1.Write a C program to create, initialize and use pointers.

Code :-

```
#include <stdio.h>

int main() {
   // Declaration of variables
   int a = 10;
   float b = 20.5;
   char c = 'X';

   // Declaration and initialization of pointers
   int *ptr_a;
   float *ptr_b;
   char *ptr_c;

   // Assigning addresses to pointers
   ptr_a = &a;
   ptr_b = &b;
   ptr_c = &c;

   printf("Value of a: %d\n", *ptr_a);
   printf("Value of b: %.2f\n", *ptr_b);
   printf("Value of c: %c\n", *ptr_c);

   return 0;
}
```

OUTPUT :-

```
Value of a: 10
Value of b: 20.50
Value of c: X
```

2.Write a C program to add two numbers using pointers.

Code :-

```
#include <stdio.h>

int main() {
   int num1, num2, sum;
   int *ptr1, *ptr2, *ptrSum;

   // Prompt the user for input
   printf("Enter the first number: ");
   scanf("%d", &num1);

   printf("Enter the second number: ");
   scanf("%d", &num2);

   // Initialize pointers to the addresses of the variables
   ptr1 = &num1;
   ptr2 = &num2;
   ptrSum = &sum;

   // Perform addition using pointers
   *ptrSum = *ptr1 + *ptr2;

   // Output the result
   printf("The sum of %d and %d is %d \n",*ptr1, *ptr2, *ptrSum);

   return 0;
}
```

OUTPUT :-

```
Enter the first number: 5
Enter the second number: 7
The sum of 5 and 7 is 12
```

3.Write a C program to swap two numbers using pointers.

Code :-

```
#include<stdio.h>
int main(){
        int a,b,c;
        int *ptr_a,*ptr_b;
        printf("enter the 1st no :");
        scanf("%d",&a);
        printf("enter the 2nd no :");
        scanf("%d",&b);
        ptr_a=&a;
        ptr_b=&b;
        printf("Before Swaping :a=%d,b=%d\n",*ptr_a,*ptr_b);
        c=*ptr_a;
        *ptr_a=*ptr_b;
        *ptr_b=c;
        printf("After Swaping :a=%d,b=%d\n",*ptr_a,*ptr_b);
        return 0;
}
```

OUTPUT :-

```
enter the 1st no :5
enter the 2nd no :6
Before Swaping :a=5,b=6
After Swaping :a=6,b=5
```

4.Write a C program to input and print array elements using pointer.

Code :-

```
#include<stdio.h>
int main()
{
	int n,arr[100],i;
	printf("enter how many array elements :");
	scanf("%d",&n);
	printf("enter the elements are :\n");
	for(i=0;i<n;i++){
		scanf("%d",&(*(arr+i)));
	}
	printf("Array elements are :\n");
	for(i=0;i<n;i++){
		printf("%d",*(arr+i));
	}
	return 0;
}
```

OUTPUT :-

```
enter how many array elements :5
enter the elements are :
2
4
6
8
1
Array elements are :
24681
```

5.Write a C program to copy one array to another using pointers.

Code :-

```
#include <stdio.h>
void copyArray(int *source, int *destination, int size) {
   for (int i = 0; i < size; i++) {
      *(destination + i) = *(source + i); // Copying elements using pointers
   }
}

int main() {
   int size;

   // Asking user for the size of the array
   printf("Enter the size of the array: ");
   scanf("%d", &size);

   int sourceArray[size]; // Declare the source array
   int destinationArray[size]; // Declare the destination array

   // Getting user input for the source array
   printf("Enter %d elements for the array:\n", size);
   for (int i = 0; i < size; i++) {
      printf("Element %d: ", i + 1);
      scanf("%d", &sourceArray[i]);
   }

   // Copying the source array to the destination array
   copyArray(sourceArray, destinationArray, size);
   // Displaying the copied array
   printf("Copied Array: ");
   for (int i = 0; i < size; i++) {
      printf("%d ", destinationArray[i]);
   }
   printf("\n");
   return 0;
}
```

OUTPUT :-

```
Enter the size of the array: 4
Enter 4 elements for the array:
Element 1: 10
Element 2: 30
Element 3: 50
Element 4: 60
Copied Array: 10 30 50 60
```

6.Write a C program to swap two arrays using pointers.

Code :-

```
#include <stdio.h>

void swapArrays(int *array1, int *array2, int size) {
   for (int i = 0; i < size; i++) {
      // Swap elements using a temporary variable
      int temp = *(array1 + i);
      *(array1 + i) = *(array2 + i);
      *(array2 + i) = temp;
   }
}

int main() {
   int size;

   // Asking user for the size of the arrays
   printf("Enter the size of the arrays: ");
   scanf("%d", &size);

   int array1[size], array2[size];

   // Getting user input for the first array
   printf("Enter %d elements for the first array:\n", size);
   for (int i = 0; i < size; i++) {
      printf("Element %d: ", i + 1);
      scanf("%d", &array1[i]);
   }

   // Getting user input for the second array
   printf("Enter %d elements for the second array:\n", size);
   for (int i = 0; i < size; i++) {
      printf("Element %d: ", i + 1);
      scanf("%d", &array2[i]);
   }

   // Swapping the arrays
   swapArrays(array1, array2, size);

   // Displaying the swapped arrays
   printf("\nAfter swapping:\n");
   printf("First Array: ");
   for (int i = 0; i < size; i++) {
      printf("%d ", array1[i]);
```

```
    }
    printf("\nSecond Array: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", array2[i]);
    }
    printf("\n");

    return 0;
}
```

OUTPUT :-

```
Enter the size of the arrays: 4
Enter 4 elements for the first array:
Element 1: 1
Element 2: 2
Element 3: 3
Element 4: 4
Enter 4 elements for the second array:
Element 1: 4
Element 2: 5
Element 3: 6
Element 4: 7

After swapping:
First Array: 4 5 6 7
Second Array: 1 2 3 4
```

7.Write a C program to reverse an array using pointers.

Code :-

```
#include <stdio.h>

void reverseArray(int *array, int size) {
   int *start = array;          // Pointer to the start of the array
   int *end = array + size - 1; // Pointer to the end of the array
   int temp;

   while (start < end) {
      // Swap the values at start and end
      temp = *start;
      *start = *end;
      *end = temp;

      // Move the pointers towards the center
      start++;
      end--;
   }
}

int main() {
   int size;

   // Asking user for the size of the array
   printf("Enter the size of the array: ");
   scanf("%d", &size);

   int array[size]; // Declare the array

   // Getting user input for the array
   printf("Enter %d elements for the array:\n", size);
   for (int i = 0; i < size; i++) {
      printf("Element %d: ", i + 1);
      scanf("%d", &array[i]);
   }

   // Reversing the array
   reverseArray(array, size);

   // Displaying the reversed array
   printf("Reversed Array: ");
   for (int i = 0; i < size; i++) {
      printf("%d ", array[i]);
```

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

    return 0;
}
```

OUTPUT :-

```
Enter the size of the array: 5
Enter 5 elements for the array:
Element 1: 2
Element 2: 3
Element 3: 4
Element 4: 5
Element 5: 6
Reversed Array: 6 5 4 3 2
```

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

Code :-

```
#include <stdio.h>
int searchElement(int *array, int size, int target) {
   for (int i = 0; i < size; i++) {
      if (*(array + i) == target) {
         return i; // Return the index if the element is found
      }
   }
   return -1; // Return -1 if the element is not found
}

int main() {
   int size, target, index;

   // Asking user for the size of the array
   printf("Enter the size of the array: ");
   scanf("%d", &size);

   int array[size]; // Declare the array

   // Getting user input for the array
   printf("Enter %d elements for the array:\n", size);
   for (int i = 0; i < size; i++) {
      printf("Element %d: ", i + 1);
      scanf("%d", &array[i]);
   }

   // Asking user for the element to search
   printf("Enter the element to search: ");
   scanf("%d", &target);

   // Searching the element in the array
   index = searchElement(array, size, target);

   // Displaying the result
   if (index != -1) {
      printf("Element %d found at index %d.\n", target, index);
   } else {
      printf("Element %d not found in the array.\n", target);
   }
```

```
    return 0;
}
```

OUTPUT :-

```
Enter the size of the array: 4
Enter 4 elements for the array:
Element 1: 1
Element 2: 2
Element 3: 3
Element 4: 4
Enter the element to search: 3
Element 3 found at index 2.
```

9.Write a C program to access two dimensional array using pointers.

Code :-

```
#include <stdio.h>

#define ROWS 3
#define COLS 4

void printArray(int (*array)[COLS], int rows) {
   for (int i = 0; i < rows; i++) {
      for (int j = 0; j < COLS; j++) {
         printf("%d ", *(*(array + i) + j)); // Accessing elements using pointers
      }
      printf("\n");
   }
}

int main() {
   int array[ROWS][COLS]; // Declare a 2D array

   // Getting user input for the 2D array
   printf("Enter elements for a %dx%d array:\n", ROWS, COLS);
   for (int i = 0; i < ROWS; i++) {
      for (int j = 0; j < COLS; j++) {
         printf("Element [%d][%d]: ", i, j);
         scanf("%d", &array[i][j]);
      }
   }

   // Displaying the array
   printf("\nThe 2D array is:\n");
   printArray(array, ROWS);

   return 0;
}
```

OUTPUT :-

```
Enter elements for a 3x4 array:
Element [0][0]: 1
Element [0][1]: 2
Element [0][2]: 3
Element [0][3]: 4
Element [1][0]: 6
Element [1][1]: 7
```

Element [1][2]: 8
Element [1][3]: 9
Element [2][0]: 2
Element [2][1]: 5
Element [2][2]: 4
Element [2][3]: 3

The 2D array is:
1 2 3 4
6 7 8 9
2 5 4 3

10.Write a C program to add two matrix and multiply two matrix using pointers.

Code :-

```
#include <stdio.h>

#define MAX 10 // Maximum size of the matrix

// Function to add two matrices
void addMatrices(int (*a)[MAX], int (*b)[MAX], int (*result)[MAX], int rows, int cols) {
   for (int i = 0; i < rows; i++) {
      for (int j = 0; j < cols; j++) {
         *(*(result + i) + j) = *(*(a + i) + j) + *(*(b + i) + j);
      }
   }
}

// Function to multiply two matrices
void multiplyMatrices(int (*a)[MAX], int (*b)[MAX], int (*result)[MAX], int r1, int c1, int c2) {
   for (int i = 0; i < r1; i++) {
      for (int j = 0; j < c2; j++) {
         *(*(result + i) + j) = 0; // Initialize result[i][j] to 0
         for (int k = 0; k < c1; k++) {
            *(*(result + i) + j) += *(*(a + i) + k) * *(*(b + k) + j);
         }
      }
   }
}

// Function to print a matrix
void printMatrix(int (*matrix)[MAX], int rows, int cols) {
   for (int i = 0; i < rows; i++) {
      for (int j = 0; j < cols; j++) {
         printf("%d ", *(*(matrix + i) + j));
      }
      printf("\n");
   }
}

int main() {
   int a[MAX][MAX], b[MAX][MAX], sum[MAX][MAX], product[MAX][MAX];
   int r1, c1, r2, c2;
```

```
// Input dimensions for the first matrix
printf("Enter rows and columns for first matrix: ");
scanf("%d %d", &r1, &c1);

// Input elements for the first matrix
printf("Enter elements of first matrix:\n");
for (int i = 0; i < r1; i++) {
   for (int j = 0; j < c1; j++) {
      printf("Element [%d][%d]: ", i, j);
      scanf("%d", &a[i][j]);
   }
}

// Input dimensions for the second matrix
printf("Enter rows and columns for second matrix: ");
scanf("%d %d", &r2, &c2);

// Check if the matrices can be added or multiplied
if (r1 != r2) {
   printf("Matrices cannot be added due to incompatible dimensions.\n");
   return 1;
}

if (c1 != r2) {
   printf("Matrices cannot be multiplied due to incompatible dimensions.\n");
   return 1;
}

// Input elements for the second matrix
printf("Enter elements of second matrix:\n");
for (int i = 0; i < r2; i++) {
   for (int j = 0; j < c2; j++) {
      printf("Element [%d][%d]: ", i, j);
      scanf("%d", &b[i][j]);
   }
}

// Adding the matrices
addMatrices(a, b, sum, r1, c1);
printf("\nSum of matrices:\n");
printMatrix(sum, r1, c1);

// Multiplying the matrices
```

```
    multiplyMatrices(a, b, product, r1, c1, c2);
    printf("\nProduct of matrices:\n");
    printMatrix(product, r1, c2);

    return 0;
}
```

OUTPUT :-

```
Enter rows and columns for first matrix: 2
2
Enter elements of first matrix:
Element [0][0]: 1
Element [0][1]: 2
Element [1][0]: 3
Element [1][1]: 4
Enter rows and columns for second matrix: 2
2
Enter elements of second matrix:
Element [0][0]: 3
Element [0][1]: 4
Element [1][0]: 5
Element [1][1]: 6

Sum of matrices:
4 6
8 10

Product of matrices:
13 16
29 36

Enter rows and columns for first matrix: 2
3
Enter elements of first matrix:
Element [0][0]: 1
Element [0][1]: 2
Element [0][2]: 3
Element [1][0]: 4
Element [1][1]: 5
Element [1][2]: 6
Enter rows and columns for second matrix: 3
2
Matrices cannot be added due to incompatible dimensions.
```