

Program No: 01	Date: 23/07/2025
----------------	------------------

Program Title : Write programs to demonstrate the use of storage classes (local variable, global variable, static variable, register variable) in C.

```
/*Program 1 USE DIFFERENT STORAGE CLASSES (LOCAL,GLOBAL,STATIC,REGISTER)
@ALBIN MAMMEN MATHEW
Roll No: 08
Date: 23/07/2025
*/
```

```
#include<stdio.h>
int a = 10;      //initialising global variable
void disp(){
    int i = 3;      //initialising local variable
    printf("this is a local variable : %d \n",i);
}
int main(){
    static int c;    //initialising static variable
    register int p = 4;    //initialising register variable
    printf("this is a static variable : %d \n",c);
    printf("this is a global variable : %d \n",a);
    printf("this is a register variable : %d \n",p);
    disp();
    return 0;
}
```

Output

```
this is a static variable : 0
this is a global variable : 10
this is a register variable : 4
this is a local variable : 3
```

Program No: 02

Date: 23/07/2025

Program Title : Use a menu-driven program to insert, search, delete and sort elements in an array using functions (use global variables).

```
/*PROGRAM-2 A MENU FOR ARRAY OPERATIONS(INSERT,DELETE,DISPLAY,SEARCH,SORT)  
USING GLOBAL VARIABLE
```

```
@ALBIN MAMMEN MATHEW
```

```
Roll No: 08
```

```
Date: 23/07/2025
```

```
*/
```

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

```
int stack[5]; //declaring stack
```

```
int top = -1; //declaring variable positionof top element
```

```
int insert(int e) { //Function to insert element into stack
```

```
    if (top + 1 == 5) {  
        printf("Error: Stack is Full");  
    }
```

```
    else {  
        stack[++top] = e;  
    }  
    return top;  
}
```

```
int erase() { //function to delete top element
```

```
    if (top == -1){  
        printf("Error: Stack is empty");  
    }
```

```
    else {  
        printf("\n %d",stack[top--]);  
    }  
    return top;  
}
```

```
void search(int b, int a[5], int top){ //function to search elements
```

```
    int isfound = 0, i;
```

```
    for (i = 0; i <= top; i++) {
```

```
        if (b == a[i]) {  
            isfound = 1;  
            printf("Element found at [%d] position. \n", i);  
        }
```

```
    }  
    if (isfound == 0)
```

```

        printf("element not found");
    }

void display() {      //function to display the elements in stack
    if (top == -1)
        printf("Empty Stack");
    else {
        int i;
        for (i = 0; i <= top; i++){
            printf("%d \t",stack[i]);
        }
        printf("\n");
    }
}

void sort(){ //function to sort the stack
    int i, j, temp;
    for (i = 0; i < 5; i++) {
        for (j = 0; j < 5; j++) {
            if(stack[i] < stack[j]) {
                temp = stack[i];
                stack[i] = stack[j];
                stack[j] = temp;
            }
        }
    }
}

int menu() { //function for menu
    int ch;
    printf("\n INSERT-1 \n DELETE-2 \n DISPLAY-3 \n SEARCH-4 \n SORT-5 \n EXIT -6 \n
Enter your choice: ");
    scanf("%d", &ch);
    return ch;
}

void processStack() { //working of menu
    int ch, b;
    for (ch = menu(); ch != 6; ch = menu()) {
        switch(ch){
            case 1:      //insert
                printf("Enter the value to insert: ");
                scanf("%d", &ch);
                insert(ch);
                break;
        }
    }
}

```

```

        case 2:      //delete
            erase();
            break;
        case 3:      //display
            display();
            break;
        case 4:      //search
            printf("Enter the value to search: ");
            scanf("%d",&b);
            search(b, stack, top);
            break;
        case 5:      // sort
            sort();
            break;
    default:     //any other options
            printf("Error: Wrong Choice");
            break;
    }
}
int main()
{
    processStack();
    return 0;
}

```

Output

```

INSERT-1
DELETE-2
DISPLAY-3
SEARCH-4
SORT-5
EXIT -6
Enter your choice : 1
Enter the value to insert : 10

INSERT-1
DELETE-2
DISPLAY-3
SEARCH-4
SORT-5
EXIT -6
Enter your choice : 1
Enter the value to insert : 20

```

```
INSERT-1
DELETE-2
DISPLAY-3
SEARCH-4
SORT-5
EXIT -6
Enter your choice : 1
Enter the value to insert : 30
```

```
INSERT-1
DELETE-2
DISPLAY-3
SEARCH-4
SORT-5
EXIT -6
Enter your choice : 3
10      20      30
```

```
INSERT-1
DELETE-2
DISPLAY-3
SEARCH-4
SORT-5
EXIT -6
Enter your choice : 2
```

```
30
INSERT-1
DELETE-2
DISPLAY-3
SEARCH-4
SORT-5
EXIT -6
Enter your choice : 3
10      20
```

```
INSERT-1
DELETE-2
DISPLAY-3
SEARCH-4
SORT-5
EXIT -6
Enter your choice : 1
Enter the value to insert : 40
```

```
INSERT-1
DELETE-2
DISPLAY-3
SEARCH-4
SORT-5
EXIT -6
Enter your choice : 1
Enter the value to insert : 10
```

```
INSERT-1
DELETE-2
DISPLAY-3
SEARCH-4
SORT-5
EXIT -6
Enter your choice : 1
Enter the value to insert : 50
```

```
INSERT-1
DELETE-2
DISPLAY-3
SEARCH-4
SORT-5
EXIT -6
Enter your choice : 3
10      20      40      10      50
```

```
INSERT-1
DELETE-2
DISPLAY-3
SEARCH-4
SORT-5
EXIT -6
Enter your choice : 4
Enter the value to search : 10
Element found at [0] position.
Element found at [3] position.
```

INSERT-1
DELETE-2
DISPLAY-3
SEARCH-4
SORT-5
EXIT -6

Enter your choice : 5

INSERT-1
DELETE-2
DISPLAY-3
SEARCH-4
SORT-5
EXIT -6

Enter your choice : 3

10 10 20 40 50

INSERT-1
DELETE-2
DISPLAY-3
SEARCH-4
SORT-5
EXIT -6

Enter your choice : 6

Program No: 03

Date: 23/07/2025

Program Title : Use a menu-driven program to insert, search, delete and sort elements in an array using functions (use only local variables).

```
/*PROGRAM-3 A MENU FOR ARRAY OPERATIONS(INSERT,DELETE,DISPLAY,SEARCH,SORT)
```

```
USING LOCAL VARIABLE
```

```
@ALBIN MAMMEN MATHEW
```

```
Roll No: 08
```

```
Date: 23/07/2025
```

```
*/
```

```
#include <stdio.h>
int insert(int a[5], int pos, int e) {      //function to insert an element
    if (pos + 1 == 5) {
        printf("Error: Array is Full\n");
    }
    else {
        a[++pos] = e;
    }
    return pos;
}

int erase(int a[5], int pos) {                //function to delete top element
    if (pos == -1) {
        printf("Error: Array is Empty\n");
    }
    else {
        printf("Deleted element: %d\n", a[pos--]);
    }
    return pos;
}

void display(int a[5], int pos){           //function to display entire array
    if (pos == -1) {
        printf("Error: Array is Empty\n");
    }
    else {
        int i;
        for (i = 0; i <= pos; i++) {
            printf("%d\t", a[i]);
        }
        printf("\n");
    }
}
```

```

    }

}

void search(int b, int a[5], int pos){ //function to search for an element and display its index
    int isfound = 0, i;
    for (i = 0; i <= pos; i++) {
        if(b==a[i]) {
            isfound=1;
            printf("Element found at [%d] position. \n",i);
        }
    }
    if (isfound == 0)
        printf("element not found");
}

void sort(int a[5]){ //function to sort the stack
    int i, j, temp;
    for (i = 0; i < 5; i++) {
        for (j = 0; j < 5; j++) {
            if(a[i] < a[j]) {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
}

int menu(){ //function to create menu interface
    int ch;
    printf("\nInsert - 1\nDelete - 2\nDisplay - 3\nSearch - 4\nSort - 5\nExit - 6\nEnter your choice: ");
    scanf("%d", &ch);
    return ch;
}

void processArray(){ //working of menu
    int a[5], pos = -1, b;
    int ch, value;
    for (ch = menu(); ch != 6; ch = menu()) {
        switch (ch) {
            case 1:
                printf("Enter value to insert: ");
                scanf("%d", &value);

```

```
        pos = insert(a, pos, value);
        break;
    case 2:
        pos = erase(a, pos);
        break;
    case 3:
        display(a, pos);
        break;
    case 4:
        printf("Enter the element to search: ");
        scanf("%d",&b);
        search(b,a,pos);
        break;
    case 5:
        sort(a);
        break;
    default:
        printf("Error: Wrong Choice.\n");
}
}

int main() {
    processArray();
    return 0;
}
```

Output

```
Insert - 1
Delete - 2
Display - 3
Search - 4
Sort - 5
Exit - 6
Enter your choice: 1
Enter value to insert: 10
```

```
Insert - 1
Delete - 2
Display - 3
Search - 4
Sort - 5
Exit - 6
Enter your choice: 1
Enter value to insert: 20
```

```
Insert - 1
Delete - 2
Display - 3
Search - 4
Sort - 5
Exit - 6
Enter your choice: 1
Enter value to insert: 10
```

```
Insert - 1
Delete - 2
Display - 3
Search - 4
Sort - 5
Exit - 6
Enter your choice: 3
10      20      10
```

```
Insert - 1
Delete - 2
Display - 3
Search - 4
Sort - 5
Exit - 6
Enter your choice: 2
Deleted element: 10
```



```
Insert - 1
Delete - 2
Display - 3
Search - 4
Sort - 5
Exit - 6
Enter your choice: 3
10      20
```

```
Insert - 1
Delete - 2
Display - 3
Search - 4
Sort - 5
Exit - 6
Enter your choice: 1
Enter value to insert: 20
```

```
Insert - 1
Delete - 2
Display - 3
Search - 4
Sort - 5
Exit - 6
Enter your choice: 1
Enter value to insert: 10
```

```
Insert - 1
Delete - 2
Display - 3
Search - 4
Sort - 5
Exit - 6
Enter your choice: 1
```



```
Enter value to insert: 10
```

```
Insert - 1
```

```
Delete - 2
```

```
Display - 3
```

```
Search - 4
```

```
Sort - 5
```

```
Exit - 6
```

```
Enter your choice: 3
```

```
10      20      20      10      10
```

```
Insert - 1
```

```
Delete - 2
```

```
Display - 3
```

```
Search - 4
```

```
Sort - 5
```

```
Exit - 6
```

```
Enter your choice: 4
```

```
Enter the element to search: 10
```

```
Element found at [0] position.
```

```
Element found at [3] position.
```

```
Element found at [4] position.
```

```
Insert - 1
```

```
Delete - 2
```

```
Display - 3
```

```
Search - 4
```

```
Sort - 5
```

```
Exit - 6
```

```
Enter your choice: 5
```

```
Insert - 1
```

```
Delete - 2
```

```
Display - 3
```

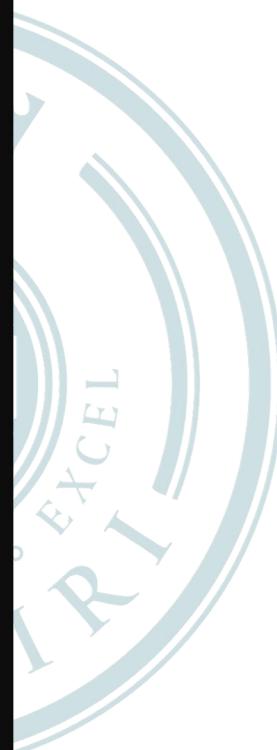
```
Search - 4
```

```
Sort - 5
```

```
Exit - 6
```

```
Enter your choice: 3
```

```
10      10      10      20      20
```



Insert - 1
Delete - 2
Display - 3
Search - 4
Sort - 5
Exit - 6
Enter your choice: 6



Program No: 04

Date: 23/07/2025

Program Title : Search for all the occurrences of an element in an integer array (positions).

```
/*PROGRAM-4 ARRAY SEARCH
@ALBIN MAMMEN MATHEW
Roll No: 08
Date: 23/07/2025
*/
```

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

```
void search(int b,int a[5]){ //function for search function
    int isfound = 0, i;
    for (i = 0; i < 5; i++){
        if (b == a[i]) {
            isfound = 1;
            printf("Element found at [%d] position. \n",i); //displays index
        }
    }
    if (isfound = 0)
        printf("element not found");
}
int main() {
    int a[5], ch, i;
    for (i = 0; i < 5; i++) {
        printf("Enter the [%d] element : ",i);
        scanf("%d",&a[i]);
    }
    for (i = 0; i < 5; i++)
        printf("%d \t",a[i]);
    printf("\n");
    printf("Enter the element to search: ");
    scanf("%d", &ch);
    search(ch,a);
    return 0;
}
```

Output

```
Enter the [0] element : 1
Enter the [1] element : 10
Enter the [2] element : 13
Enter the [3] element : 12
Enter the [4] element : 13
1      10      13      12      13
Enter the element to search: 13
Element found at [2] position.
Element found at [4] position.
```



Program No: 05	Date: 23/07/2025
----------------	------------------

Program Title : Sort the array elements in ascending order (minimum three functions: read, disp and sort).

```
/*PROGRAM-5 SORT ARRAY IN ASC WITH ALTEAST 3 FUNCTIONS - READ DISP SORT
@ALBIN MAMMEN MATHEW
Roll No: 08
Date: 23/07/2025
*/
#include<stdio.h>
void read(int n,int arr[n]){ //function to insert elements in array
    int i;
    for (i = 0; i < n; i++) {
        printf("Enter the value for %d : ",i);
        scanf("%d", &arr[i]);
    }
}

void disp(int n,int arr[n]) { //function to display all elemennts in array
    int i;
    for (i = 0; i < n; i++) {
        printf("%d\t", arr[i]);
    }
}

void sort(int n,int arr[n]) { //function for sorting the elements in array in ascending order
    int i, j, temp;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            if (arr[i] < arr[j]) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

int menu() { //funtion for menu interface
    int ch;
    printf("\nREAD-1\nSORT-2\nDISPLAY-3\nEXIT-4\nENTER YOUR CHOICE: ");
}
```

```
    scanf("%d", &ch);
    return ch;
}

void processArray(int n,int arr[n]) { //working of menu
    int ch;
    for (ch = menu(); ch != 4; ch = menu()) {
        switch(ch) {
            case 1:
                read(n,arr);
                break;
            case 2:
                sort(n,arr);
                break;
            case 3:
                disp(n,arr);
                break;
            default:
                printf("Error: Wrong Choice\n");
                break;
        }
    }
}

int main() {
    int n;
    printf("Enter limit of Array:");
    scanf("%d",&n);
    int arr[n];
    processArray(n,arr);
    return 0;
}
```

Output

```
Enter limit of Array:8
```

```
READ-1
```

```
SORT-2
```

```
DISPLAY-3
```

```
EXIT-4
```

```
ENTER YOUR CHOICE : 1
```

```
Enter the value for 0 :2
```

```
Enter the value for 1 :3
```

```
Enter the value for 2 :4
```

```
Enter the value for 3 :4
```

```
Enter the value for 4 :1
```

```
Enter the value for 5 :1
```

```
Enter the value for 6 :6
```

```
Enter the value for 7 :8
```

```
READ-1
```

```
SORT-2
```

```
DISPLAY-3
```

```
EXIT-4
```

```
ENTER YOUR CHOICE : 3
```

```
2      3      4      4      1      1      6      8
```

```
READ-1
```

```
SORT-2
```

```
DISPLAY-3
```

```
EXIT-4
```

```
ENTER YOUR CHOICE : 2
```

```
READ-1
```

```
SORT-2
```

```
DISPLAY-3
```

```
EXIT-4
```

```
ENTER YOUR CHOICE : 3
```

```
1      1      2      3      4      4      6      8
```

```
READ-1
```

```
SORT-2
```

```
DISPLAY-3
```

```
EXIT-4
```

```
ENTER YOUR CHOICE : 4
```

Program No: 06

Date: 23/07/2025

Program Title : Display the array elements in the same order using a recursive function.

```
/*PROGRAM-6 DISPLAY ARRAY USING RECURSIVE FUNCTION  
@ALBIN MAMMEN MATHEW  
Roll No: 08  
Date: 25/07/2025  
*/
```

```
#include<stdio.h>  
int j = 0, arr[10];  
void display() { //display function as a recursive function  
    if (j < 10) {  
        printf("%d\t", arr[j]);  
        j++;  
        display();  
    }  
    if (j == 10)  
        j = 0;  
}  
  
int main() {  
    int i=0;  
    for (i = 0; i < 10; i++) {  
        printf("Enter the value for %d:", i);  
        scanf("%d", &arr[i]);  
    }  
    display();  
    return 0;  
}
```

Output

```
Enter the value for 0 :1  
Enter the value for 1 :2  
Enter the value for 2 :3  
Enter the value for 3 :4  
Enter the value for 4 :5  
Enter the value for 5 :6  
Enter the value for 6 :7  
Enter the value for 7 :8  
Enter the value for 8 :9  
Enter the value for 9 :10  
1 2 3 4 5 6 7 8 9 10
```

Program No: 07

Date: 23/07/2025

Program Title : Display array elements in reverse order using a recursive function.

```
/*PROGRAM-7 DISPLAY ARRAY INN REVERSE USING RECURSIVE FUNCTION
```

```
@ALBIN MAMMEN MATHEW
```

```
Roll No: 08
```

```
Date: 26/07/2025
```

```
*/
```

```
#include<stdio.h>
int j = 10, arr[10];
void display() {           //function to display elements in reverse using recursive function
    if (j > 0) {
        j--;
        printf("%d\t",arr[j]);
        display();
    if (j == 0)
        j = 10;
    }
}

int main() {
    int i = 0;
    for (i = 0; i < 10; i++) {
        printf("Enter the value for %d:", i);
        scanf("%d", &arr[i]);
    }
    display();
    return 0;
}
```

Output

```
Enter the value for 0 :1
Enter the value for 1 :2
Enter the value for 2 :3
Enter the value for 3 :4
Enter the value for 4 :5
Enter the value for 5 :6
Enter the value for 6 :7
Enter the value for 7 :8
Enter the value for 8 :9
Enter the value for 9 :10
10      9      8      7      6      5      4      3      2      1
```

Program No: 08

Date: 23/07/2025

Program Title : Write a program to Perform the addition of two matrix and Subtraction of one matrix from another.

```
/*PROGRAM-8 MATRIX ADDITION AND SUBTRACTION
@ALBIN MAMMEN MATHEW
Roll No: 08
Date: 25/07/2025
*/
#include<stdio.h>
int a[10][10], b[10][10], m, n;

void insert(int e[10][10],int m, int n){ //function to insert values into the matrix
    int i, j;
    for (i = 0; i < m; i++){
        for (j = 0; j < n; j++){
            printf("Enter the value of [%d] [%d] \n: ", i, j);
            scanf("%d", &e[i][j]);
        }
    }
}

void print(int p[10][10],int m, int n){ //function to print a matrix
    int i, j;
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            printf("%d\t",p[i][j]);
        }
        printf("\n");
    }
}

void add() { //function to add two matrices
    int sum[10][10], i, j;
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            sum[i][j]=a[i][j]+b[i][j];
        }
    }
    print(sum,m,n);
}
```

```

void diffn() {           //function to subtract a matrix from another
    int dif[10][10], i, j;
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            dif[i][j]=a[i][j]-b[i][j];
        }
    }
    print(dif,m,n);
}

int main() {
    printf("Enter the dimensions of the Matrix: ");
    scanf("%d%d",&m,&n);

    printf("Enter the First matrix :\n");
    insert(a,m,n);
    printf("Enter the Second matrix :\n");
    insert(b,m,n);

    printf("The first matrix is : \n");
    print(a,m,n);
    printf("The Second matrix is : \n");
    print(b,m,n);

    printf("The sum of matrices is : \n");
    add(m,n);
    printf("The difference of matrices is : \n");
    diffn(m,n);
    return 0;
}

```

Output

```

Enter the dimensions of the Matrix: 2 3
Enter the First matrix :
Enter the value of [0] [0] : 1
Enter the value of [0] [1] : 2
Enter the value of [0] [2] : 3
Enter the value of [1] [0] : 4
Enter the value of [1] [1] : 5
Enter the value of [1] [2] : 6

```

```
Enter the Second matrix :  
Enter the value of [0] [0] : 0  
Enter the value of [0] [1] : 10  
Enter the value of [0] [2] : 0  
Enter the value of [1] [0] : 20  
Enter the value of [1] [1] : 0  
Enter the value of [1] [2] : 30  
The first matrix is :  
1      2      3  
4      5      6  
The Second matrix is :  
0      10     0  
20     0      30  
The sum of matrices is :  
1      12     3  
24     5      36  
The difference of matrices is :  
1      -8     3  
-16    5     -24
```



Program No: 09	Date: 23/07/2025
----------------	------------------

Program Title : Write a program to perform multiplication of two matrix.

```

/*PROGRAM-9 MATRIX MULTIPLICATION
@ALBIN MAMMEN MATHEW
Roll No: 08
Date: 25/07/2025
*/
#include<stdio.h>
int a[10][10],b[10][10], m, n, p;

void insert(int e[10][10], int m, int n) { //function to insert values in the matrix
    int i, j;
    for (i = 0; i < m; i++) {
        for(j=0;j<n;j++){
            printf("Enter the value of [%d] [%d]: ", i, j);
            scanf("%d",&e[i][j]);
        }
    }
}

void print(int p[10][10], int m, int n) { //funtion to display the matrix
    int i, j;
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            printf("%d\t",p[i][j]);
        }
        printf("\n");
    }
}

void multi() { //function to multiply two matrices
    int prod[10][10], i, j, k;
    for ( i = 0; i < m; i++) { //declaring intial value of elements of product to be zero
        for (j = 0; j < p; j++) {
            prod[i][j]=0;
        }
    }
    for (i = 0; i < m; i++) {
        for (j = 0; j < p; j++) {

```

```

        for (k = 0; k < n; k++) {
            prod[i][j] += a[i][k] * b[k][j];
        }
    }
    print(prod,m,p);
}

int main() {
    printf("Enter the dimensions of first Matrix: ");
    scanf("%d%d",&m,&n);
    printf("Enter the First matrix :\n");
    insert(a,m,n);
    printf("Enter the number of columns for the Second Matrix: ");
    scanf("%d", &p);
    printf("Enter the Second matrix :\n");
    insert(b,n,p);
    printf("The first matrix is : \n");
    print(a,m,n);
    printf("The Second matrix is : \n");
    print(b,n,p);
    printf("The product of matrices is : \n");
    multi();
    return 0;
}

```

Output

```

Enter the value of [0] [1] : 2
Enter the value of [0] [2] : 3
Enter the value of [1] [0] : 4
Enter the value of [1] [1] : 5
Enter the value of [1] [2] : 6
Enter the number of columns for the Second Matrix: 3
Enter the Second matrix :
Enter the value of [0] [0] : 2
Enter the value of [0] [1] : 4
Enter the value of [0] [2] : 6
Enter the value of [1] [0] : 8
Enter the value of [1] [1] : 10
Enter the value of [1] [2] : 12
Enter the value of [2] [0] : 14
Enter the value of [2] [1] : 16
Enter the value of [2] [2] : 18

```

```
The first matrix is :  
1      2      3  
4      5      6  
The Second matrix is :  
2      4      6  
8      10     12  
14     16     18  
The product of matrices is :  
60     72     84  
132    162    192
```



Program No: 10	Date: 23/07/2025
----------------	------------------

Program Title : Write a program to find the transpose of a matrix.

```

/*PROGRAM-10 MATRIX TRANSPOSE
@ALBIN MAMMEN MATHEW
Roll No: 08
Date: 25/07/2025
*/
#include<stdio.h>
int a[10][10], t[10][10];
void print(int e[10][10], int m, int n) {           //function to print the matrix.
    int i, j;
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            printf("%d\t", e[i][j]);
        }
        printf("\n");
    }
}
void transpose(int m, int n) {                      //function to find transpose of matrix
    int i,j;
    for (i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            t[j][i] = a[i][j];
        }
    }
    printf("The Transpose of the matrix is :\n");
    print(t,n,m);
}

int main() {
    int m,n;
    printf("Enter the dimensions of the Matrix:");
    scanf("%d%d", &m, &n);
    int i, j;
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            printf("Enter the value of [%d] [%d]: ", i, j);
            scanf("%d",&a[i][j]);
        }
    }
}

```

```
    }
    printf("The matrix is :\n");
    print(a,m,n);
    transpose(m,n);
    return 0;
}
```

Output

```
Enter the dimensions of the Matrix :2 4
Enter the value of [0] [0] : 1
Enter the value of [0] [1] : 2
Enter the value of [0] [2] : 3
Enter the value of [0] [3] : 4
Enter the value of [1] [0] : 5
Enter the value of [1] [1] : 6
Enter the value of [1] [2] : 7
Enter the value of [1] [3] : 8
The matrix is :
1      2      3      4
5      6      7      8
The Transpose of the matrix is :
1      5
2      6
3      7
4      8
```

Program No: 11

Date: 23/07/2025

Program Title : Write a program to find the Determinant of a matrix (2x2 and 3x3).

```
/* PROGRAM-11 DETERMINANT OF A MATRIX
@ALBIN MAMMEN MATHEW
Roll No: 08
Date: 26/07/2025
*/
#include <stdio.h>

int main() {
    int size, a[3][3], i, j;
    float det;
    printf("Enter the size of the square matrix (2 or 3): ");
    scanf("%d", &size);

    if (size != 2 && size != 3) {
        printf("Only 2x2 and 3x3 matrices are supported.\n");
        return 1;
    }

    printf("Enter the elements of the matrix:\n");
    for (i = 0; i < size; i++) {
        for (j = 0; j < size; j++) {
            printf("a[%d][%d]: ", i, j);
            scanf("%d", &a[i][j]);
        }
    }

    printf("The matrix is:\n");
    for (i = 0; i < size; i++) {
        for (j = 0; j < size; j++) {
            printf("%d\t", a[i][j]);
        }
        printf("\n");
    }

    if (size == 2) {
        det=a[0][0] * a[1][1] - a[0][1] * a[1][0]; // For 2x2: |A| = ad - bc
    } else if (size==3) {
        det = a[0][0] * (a[1][1] * a[2][2] - a[1][2] * a[2][1]) //expansion of formula
        - a[0][1] * (a[1][0] * a[2][2] - a[1][2] * a[2][0])
    }
}
```

```
        + a[0][2] * (a[1][0] * a[2][1] - a[1][1] * a[2][0]);
    }

    printf("Determinant of the matrix = %.2f\n", det);
    return 0;
}
```

Output

```
Enter the size of the square matrix (2 or 3): 3
Enter the elements of the matrix:
a[0][0]: 2
a[0][1]: -3
a[0][2]: 1
a[1][0]: 2
a[1][1]: 0
a[1][2]: -1
a[2][0]: 1
a[2][1]: 4
a[2][2]: 5
The matrix is:
2      -3      1
2       0      -1
1       4       5
Determinant of the matrix = 49.00
```

```
Enter the size of the square matrix (2 or 3): 2
Enter the elements of the matrix:
a[0][0]: 1
a[0][1]: 2
a[1][0]: 3
a[1][1]: 4
The matrix is:
1      2
3      4
Determinant of the matrix = -2.00
```

Program No: 12

Date: 23/07/2025

Program Title : Implement stack operations using arrays.

```
/*PROGRAM-12 STACK OPERATIONS USING ARRAY
```

```
@ALBIN MAMMEN MATHEW
```

```
Roll No: 08
```

```
Date: 26/07/2025
```

```
*/
```

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

```
int push(int stack[5], int top, int e) { //function to push elements onto stack
```

```
    if (top + 1 == 5) {
```

```
        printf("Error: Stack is Full\n");
```

```
    } else {
```

```
        stack[++top] = e;
```

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

```
}
```

```
    return top;
```

```
}
```

```
int pop(int stack[5], int top){ //function to pop the top element from stack
```

```
    if (top == -1){
```

```
        printf("Error: Stack is Empty\n");
```

```
    } else {
```

```
        printf("Popped element: %d\n", stack[top--]);
```

```
}
```

```
    return top;
```

```
}
```

```
void peek(int stack[5], int top) { //function to peek the top element in stack
```

```
    if (top == -1) {
```

```
        printf("Stack is Empty\n");
```

```
}else{
```

```
        printf("Top element: %d \n",stack[top]);
```

```
}
```

```
}
```

```
int menu() { //function for menu interface
```

```
    int ch;
```

```
    printf("\nPush - 1\nPop - 2\nPeek - 3\nExit - 4\nEnter your choice: ");
```

```
    scanf("%d", &ch);
```

```
    return ch;
```

```

}

void processStack() {      //working of menu
    int stack[5], top = -1;
    int ch, value;
    for (ch = menu(); ch != 4; ch = menu()) {
        switch (ch) {
            case 1:
                printf("Enter value to insert: ");
                scanf("%d", &value);
                top = push(stack, top, value);
                break;
            case 2:
                top = pop(stack, top);
                break;
            case 3:
                peek(stack, top);
                break;
            default:
                printf("Error: Wrong Choice.\n");
        }
    }
}

int main()
{
    processStack();
    return 0;
}

```

Output

```

Push - 1
Pop - 2
Peek - 3
Exit - 4
Enter your choice: 1
Enter value to insert: 10
Pushed

```

```
Push - 1
Pop - 2
Peek - 3
Exit - 4
Enter your choice: 1
Enter value to insert: 20
Pushed

Push - 1
Pop - 2
Peek - 3
Exit - 4
Enter your choice: 1
Enter value to insert: 3
Pushed

Push - 1
Pop - 2
Peek - 3
Exit - 4
Enter your choice: 3
Top element: 3

Push - 1
Pop - 2
Peek - 3
Exit - 4
Enter your choice: 2
Popped element: 3

Push - 1
Pop - 2
Peek - 3
Exit - 4
Enter your choice: 3
Top element: 20

Push - 1
Pop - 2
Peek - 3
Exit - 4
Enter your choice: 4
```



Program No: 13	Date: 23/07/2025
----------------	------------------

Program Title : Read a String and Just print it in the reverse order.

```
/* PROGRAM-13 STRING REVERSAL
@ALBIN MAMMEN MATHEW
Roll No: 08
Date: 27/07/2025
*/
```

```
#include<stdio.h>

void str_rev(char a[20]){      //program to reverse a string
    int i=0;
    while (a[i] != '\0') {
        i++;
    }
    for ( ; i >= 0; i--) {
        printf("%c", a[i]);
    }
}
int main() {
    char a[20];
    printf("Enter a String: ");
    gets(a);
    str_rev(a);
    return 0;
}
```

Output

```
Enter a String: Hello World
dlrow olleH
```

Program No: 14	Date: 23/07/2025
----------------	------------------

Program Title : Read a String and Reverse the string in the same array itself.

```
/* PROGRAM-14 STRING REVERSAL IN THE SAME ARRAY
@ALBIN MAMMEN MATHEW
Roll No: 08
Date: 27/07/2025
*/
```

```
#include <stdio.h>
#include <string.h>
void reverse(char str[20]) { //function to reverse the string
    int i = 0, j;
    char temp;
    j = strlen(str) - 1;
    while (i < j) { //swaping end characters
        temp = str[i];
        str[i] = str[j];
        str[j] = temp;
        i++;
        j--;
    }
}

int main() {
    char str[20];
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    printf("String is %s", str);
    reverse(str);
    printf("Reversed string : %s\n", str);
    return 0;
}
```

Output

```
Enter a string: HELLO world!
String is HELLO world!
Reversed string :
!dlrow OLLEH
```

Program No: 15

Date: 23/07/2025

Program Title : Read n Strings and display them in the ascending order.

```
/* PROGRAM-15 SORTING N STRINGS
@ALBIN MAMMEN MATHEW
Roll No: 08
Date: 28/07/2025
*/
#include <stdio.h>
#include <string.h>
int main() {
    int n, i, j;
    char str[20][100], temp[100];

    printf("Enter the number of strings: ");
    scanf("%d", &n);
    getchar(); // to consume the newline after scanf

    for (i = 0; i < n; i++) {
        printf("Enter string %d: ", i + 1);
        fgets(str[i], sizeof(str[i]), stdin);
        str[i][strcspn(str[i], "\n")] = '\0'; // remove newline
    }

    for (i = 0; i < n - 1; i++) // Sort strings using bubble sort
        for (j = i + 1; j < n; j++) {
            if (strcmp(str[i], str[j]) > 0) {
                strcpy(temp, str[i]);
                strcpy(str[i], str[j]);
                strcpy(str[j], temp);
            }
        }

    printf("\nStrings in ascending order:\n");
    for (i = 0; i < n; i++) {
        printf("%s\n", str[i]);
    }

    return 0;
}
```

Output

```
Enter the number of strings: 6
Enter string 1: Albin
Enter string 2: Shyam
Enter string 3: Febin
Enter string 4: Anandhu
Enter string 5: Tom
Enter string 6: Sharvin
```

```
Strings in ascending order:
```

```
Albin
Anandhu
Febin
Sharvin
Shyam
Tom
```



Program No: 16

Date: 23/07/2025

Program Title : Reverse a string using Stack.

```
/* PROGRAM-16 STRING REVERSAL USING STACK
@ALBIN MAMMEN MATHEW
Roll No: 08
Date: 28/07/2025
*/
```

```
#include <stdio.h>
char stack[100];
int top=-1;
void push(char e) { //function to push elements onto stack
    if (top + 1 == 100) {
        printf("Error: Stack is Full\n");
    }
    else {
        stack[++top] = e;
        printf("%c",e);
    }
}

void pop(){ //function to top pop element from stack
    if (top == -1) {
        printf("Error: Stack is Empty\n");
    }
    else {
        printf("%c", stack[top--]);
    }
}

void peek() { //function to display the top element of the stack
    int i;
    if (top == -1) {
        printf("Stack is Empty\n");
    }
    else {
        printf("%c \n", stack[top]);
    }
}
```

```

int menu() {
    int ch;
    printf("\nPush - 1\nPop - 2\nPeek - 3\nReverse - 4\nExit - 5\nEnter your choice: ");
    scanf("%d", &ch);
    return ch;
}

void processArray() {
    int ch;
    char value;
    for (ch = menu(); ch != 5; ch = menu()) {
        switch (ch) {
            case 1:
                printf("Enter value to insert: ");
                scanf(" %c", &value);
                push(value);
                break;
            case 2:
                pop();
                break;
            case 3:
                peek();
                break;
            case 4:
                while (top != -1) { //function to pop all elements
                    pop();
                }
                break;
            default:
                printf("Error: Wrong Choice.\n");
        }
    }
}

int main() {
    processArray();
    return 0;
}

```

Output

```
Push - 1
Pop - 2
Peek - 3
Reverse - 4
Exit - 5
Enter your choice: 1
Enter value to insert: H
H
Push - 1
Pop - 2
Peek - 3
Reverse - 4
Exit - 5
Enter your choice: 1
Enter value to insert: e
e
Push - 1
Pop - 2
Peek - 3
Reverse - 4
Exit - 5
Enter your choice: 1
Enter value to insert: l
l
Push - 1
Pop - 2
Peek - 3
Reverse - 4
Exit - 5
Enter your choice: 1
Enter value to insert: l
l
```



```
Push - 1
Pop - 2
Peek - 3
Reverse - 4
Exit - 5
Enter your choice: 1
Enter value to insert: o
o
Push - 1
Pop - 2
Peek - 3
Reverse - 4
Exit - 5
Enter your choice: 1
Enter value to insert: !
!
Push - 1
Pop - 2
Peek - 3
Reverse - 4
Exit - 5
Enter your choice: 4
olleH
Push - 1
Pop - 2
Peek - 3
Reverse - 4
Exit - 5
Enter your choice: 5
```

