EXPERIMENT-5

QUESTION-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.

CODE:

```
#include <stdio.h>
int main()
{
int n, i, largest, secondLargest;

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

if(n < 2)
{
printf("At least two numbers are required.\n");
return 0;
}
int arr[n];
printf("Enter %d integers:\n", n);
for(i = 0; i < n; i++)
scanf("%d", &arr[i]);

largest = secondLargest = arr[0];

for(i = 1; i < n; i++)
{
if(arr[i] > largest)
{
secondLargest = largest;
largest = arr[i];
}
else if(arr[i] > secondLargest && arr[i] != largest)
{
```

```
secondLargest = arr[i];
}
}
int count = 0;
for(i = 0; i < n; i++)
{
if(arr[i] == largest)
count++;
}
if(count >= 2)
printf("No second largest element (elements equal).\n");
else
printf("Second Largest Element = %d\n", secondLargest);

return 0;
}
```

OUTPUTS:

(a) Enter number of elements: 1

At least two numbers are required.

(b) Enter number of elements: 4

Enter 4 integers:

1

2

1

2

No second largest element (elements equal).

(c) Enter number of elements: 5

Enter 5 integers:

1

2

3

4

5

Second Largest Element = 4

QUESTION-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.

CODE:

```
#include <stdio.h>
int main()
{
int n, i;
int pos = 0, neg = 0, even = 0, odd = 0,zero=0;

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

int arr[n];
printf("Enter %d integers:\n", n);

for(i = 0; i < n; i++)
{
scanf("%d", &arr[i]);

if(arr[i] > 0)
```

```
pos++;
else if(arr[i]< 0)
neg++;
else
zero++;

if(arr[i]!=0)
{
if(arr[i] % 2 == 0)
even++;
else
odd++;
}
}
printf("Positive numbers = %d\n", pos);
printf("Negative numbers = %d\n", neg);
printf("Even numbers = %d\n", even);
printf("Odd numbers = %d\n", odd);
return 0;
}
```

OUTPUT:

(a) Enter number of elements: 5

Enter 5 integers:

0

0

0

0

0

Positive numbers = 0

Negative numbers = 0

Even numbers = 0

Odd numbers = 0

(b) Enter number of elements: 4

Enter 4 integers:

-1

2

0

3

Positive numbers = 2

Negative numbers = 1

Even numbers = 1

Odd numbers = 2

(c) Enter number of elements: 4

Enter 4 integers:

1

c

Positive numbers = 1

Negative numbers = 1

Even numbers = 0

Odd numbers = 2

NOTE:the above error can be fixed as:

```
if(scanf("%d", &arr[i]) != 1)
{
    printf("Invalid input! Please enter an integer.\n");
    return 0;
}
```

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

CODE:

```c
#include <stdio.h>

int main()
{
int n, i, num, count = 0;

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

int arr[n];
printf("Enter %d integers:\n", n);

for(i = 0; i < n; i++)
if(scanf("%d", &arr[i])!=1)
return 0;

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

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

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

return 0;
}
```

OUTPUT:

(a) Enter number of elements: 5

Enter 5 integers:

2

3

4

2

3

Enter number to find frequency of: 2

Frequency of 2 = 2

(b) Enter number of elements: 2

Enter 2 integers:

3

3

Enter number to find frequency of: 4

Frequency of 4 = 0

(c) Enter number of elements: 0

Enter 0 integers:

Enter number to find frequency of: 2

Frequency of 2 = 0

NOTE: above case can be fixed as

```
if(n<=1)
{
   printf(“minimum 1 number should be entered”);
}
```

QUESTION4: WAP that reads two matrices A(mxn) and B(pxq) 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 incompatibility.

CODE:

```c
#include <stdio.h>

int main()
{
int m, n, p, q;
int i, j, k;

printf("Enter order of matrix A (m n): ");
if(scanf("%d %d", &m, &n)!=2)
{
printf("invalid input");
return 0;
}

printf("Enter order of matrix B (p q): ");
if(scanf("%d %d", &p, &q)!=2)
{
printf("invalid input");
return 0;
}
```

```
if(n != p)
{
printf("Matrix multiplication not possible. Columns of A must equal rows of B.\n");
return 0;
}

int A[m][n], B[p][q], C[m][q];

printf("\nEnter elements of Matrix A:\n");
for(i = 0; i < m; i++)
for(j = 0; j < n; j++)
if(scanf("%d", &A[i][j])!=1)
{
printf("invalid input");
return 0;
}

printf("\nEnter elements of Matrix B:\n");
for(i = 0; i < p; i++)
for(j = 0; j < q; j++)
if(scanf("%d", &B[i][j])!=1)
{
printf("invalid input");
return 0;
}
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][k] * B[k][j];
}

printf("\nMatrix A:\n");
for(i = 0; i < m; i++)
{
for(j = 0; j < n; j++)
printf("%5d", A[i][j]);
printf("\n");
```

```
}

printf("\nMatrix B:\n");
for(i = 0; i < p; i++)
{
for(j = 0; j < q; j++)
printf("%5d", B[i][j]);
printf("\n");
}

printf("\nResultant Matrix (AxB):\n");
for(i = 0; i < m; i++)
{
for(j = 0; j < q; j++)
printf("%5d", C[i][j]);
printf("\n");
}

return 0;
}
```

OUTPUT:

(a) Enter order of matrix A (m n): 1

1

Enter order of matrix B (p q): 1

1

Enter elements of Matrix A:

2

Enter elements of Matrix B:

2

Matrix A:

2

Matrix B:

2

Resultant Matrix (AxB):

4

(b) Enter order of matrix A (m n): 2

3

Enter order of matrix B (p q): 4

5

Matrix multiplication not possible. Columns of A must equal rows of B.

(c) Enter order of matrix A (m n): 2 2

Enter order of matrix B (p q): 2 2

Enter elements of Matrix A:

1

2

3

4

Enter elements of Matrix B:

5

6

7

8

Matrix A:

1 2

3 4

Matrix B:

5 6

7 8

Resultant Matrix (AxB):

19 22

43 50