

Name Priyadarshi Prabhakar SAP ID 590029237

EXPERIMENT 5 : ARRAY

Activity 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.*

ALGORITHM:

STEP 1: START

STEP 2: Read the no. of elements (n).

STEP 3: If $n < 2$ then print "No second largest element" and go to **STEP8**
else go to **STEP 4**

STEP 4: Read all elements and store them into an array **a** of size **n**

STEP 5: Initialize largest and second using the first two array elements:

if $a[0] > a[1]$ then largest = $a[0]$ & second = $a[1]$ and go to **STEP6**
else largest = $a[1]$ & second = $a[0]$ and go to **STEP6**

STEP 6: Repeat for $i = 2$ to $i = n-1$:

if $a[i] > \text{largest}$ then set second = largest & largest = $a[i]$
elseif $a[i] > \text{second}$ **AND** $a[i] \neq \text{largest}$ then set second = $a[i]$

STEP 7: If largest = second then print "No second largest element"
else print "Second largest = second"

STEP 8: END

PSEUDOCODE :

START

```
READ n
IF n < 2 THEN
  PRINT "No second largest element"
  STOP
END IF
READ array a
IF a[0] > a[1] THEN
  SET largest = a[0]
  SET second = a[1]
ELSE
  SET largest = a[1]
  SET second = a[0]
END IF
SET i = 2
WHILE i < n DO
  IF a[i] > largest THEN
    SET second = largest
    SET largest = a[i]
  ELSE IF a[i] > second AND a[i] != largest THEN
    SET second = a[i]
  END IF
  SET i = i + 1
END WHILE
```

```
IF largest = second THEN
PRINT "No second largest element"
ELSE
PRINT "Second largest =", second
END IF
END
```

CODE :

```
#include <stdio.h>

int main() {
int n, i;
printf("Enter number of elements: ");
scanf("%d", &n);
int a[n];
printf("Enter %d integers:\n", n);
for(i = 0; i < n; i++) {
scanf("%d", &a[i]);
}
int largest, second;
if (n < 2) {
printf("No second largest element.\n");
return 0;
```

```

}
if (a[0] > a[1]) {
    largest = a[0];
    second = a[1];
} else {
    largest = a[1];
    second = a[0];
}
for(i = 2; i < n; i++) {
    if(a[i] > largest) {
        second = largest;
        largest = a[i];
    } else if(a[i] > second && a[i] != largest) {
        second = a[i];
    }
}
if(largest == second)
    printf("No second largest element.\n");
else
    printf("Second largest = %d\n", second);
return 0;
}
printf("Second largest = %d\n", second);

```

```
return 0;  
}
```

OUTPUT :

```
PS E:\Cprogramming works\LAB REPORT CODE> gcc .\array.c  
PS E:\Cprogramming works\LAB REPORT CODE> .\a.exe  
Enter number of elements: 4  
Enter 4 integers:  
2 4 5 9  
Second largest = 5  
PS E:\Cprogramming works\LAB REPORT CODE> █
```

Activity 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.

ALGORITHM:

STEP 1: START

STEP 2: Read the no. of elements (n).

STEP 3: Read all elements and store them into an array **a** of size **n**

STEP 4: Initialize counters: pos, negative, odd, even.

STEP 5: Repeat for $i = 0$ to $i = n$:

if $a[i] > 0$ then increment positive by 1

elseif $a[i] < 0$ then increment negative by 1

if $a[i] \% 2 == 0$ then increment even by 1
else increment odd by 1

STEP 6: Print positive, negative, even, odd.

STEP 7: END

PSEUDOCODE :

START

INPUT n

IF n < 2 THEN

PRINT "No second largest element"

STOP

END IF

READ array a of size n

IF $a[0] > a[1]$ THEN

SET largest = $a[0]$

SET second = $a[1]$

ELSE

SET largest = $a[1]$

SET second = $a[0]$

END IF

SET i = 2

WHILE i < n DO

IF $a[i] > \text{largest}$ THEN

```

SET second = largest
SET largest = a[i]
ELSE IF a[i] > second AND a[i] ≠ largest THEN
SET second = a[i]
END IF
SET i = i + 1
END WHILE
IF largest = second THEN
PRINT "No second largest element"
ELSE
PRINT "Second largest =", second
END IF
END

```

CODE :

```

#include <stdio.h>

int main() {
    int n, i;

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

    int a[n];

    printf("Enter %d integers:\n", n);
    for(i = 0; i < n; i++) {

```

```
scanf("%d", &a[i]);  
}  
  
int positive = 0, negative = 0, odd = 0, even = 0;  
  
for(i = 0; i < n; i++) {  
    if(a[i] > 0)  
        positive++;  
    else if(a[i] < 0)  
        negative++;  
    if(a[i] % 2 == 0)  
        even++;  
    else  
        odd++;  
}  
  
printf("Positive = %d\n", positive);  
printf("Negative = %d\n", negative);  
printf("Odd = %d\n", odd);  
printf("Even = %d\n", even);  
  
return 0;  
}
```

OUTPUT :


```
PS E:\Cprogramming works\LAB REPORT CODE> gcc .\integerarray.c
PS E:\Cprogramming works\LAB REPORT CODE> .\a.exe
Enter number of elements: 3
Enter 3 integers:
2 5 7
Positive = 3
Negative = 0
Odd
Even
Focus folder in explorer (ctrl + click)
PS E:\Cprogramming works\LAB REPORT CODE> |
```

Activity 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.*

ALGORITHM:

STEP 1: START

STEP 2: Read the no. of elements (n).

STEP 3: Read all elements and store them into an array **a** of size **n**

STEP 4: Read the number key whose frequency is to be found.

STEP 5: Initialize count = 0

STEP 6: Repeat for $i = 0$ to $i < n$
if $a[i] == \text{key}$, increment count by 1

STEP 7: Print count.

STEP 8: END

PSEUDOCODE :

```

START
INPUT n
DECLARE array a[n]
PRINT "Enter n integers:"
FOR i = 0 TO n-1
INPUT a[i]
END FOR
PRINT "Enter the number to find frequency:"
INPUT key
SET count = 0
FOR i = 0 TO n-1
IF a[i] == key THEN
count = count + 1
END IF
END FOR
PRINT "Frequency of", key, "=", count
END

```

CODE :

```

#include <stdio.h>

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

printf("Enter number of elements: ");

scanf("%d", &n);

```

```
int a[n];

printf("Enter %d integers:\n", n);

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

scanf("%d", &a[i]);

}

printf("Enter the number to find its frequency: ");

scanf("%d", &key);

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

if(a[i] == key) {

count++;

}

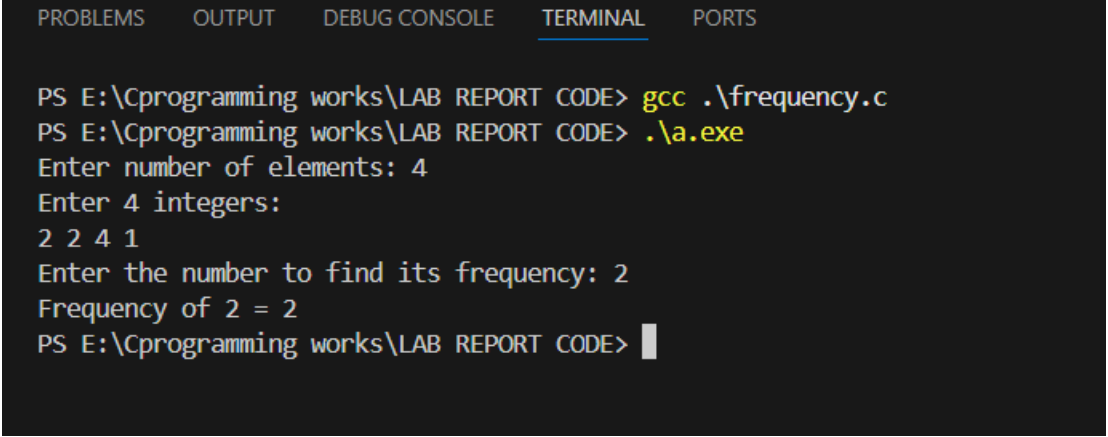
}

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

return 0;

}
```

OUTPUT :



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Cprogramming works\LAB REPORT CODE> gcc .\frequency.c
PS E:\Cprogramming works\LAB REPORT CODE> .\a.exe
Enter number of elements: 4
Enter 4 integers:
2 2 4 1
Enter the number to find its frequency: 2
Frequency of 2 = 2
PS E:\Cprogramming works\LAB REPORT CODE> 
```

Activity 4 : WAP that reads two matrices A ($m \times n$) and B ($p \times q$) 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 incompatibility.

ALGORITHM:

STEP 1: START

STEP 2: Read no. of rows and column matrix A – m & n

STEP 3: Read no. of rows and column matrix B – p & q

STEP 4: If $n \neq p$ **print** “Multiplication not possible” & **goto** STEP 10
else goto STEP5

STEP 5: Read the elements of matrix A

STEP 6: Read the elements of matrix B

STEP 7: Initialize all elements of Matrix C ($m \times q$) to 0

STEP 8: Repeat for $i = 0$ to $i = m - 1$:

for $j = 0$ to $j = q - 1$

for $k = 0$ to $k = n - 1$

$C[i][j] = C[i][j] + A[i][k] \times B[k][j]$

STEP 9: Print Matrix C

STEP 10: END

PSEUDOCODE :

START

// Read matrix dimensions

```
PRINT "Enter rows and columns of Matrix A:"

INPUT m, n

PRINT "Enter rows and columns of Matrix B:"

INPUT p, q

// Check compatibility

IF n != p THEN

PRINT "Matrix multiplication not possible."

PRINT "Because columns of A (", n, ") != rows of B (", p, ")"

STOP

END IF

// Declare matrices A[m][n], B[p][q], C[m][q]

// Read Matrix A in row-major order

PRINT "Enter elements of Matrix A (row-wise):"

FOR i = 0 TO m-1

FOR j = 0 TO n-1

INPUT A[i][j]

END FOR

END FOR

// Read Matrix B in row-major order

PRINT "Enter elements of Matrix B (row-wise):"

FOR i = 0 TO p-1

FOR j = 0 TO q-1

INPUT B[i][j]
```

```

END FOR

END FOR

// Initialize and compute Matrix C = A x B

FOR i = 0 TO m-1
    FOR j = 0 TO q-1
        SET C[i][j] = 0
        FOR k = 0 TO n-1
            C[i][j] = C[i][j] + A[i][k] * B[k][j]
        END FOR
    END FOR
END FOR

// Print resultant matrix

PRINT "Resultant Matrix (A x B):"

FOR i = 0 TO m-1
    FOR j = 0 TO q-1
        PRINT C[i][j], " " // print values in matrix format
    END FOR
    PRINT new line
END FOR

END

```

CODE :

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

```
int main() {  
    int n, i, key, count = 0;  
    printf("Enter number of elements: ");  
    scanf("%d", &n);  
    int a[n];  
    printf("Enter %d integers:\n", n);  
    for(i = 0; i < n; i++) {  
        scanf("%d", &a[i]);  
    }  
    printf("Enter the number to find its frequency: ");  
    scanf("%d", &key);  
    for(i = 0; i < n; i++) {  
        if(a[i] == key) {  
            count++;  
        }  
    }  
    printf("Frequency of %d = %d\n", key, count);  
    return 0;  
}
```

OUTPUT :

```
PS E:\Cprogramming works\LAB REPORT CODE> gcc .\matrix.c
```

```
PS E:\Cprogramming works\LAB REPORT CODE> .\a.exe
```

```
Enter rows and columns of Matrix A: 2 2
```

```
Enter rows and columns of Matrix B: 2 2
```

```
Enter elements of Matrix A (row-wise):
```

```
3 4
```

```
1 5
```

```
Enter elements of Matrix B (row-wise):
```

```
2 5
```

```
2 6
```

```
Resultant Matrix (A x B):
```

```
14 39
```

```
12 35
```

```
PS E:\Cprogramming works\LAB REPORT CODE> █
```