LAB SESSION 3:

SHAYAN HUSSAIN CS-22100

1. Write a program in C to swap elements using call by reference.

```
void swap(int *a, int *b) {
   int temp = *a;
   *a = *b;
   *b = temp;
}
int main() {
   int num1, num2;
   printf("Enter two numbers: ");
   scanf("%d %d", &num1, &num2);
   printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);
   swap(&num1, &num2);
   printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);
   return 0;
}
```

OUTPUT:

```
Enter two numbers: 1 2
Before swapping: num1 = 1, num2 = 2
After swapping: num1 = 2, num2 = 1
```

2. Write a program in C to print a string in reverse using pointers.

```
void printReverse(const char *str) {
   int length = strlen(str);
   const char *ptr = str + length - 1;
   printf("Reversed string: ");
   while (ptr >= str) {
```

```
        printf("%c", *ptr);
        ptr--;
    }
    printf("\n");
}
int main() {
    char inputString[100];
    printf("Enter a string: ");
    fgets(inputString, sizeof(inputString), stdin);
    inputString[strcspn(inputString, "\n")] = '\0';
    printReverse(inputString);
    return 0;
}
```

OUTPUT:

```
Enter a string: SHAYAN
Reversed string: NAYAHS
```

3. Write a C program to input and print array elements using pointers.

```
#include <stdio.h>
#define MAX_SIZE 100
int main() {
    int array[MAX_SIZE];
    int size, i;
    printf("Enter the size of the array (max %d): ", MAX_SIZE);
    scanf("%d", &size);
    if (size <= 0 || size > MAX_SIZE) {
        printf("Invalid size entered. Please enter a size between 1 and %d\n", MAX_SIZE);
        return 1;
    }
```

```
    printf("Enter the elements of the array:\n");
    for (i = 0; i < size; i++) {
        printf("Enter element %d: ", i + 1);
        scanf("%d", &array[i]);
    }
    printf("The array elements are: ");
    int *ptr = array;
    for (i = 0; i < size; i++) {
        printf("%d ", *ptr);
        ptr++;
    }
    printf("\n");
    return 0;
}
```

OUTPUT:

```
Enter the size of the array (max 100): 5
Enter the elements of the array:
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter element 4: 4
Enter element 5: 5
The array elements are: 1 2 3 4 5
```

4. Write a C program to search for an element in an array using pointers.

```
#include <stdio.h>
#define MAX_SIZE 100
int searchElement(int *ptr, int size, int key) {
    int index = -1;
    for (int i = 0; i < size; i++) {
```

```
        if (*ptr == key) {
            index = i;
            break;
        }
        ptr++;
    }
    return index;
}
int main() {
    int array[MAX_SIZE];
    int size, key;
    printf("Enter the size of the array (max %d): ", MAX_SIZE);
    scanf("%d", &size);
    if (size <= 0 || size > MAX_SIZE) {
        printf("Invalid size entered. Please enter a size between 1 and %d\n", MAX_SIZE);
        return 1;
    }
    printf("Enter the elements of the array:\n");
    for (int i = 0; i < size; i++) {
        printf("Enter element %d: ", i + 1);
        scanf("%d", &array[i]);
    }
    printf("Enter the element to search: ");
    scanf("%d", &key);
    int *ptr = array;
    int index = searchElement(ptr, size, key);
    if (index != -1) {
        printf("Element %d found at index %d\n", key, index);
    } else {
```

```
        printf("Element %d not found in the array\n", key);
    }
    return 0;
}
```

OUTPUT:

```
Enter the size of the array (max 100): 4
Enter the elements of the array:
Enter element 1: 1
2Enter element 2:
1
Enter element 3: 2
Enter element 4: 3
Enter the element to search: 3
Element 3 found at index 3
```

5. Write a C program to add two matrices using pointers.

```
#include <stdio.h>
#define MAX_ROWS 100
#define MAX_COLS 100
void addMatrices(int (*mat1)[MAX_COLS], int (*mat2)[MAX_COLS], int (*result)[MAX_COLS], int rows, int cols) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            *(*(result + i) + j) = *(*(mat1 + i) + j) + *(*(mat2 + i) + j);
        }
    }
}
void displayMatrix(int (*matrix)[MAX_COLS], int rows, int cols) {
    printf("Resultant Matrix:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
```

```
            printf("%d\t", *(*(matrix + i) + j));
        }
        printf("\n");
    }
}
int main() {
    int matrix1[MAX_ROWS][MAX_COLS], matrix2[MAX_ROWS][MAX_COLS],
result[MAX_ROWS][MAX_COLS];
    int rows, cols;
    printf("Enter the number of rows and columns of matrices: ");
    scanf("%d %d", &rows, &cols);
    printf("Enter elements of matrix 1:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("Enter element [%d][%d]: ", i + 1, j + 1);
            scanf("%d", &matrix1[i][j]);
        }
    }
    printf("Enter elements of matrix 2:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("Enter element [%d][%d]: ", i + 1, j + 1);
            scanf("%d", &matrix2[i][j]);
        }
    }
    addMatrices(matrix1, matrix2, result, rows, cols);
    displayMatrix(result, rows, cols);
    return 0;
}
```

OUPUT:

```
Enter the number of rows and columns of
    matrices: 2
3
Enter elements of matrix 1:
Enter element [1][1]: 1
Enter element [1][2]: 2
Enter element [1][3]: 3
4Enter element [2][1]: 4
Enter element [2][2]: 5
Enter element [2][3]: 6
Enter elements of matrix 2:
Enter element [1][1]: 7
Enter element [1][2]: 8
Enter element [1][3]: 9
Enter element [2][1]: 10
Enter element [2][2]: 11
Enter element [2][3]: 12
Resultant Matrix:
8   10  12
14  16  18
```