EXPERIMENT NO. : 5

Array

1). WAP to read a list of integers and store it in a single dimensional array. Write a C program to print the second largest integer in a list of integers.

```
#include <stdio.h>
int main ()
{
int n, i ;
int arr [100];
int largest , second_largest ;

printf ("Enter the number of elements : ");
scanf ("%d", &n);

printf ("Enter %d integers : \n", n);
for (i = 0 ; i < n ; i ++) {
  scanf ("%d", & arr [i]) ;
}
if (arr [0] > arr [1])
{
largest = arr [0];
second_largest = arr [1];
} else
 {
```

Date _____

Expt. No. _____ Page No. _____

```
    largest = arr[1];
    second_largest = arr[0];
}
  for (i = 2; i < n; i++)
{
  if (arr[i] > largest)
{
    second_largest = largest;
    largest = arr[i];
}
else if (arr[i] > second_largest && arr[i] != largest)
{
second_largest = arr[i];
}
}
printf("The second largest number is : %d\n", second_
                                        largest);
return 0;
}
```

Output :- Enter the number of elements : 6
Enter 6 integers :
20 3 78 83 100 105
The second largest number is : 100

Teacher's Signature _____

main.c

Share Run

```c
#include <stdio.h>

int main() {
    int n, i;
    int arr[100];
    int largest, secondLargest;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d integers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    if (arr[0] > arr[1]) {
        largest = arr[0];
        secondLargest = arr[1];
    } else {
        largest = arr[1];
        secondLargest = arr[0];
    }
    for(i = 2; i < n; i++) {
        if (arr[i] > largest) {
            secondLargest = largest;
            largest = arr[i];
        } else if (arr[i] > secondLargest && arr[i] != largest) {
            secondLargest = arr[i];
        }
    }
    printf("\nSecond largest number = %d\n", secondLargest);
    return 0;
}
```

Output

Clear

```
Enter number of elements: 6
Enter 6 integers:
20 3 78 83 100 105

Second largest number = 100


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

Search

ENG
IN

23:19
13-10-2025

Date

Expt. No. Page No.

2) WAP to read a list of integers and store it in a single dimensional array. Write a C program to count and display positive, negative, odd, and even numbers, in an array.

```
#include <stdio.h>
int main ()
{
int n, i ;
int arr [100];
int positive = 0 , negative = 0 ; odd = 0 , even = 0 ;

printf ("Enter the number of elements : ") ;
scanf ("%d" , &n);

printf ("Enter %d integers : \n" , n);
for (i = 0 ; i < n ; i ++)
{
 scanf (" %d " , &arr [i]);
}
for (i = 0 ; i < n ; i ++ ){
 if (arr [i] >= 0)
    positive ++ ;
 else
    negative ++ ;
 if (arr [i] % 2 == 0)
   even ++ ;
 else
   odd ++ ;
```

Teacher's Signature

```
}
printf ("\n Results : \n");
printf ("Positive numbers : %d \n", positive);
printf ("Negative numbers : %d \n", negative);
printf ("Even numbers : %d \n", even);
printf ("Odd numbers : %d \n", odd);

return 0;
}
```

Output :- Enter the number of elements : 8
Enter 8 Integers :
3 -20 83 -78 19 0 -14 -28

Results :
Positive numbers : 4
Negative numbers : 4
Even numbers : 5
Odd numbers : 3

main.c

Share Run

```c
#include <stdio.h>

int main() {
    int n, i;
    int arr[100];
    int positive = 0, negative = 0, even = 0, odd = 0;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d integers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    for(i = 0; i < n; i++) {
        if(arr[i] > 0)
            positive++;
        else if(arr[i] < 0)
            negative++;

        if(arr[i] % 2 == 0)
            even++;
        else
            odd++;
    }
    printf("\nCount of Positive numbers: %d", positive);
    printf("\nCount of Negative numbers: %d", negative);
    printf("\nCount of Even numbers: %d", even);
    printf("\nCount of Odd numbers: %d\n", odd);

    return 0;
}
```

Output

Clear

```
Enter number of elements: 8
Enter 8 integers:
3 -20 83 -78 19 0 -14 -28

Count of Positive numbers: 3
Count of Negative numbers: 4
Count of Even numbers: 5
Count of Odd numbers: 3


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

Search

ENG
IN

23:22
13-10-2025

Date

Expt. No. Page No.

3) WAP to read a list of integers and store it in a single dimensional array. Write program to find the frequency of a particular number in a list of integers.

```
#include <stdio.h>
int main ( )
{
int n, i, key, count = 0;
int arr [100];

printf ("Enter the number of elements : ");
scanf ("%d", &n);

printf ("Enter %d integers : \n", n);
for (i = 0 ; i < n ; i++)
{
scanf ("%d", &arr [i]);
}
printf ("Enter the number to find its frequency: ");
scanf ("%d", &key);

for (i = 0 ; i < n ; i++)
{
if (arr [i] == key) {
    count++;
  }
}
```

Teacher's Signature

Expt. No. _____ Date _____

Page No. _____

```
printf("\n The frequency of %d is: %d\n", key,
                                        count);
return 0;
}
```

Output :- Enter the number of elements : 8
Enter 8 integers :
20 3 19 3 19 78 20 19
Enter the number to find its frequency : 19

The frequency of 19 is : 3

Teacher's Signature _____

main.c

Share Run

```c
#include <stdio.h>

int main() {
    int n, i, num, count = 0;
    int arr[100];

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d integers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Enter the number to find its frequency: ");
    scanf("%d", &num);

    for(i = 0; i < n; i++) {
        if(arr[i] == num)
            count++;
    }

    printf("\nFrequency of %d = %d\n", num, count);

    return 0;
}
```

Output

Clear

```
Enter number of elements: 8
Enter 8 integers:
20 3 19 3 19 78 20 19
Enter the number to find its frequency: 19

Frequency of 19 = 3


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

Search

ENG
IN

23:27
13-10-2025

Expt. No. _____ Date _____

Page No. _____

4) WAP that reads two matrices A(m x n) and B(p x q) and computes the product A and B. Read matrix A and matrix B in row major order respectively. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of uncompatibility

```
#include <stdio.h>
int main ( )
{
int m, n, p, q, A[20][20], B[20][20],
                C[20][20], i, j, k;
printf ("Enter rows & cols of A : ");
scanf (" %d  %d", &m, &n);

printf (" Enter rows & cols of B : ");
scanf (" %d

if (n != p)
{
printf ("\n Multiplication not possible ! (cols of
                A != Rows of B) \n");

return 0;
}
printf ("Enter A ( %d x %d ) : \n", m, n);
for (i = 0; i < m; i++)
```

Teacher's Signature _____

Date _____

Expt. No. _____ Page No. _____

```
for (j=0 ; j<q ; j++)
scanf ("%d", &B[i][j]);

for(i=0 ; i<m ; i++)
for (j=0 ; j<q ; j++)
{
  C[i][j] = 0;
for [k=0 ; k<n ; k++)
  C[i][j] += A[i][j] * B[k][j];
}
printf ("\n Matrix A: \n");

for(i = 0 ; i<m ; i++)
{
for (j = 0 ; j<n ; j++)

printf ("%4d", A[i][j]);
printf ("\n");
 }
printf ("\n Matrix B: \n");

for (i = 0 ; i<p ; i++)
{
for (j = 0 ; j<q ; j++)

printf ("%4d", B[i][j]);
printf ("\n");
}
printf ("\n Resultant Matrix (A x B): \n");
```

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

```
for (i = 0 ; i < m ; i++)
{
for (j = 0 ; j < q ; j++)

printf ("%4d", C[i][j]);
printf ("\n");
}

return 0;
}
```

Output :-

```
Enter rows & cols of A : 2 3
Enter rows & cols of B : 3 2

Enter A (2 x 3) :
 1   2   3
 4   5   6

Enter B (3 x 2) :
 7   8
 9   10
 11  12

Matrix A :
 1   2   3
 4   5   6
```

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

Matrix B :

7	8
9	10
11	12

Resultant Matrix (A x B) :

56	64
139	154

Teacher's Signature _____

main.c

Share Run

```c
#include <stdio.h>

int main() {
    int A[20][20], B[20][20], C[20][20];
    int m, n, p, q;
    int i, j, k;

    printf("Enter the order of Matrix A (rows columns): ");
    scanf("%d %d", &m, &n);
    printf("Enter the order of Matrix B (rows columns): ");
    scanf("%d %d", &p, &q);

    if (n != p) {
        printf("\nMatrix multiplication not possible!\n");
        printf("Reason: Number of columns of A (%d) is not equal to number of rows
            of B (%d).\n", n, p);
        return 0;
    }
    printf("\nEnter elements of Matrix A (%d x %d) in row-major order:\n", m, n);
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            scanf("%d", &A[i][j]);
        }
    }
    printf("\nEnter elements of Matrix B (%d x %d) in row-major order:\n", p, q);
    for(i = 0; i < p; i++) {
        for(j = 0; j < q; j++) {
            scanf("%d", &B[i][j]);
        }
    }
```

Output

Clear

```
Enter the order of Matrix A (rows columns): 2 3
Enter the order of Matrix B (rows columns): 3 2

Enter elements of Matrix A (2 x 3) in row-major order:
1 2 3
4 5 6

Enter elements of Matrix B (3 x 2) in row-major order:
7 8
9 10
11 12

Matrix A (2 x 3):
   1   2   3
   4   5   6

Matrix B (3 x 2):
   7   8
   9  10
  11  12

Product Matrix (A x B) = C (2 x 2):
  58  64
 139 154


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

Search
ENG IN
23:56
13-10-2025

main.c

Share Run

```c
    for(i = 0; i < m; i++) {
        for(j = 0; j < q; j++) {
            C[i][j] = 0;
        }
    }
    for(i = 0; i < m; i++) {
        for(j = 0; j < q; j++) {
            for(k = 0; k < n; k++) {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }
    printf("\nMatrix A (%d x %d):\n", m, n);
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("%4d", A[i][j]);
        }
        printf("\n");
    }
    printf("\nMatrix B (%d x %d):\n", p, q);
    for(i = 0; i < p; i++) {
        for(j = 0; j < q; j++) {
            printf("%4d", B[i][j]);
        }
        printf("\n");
    }
    printf("\nProduct Matrix (A x B) = C (%d x %d):\n", m, q);
    for(i = 0; i < m; i++) {
        for(j = 0; j < q; j++) {
            printf("%4d", C[i][j]);
```

Output

Clear

```
Enter the order of Matrix A (rows columns): 2 3
Enter the order of Matrix B (rows columns): 3 2

Enter elements of Matrix A (2 x 3) in row-major order:
1 2 3
4 5 6

Enter elements of Matrix B (3 x 2) in row-major order:
7 8
9 10
11 12

Matrix A (2 x 3):
   1   2   3
   4   5   6

Matrix B (3 x 2):
   7   8
   9  10
  11  12

Product Matrix (A x B) = C (2 x 2):
  58  64
 139 154


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

Search
ENG IN
23:57
13-10-2025

main.c

Share Run

```c
    for(i = 0; i < m; i++) {
        for(j = 0; j < q; j++) {
            for(k = 0; k < n; k++) {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }
    printf("\nMatrix A (%d x %d):\n", m, n);
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("%4d", A[i][j]);
        }
        printf("\n");
    }
    printf("\nMatrix B (%d x %d):\n", p, q);
    for(i = 0; i < p; i++) {
        for(j = 0; j < q; j++) {
            printf("%4d", B[i][j]);
        }
        printf("\n");
    }
    printf("\nProduct Matrix (A x B) = C (%d x %d):\n", m, q);
    for(i = 0; i < m; i++) {
        for(j = 0; j < q; j++) {
            printf("%4d", C[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Output

Clear

```
Enter the order of Matrix A (rows columns): 2 3
Enter the order of Matrix B (rows columns): 3 2

Enter elements of Matrix A (2 x 3) in row-major order:
1 2 3
4 5 6

Enter elements of Matrix B (3 x 2) in row-major order:
7 8
9 10
11 12

Matrix A (2 x 3):
   1   2   3
   4   5   6

Matrix B (3 x 2):
   7   8
   9  10
  11  12

Product Matrix (A x B) = C (2 x 2):
  58  64
 139 154


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

Search
ENG IN
23:57
13-10-2025