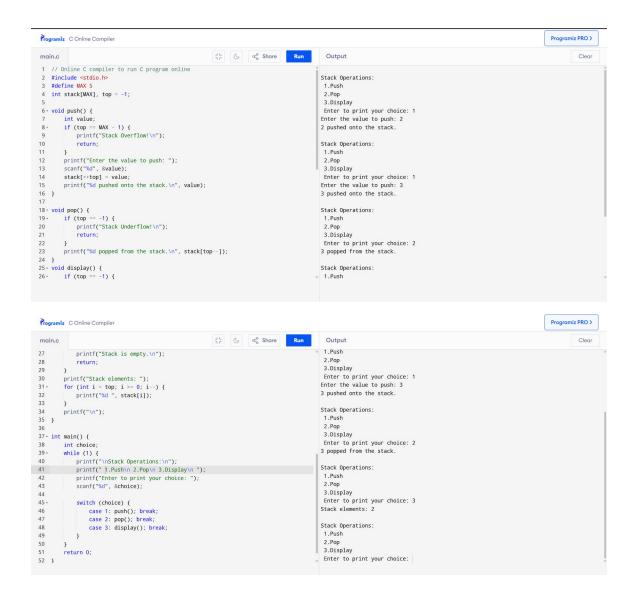
### **2D MULTIPLICATION**

```
Programiz C Online Compiler
                                                                               Share Run
                                                                                                                                Output
   1 // Online C compiler to run C program online
                                                                                                                               Matrix result:
   2 #include <stdio.h>
                                                                                                                               19 22
3 - int main() {
                                                                                                                              43 50
            int A[2][2] = { {1, 2}, {3, 4} };
int B[2][2] = { {5, 6}, {7, 8} };
            int fesult[2][2];
for (int i = 0; i < 2; i++) {
    for (int j = 0; j < 2; j++) {
        result[i][j] = 0;
    }</pre>
                                                                                                                               === Code Execution Successful ===
  11
12 -
            13 -
  14 -
 16
17
            printf("Matrix result:\n");
for (int i = 0; i < 2; i++) {
    for (int j = 0; j < 2; j++) {
        printf("%d ", result[i][j]);
}</pre>
  19
20 -
```

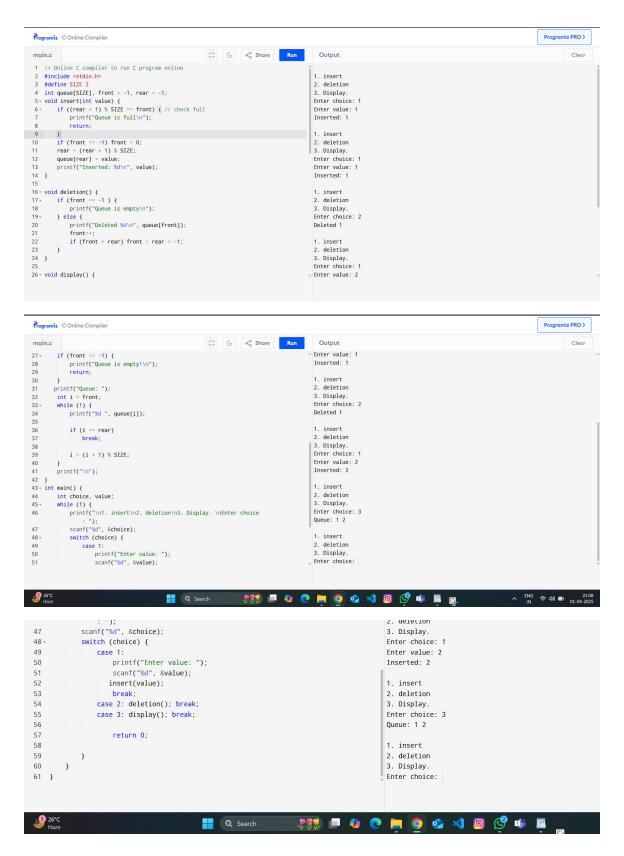
## transposing



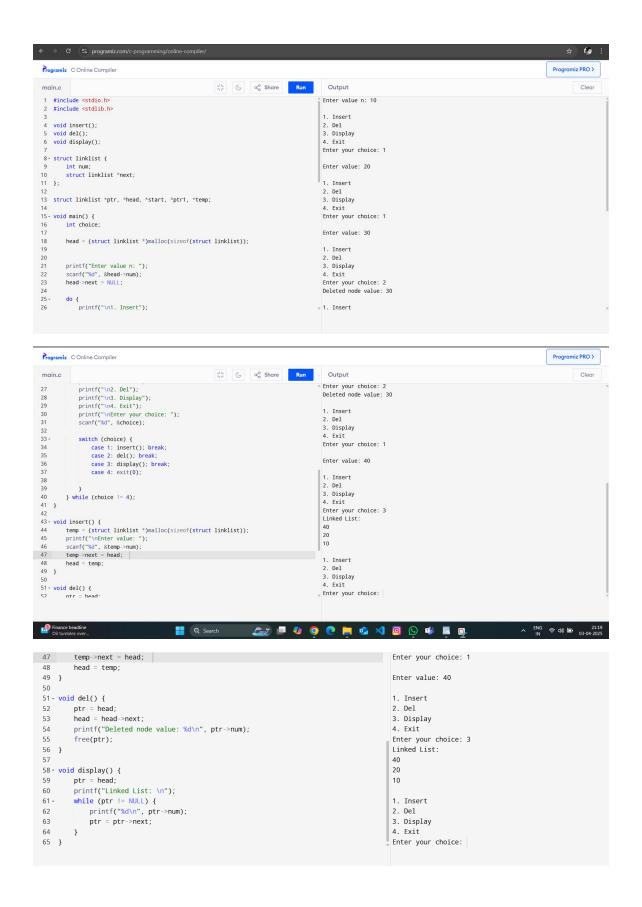
stack push and pop



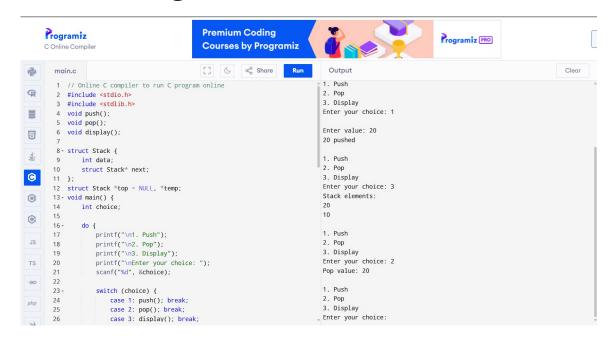
### queue



Linked list operation



# stack through linked list



#### Programiz C Online Compiler

```
לב כ כ Share
                                                                              Run
main.c
                                                                                         Output
                                                                                       1. Push
27
                                                                                       2. Pop
       } while (choice != 4);
28
                                                                                        3. Display
29 }
                                                                                        Enter your choice: 1
30 - void push() {
        temp = (struct Stack*)malloc(sizeof(struct Stack));
31
                                                                                        Enter value: 20
32
                                                                                       20 pushed
33 -
        if (temp == NULL) {
           printf("Memory allocation failed!\n");\\
34
                                                                                       1. Push
35
           return;
                                                                                       2. Pop
36
                                                                                       3. Display
37
                                                                                       Enter your choice: 3
38
       printf("\nEnter value: ");
                                                                                       Stack elements:
       scanf("%d", &temp->data);
39
                                                                                       20
40
                                                                                        10
       temp->next = top;
42
       top = temp;
                                                                                        1. Push
43
                                                                                       2. Pop
44
       printf("%d pushed\n", temp->data);
                                                                                        3. Display
45 }
                                                                                       Enter your choice: 2
46 - void pop() {
                                                                                        Pop value: 20
       if (top == NULL) {
47 -
          printf("Stack Underflow \n");
48
                                                                                        1. Push
49
           return;
                                                                                       2. Pop
50
                                                                                       3. Display
51
                                                                                       Enter your choice:
       struct Stack* temp - ton:
```

```
Programiz C Online Compiler
                                                      לב כ כ Share
                                                                                        Output
main.c
                                                                             Run
48
           printf("Stack Underflow \n");
                                                                                     ▲ 1. Push
                                                                                      2. Pop
49
           return;
50
                                                                                      3. Display
                                                                                      Enter your choice: 1
51
       struct Stack* temp = top;
53
       printf("Pop value: %d\n", temp->data);
                                                                                      Enter value: 20
54
       top = top->next;
                                                                                      20 pushed
55
       free(temp);
56 }
                                                                                      1. Push
57 - void display() {
                                                                                      2. Pop
       struct Stack* temp = top;
                                                                                      3. Display
59
                                                                                      Enter your choice: 3
       if (top == NULL) {
60 -
                                                                                      Stack elements:
          printf("empty.\n");
                                                                                      10
62
           return;
63
                                                                                      1. Push
64
                                                                                      2. Pop
       printf("Stack elements:\n");
65
66 -
       while (temp != NULL) {
                                                                                      Display
         printf("%d\n", temp->data);
                                                                                      Enter your choice: 2
67
                                                                                      Pop value: 20
68
           temp = temp->next;
69
70 }
                                                                                      1. Push
71
                                                                                      2. Pop
72
                                                                                      Display
                                                                                      Enter your choice:
73
```

# queue through link list

```
Programiz C Online Compiler
                                                                      ς Share Run
                                                                                                                    Output
 1 // Online C compiler to run C program online
2 #include <stdio.h>
3 #include <stdlib.h>
                                                                                                                  1. insert
                                                                                                                  3. Display
Enter your choice: 1
 4 void insert();
                                                                                                                  Enter value: 20
 6 void display();
                                                                                                                  20 inset
 8- struct Queue {
9    int data;
10    struct Queue* next;
                                                                                                                 1. insert
2. delete
                                                                                                                  3. Display
                                                                                                                  Enter your choice: 3
12 struct Queue *front = NULL, *rear = NULL, *temp;
                                                                                                                  Queue elements:
13 - void main() {
15
               printf("\n1. insert");
                                                                                                                  1. insert
17
18
               printf("\n2. delete");

    delete
    Display

               printf("\n3. Display");
printf("\nEnter your choice: ");
19
20
                                                                                                                 Enter your choice: 2
Deleted value: 10
               scanf("%d", &choice);
21
22
23 -
                                                                                                                  1. insert
                    case 1: insert(); break;
case 2: del(); break;
                                                                                                                3. Display Enter your choice:
                    case 3: display(); break;
```

```
1. insert
                                                                                              2. delete
        } while (choice != 4);
28
29 }
                                                                                              3. Display
                                                                                              Enter your choice: 1
        void insert () {
31
        temp = (struct Queue*)malloc(sizeof(struct Queue));
                                                                                              Enter value: 20
32
                                                                                              20 inset
33 -
        if (temp == NULL) {
    printf("Memory allocation failed!\n");
34
                                                                                              1. insert
35
                                                                                              2. delete
36
37
                                                                                              3. Display
                                                                                              Enter your choice: 3
38 printf("\nEnter value: ");
                                                                                              Queue elements:
39
        scanf("%d", &temp->data);
                                                                                              10
40
        temp->next = NULL;
        if (rear == NULL) {
   front = rear = temp;
41 -
42
                                                                                             1. insert
2. delete
43 -
        } else {
            rear->next = temp;
                                                                                              3. Display
45
            rear = temp;
                                                                                              Enter your choice: 2
46
47
        printf("%d insert\n", temp->data);
                                                                                              Deleted value: 10
48 }
                                                                                              1. insert
49 - void del() {
50 ÷
        if (front == NULL) {
             printf("Queue Underflow\n");
                                                                                              3. Display
                                                                                             Enter your choice:
52
             return:
```

```
main.c
                                                           Share Run
                                                                                                Output
 51
                                                                                             △ 2. delete
             printf("Queue Underflow\n");
                                                                                              3. Display
Enter your choice: 1
 52
             return;
 53
 54
         temp = front;
         printf("Deleted value: %d\n", temp->data);
                                                                                              Enter value: 20
         front = front->next;
         if (front == NULL) {
   rear = NULL;
 57 -
58
                                                                                              1. insert
 59
                                                                                              2. delete
 60
         free(temp);
                                                                                               3. Display
                                                                                              Enter your choice: 3
 62 void display() {
                                                                                              Queue elements:
 63
         temp = front;
                                                                                              10
 64
                                                                                              20
         if (front == NULL) {
 65 -
            printf("Queue is empty.\n");
 66
                                                                                              1. insert
             return;
 68
                                                                                              3. Display
 69
                                                                                              Enter your choice: 2
         printf("Queue elements:\n");
while (temp != NULL) {
                                                                                              Deleted value: 10
 70
 71 -
            printf("%d\n", temp->data);
temp = temp->next;
 72
                                                                                              1. insert
 73
                                                                                              2. delete
 74
                                                                                              3. Display
 75 }
                                                                                              Enter your choice:
                                                                                               === Session Ended. Please Run the code again ===
```

#### Concatenation

#### Programiz C Online Compiler

```
Output
 main.c
 1 #include <stdio.h>
2 #include <stdlib.h>
3 * struct Node {
                                                                                          elements in List 1: 1
                                                                                         Enter value 1: 2
elements in List 2: 4
 4 int data;
5 struct Noo
 5 struct Node* next;
6 };
                                                                                         Enter value 1: 1
                                                                                         Enter value 2: 2
                                                                                         Enter value 3: 3
Enter value 4: 4
                                                                                         Concatenated List: 2 -> 1 -> 2 -> 3 -> 4 -> NULL
12 }
                                                                                         === Code Execution Successful ===
13 - void apNode(struct Node** head, int data) {
      struct Node* newNode = createNode(data);
14
15 +
       if (*head == NULL) {
        *head = newNode;
return;
16
17
      struct Node* temp = *head;
while (temp->next != NULL) {
   temp = temp->next;
}
18
19
20 -
21
22
23
       temp->next = newNode;
25 \cdot \text{void} concatenate(struct Node** head1, struct Node* head2) {
26 - if (*head1 == NULL) {
```

#### Programiz C Online Compiler

```
∝° Share
                                                                                 Run
main.c
27
            *head1 = head2;
28
            return;
29
        }
30
        struct Node* temp = *head1;
31 -
        while (temp->next != NULL) {
32
            temp = temp->next;
33
34
        temp->next = head2;
35 }
36 * void printList(struct Node* head) {
37 ⋅
        while (head != NULL) {
38
            printf("%d -> ", head->data);
39
            head = head->next;
40
        }
41
        printf("NULL\n");
42 }
43 - void inputList(struct Node** head, int n) {
44
        int val;
45 -
        for (int i = 0; i < n; i++) {
46
            printf("Enter value %d: ", i + 1);
47
           scanf("%d", &val);
48
            apNode(head, val);
49
        }
50 }
51 - int main() {
        struct Node* list1 = NULL:
```

```
48
            apNode(head, val);
       }
49
50 }
51 - int main() {
52
        struct Node* list1 = NULL;
53
        struct Node* list2 = NULL;
54
        int n1, n2;
        printf(" elements in List 1: ");
55
56
        scanf("%d", &n1);
        inputList(&list1, n1);
57
58
59
        printf(" elements in List 2: ");
60
        scanf("%d", &n2);
        inputList(&list2, n2);
61
62
63
        concatenate(&list1, list2);
64
        printf("\nConcatenated List: ");
65
66
        printList(list1);
67
68
        return 0;
69 }
70
```

#### tree traversal

```
Run
                                                                                Output
        main.c
        1 #include <stdio.h>
                                                                               number of nodes in the tree: 3
 R
        2 #include <stdlib.h>
                                                                               data part of the node: 60
                                                                               data part of the node: 55
        3 - struct tree {
             int data;
                                                                               data part of the node: 70
 5
              struct tree *lchild, *rchild;
        6 };
                                                                              Enter the traversal you want
 5
                                                                              1. Inorder
                                                                              2. Preorder
       8 struct tree *insert(struct tree *p, int n);
        9 void inorder(struct tree *p);
                                                                              3. Postorder
       10 void preorder(struct tree *p);
       11 void postorder(struct tree *p);
                                                                               inorder traversal is:
                                                                              55 60 70
       13 - int main() {
       14
              int x, y, i;
                                                                              === Code Execution Successful ===
       15
              struct tree *root = NULL;
       16
       17
              printf(" number of nodes in the tree: ");
       18
              scanf("%d", &x);
       19
              while (x-- > 0) {
 TS
       20 -
               printf(" data part of the node: ");
       21
       22
                  scanf("%d", &y);
                  root = insert(root, y);
       23
       24
php
       25
           printf("\nEnter the traversal you want\n");
       26
                                            [] ⟨ ⟨ cc Share
                                                                             Output
4
      main.c
                                                                            number of nodes in the tree: 3
      27
              printf("1. Inorder\n2. Preorder\n3. Postorder\n");
R
                                                                             data part of the node: 60
      28
             scanf("%d", &i);
                                                                             data part of the node: 55
      29
data part of the node: 70
            switch (i) {
      30 -
      31
               case 1:
                                                                            Enter the traversal you want
                   printf(" inorder traversal is:\n");
      32
                                                                            1. Inorder
      33
                     inorder(root);
                                                                            2. Preorder
      34
                    break;
                                                                            3. Postorder
      35
                 case 2:
                 printf("preorder traversal is:\n");
      36
0
                                                                            inorder traversal is:
      37
                     preorder(root);
                                                                            55 60 70
                    break;
(
      39
                case 3:
                                                                            === Code Execution Successful ===
```

printf(" postorder traversal is:\n");

struct tree \*newNode = (struct tree \*)malloc(sizeof(struct

postorder(root);

47 - struct tree \*insert(struct tree \*p, int n) {

tree));

newNode->data = n;

40

41

42

44 } 45

} 43

46 // insert node

48 \* if (p == NULL) {

(3)

JS

TS

php

```
∝ Share
                                                                                                   C
                                                                                       Run
          main.c
                            tree));
                                                                                                  nu
          50
                        newNode->data = n;
  R
                                                                                                  da
                        newNode->lchild = newNode->rchild = NULL;
          51
                                                                                                  da
                        return newNode;
          52
                                                                                                  da
          53
                   }
          54
                                                                                                 Ent
                   if (n < p->data)
          55
                                                                                                 1.
          56
                       p->lchild = insert(p->lchild, n);
                                                                                                 2.
          57
                   else
                                                                                                 3.
          58
                       p->rchild = insert(p->rchild, n);
                                                                                                 1
          59
  0
                                                                                                 in
          60
                   return p;
                                                                                                 55
          61 }
  (
          62
                                                                                                 ___
             // Inorder
          63
  64 - void inorder(struct tree *p) {
          65 -
                   if (p != NULL) {
                       inorder(p->lchild);
          66
  JS
          67
                       printf("%d ", p->data);
                       inorder(p->rchild);
          68
  TS
          69
                   }
          70 }
  -GO
          71 // Preorder
          72 - void preorder(struct tree *p) {
  php
          73 -
                   if (p != NULL) {
                       printf("%d ", p->data);
          74
                                                       Q Search
                                                                        LITTEL THE CLAVEL SOL YOU WOLL
9
                printf("%d ", p->data);
      67
                                                                       1. Inorder
                inorder(p->rchild);
      68
                                                                       2. Preorder
     69
                                                                       3. Postorder
      70 }
0
     71 // Preorder
                                                                        inorder traversal is:
      72 - void preorder(struct tree *p) {
                                                                       55 60 70
            if (p != NULL) {
     73 -
(
      74
                printf("%d ", p->data);
                                                                       === Code Execution Successful ===
                preorder(p->lchild);
      75
0
      76
                preorder(p->rchild);
     77
JS
     78 }
      79 // Postorder
      80 - void postorder(struct tree *p) {
TS
      81 -
            if (p != NULL) {
                postorder(p->lchild);
     82
GO
     83
                postorder(p->rchild);
                printf("%d ", p->data);
      84
     85
             }
     86 }
```

## tree searching (sorted)

