```c
// Program for illustrating the use of double pointer
#include <stdio.h>

int main()
{
    // Declare and initialize variables
    int x = 10;

    // Declare pointers
    int *p = &x;
    int **q = &p;

    // Output: Print values and addresses
    printf("Value of x = %d\n", x);
    printf("Address of x = %p\n", p);
    printf("Value at address of x = %d\n", *p);
    printf("Address of p = %p\n", q);
    printf("Value at value at address of x = %d\n", **q);

    return 0;
}

```

```
manish@fedora: ~/vs-code/bca-programming-repo/C/pointe
$ ./question1
Value of x = 10
Address of x = 0x7ffdfeb06f64
Value at address of x = 10
Address of p = 0x7ffdfeb06f58
Value at value at address of x = 10
```

```c
// Program to illustrating the use of array of pointers.
#include <stdio.h>

int main()
{
    // Declare and initialize variables
    int a = 1, b = 12, c = 3, d = 4;

    // Declare an array of pointers
    int *p[4] = {&a, &b, &c, &d};

    // Output: Print values and addresses using a loop
    for (int i = 0; i < 4; i++)
    {
        printf("Value of var%d: %d\tAddress: %p\n", i + 1, *p[i], p[i]);
    }

    return 0;
}

```

```
manish@fedora: ~/vs-code/bca-programming-repo/C/pointe
$ ./question2
Value of var1: 1        Address: 0x7ffcb1a7c178
Value of var2: 12       Address: 0x7ffcb1a7c174
Value of var3: 3        Address: 0x7ffcb1a7c170
Value of var4: 4        Address: 0x7ffcb1a7c16c

manish@fedora: ~/vs-code/bca-programming-repo/C/pointe
```

```c
// Program to demonstrate the accessing of a value by two pointer variables.
#include <stdio.h>

int main()
{
    // Declare and initialize variables
    int a = 5;

    // Declare pointers and initialize them accordingly
    int *pa = &a;
    int **qa = &pa;

    // Output: Print the values using pointers
    printf("a = %d, *pa = %d, **qa = %d\n", a, *pa, **qa);

    return 0;
}
```

```
manish@fedora: ~/vs-code/bca-programming-repo/C/pointe
$ ./question3
a = 5, *pa = 5, **qa = 5

manish@fedora: ~/vs-code/bca-programming-repo/C/pointe
$
```

```c
// Program to swap pointer values
#include <stdio.h>

int main()
{
    // Declare and initialize variables
    int a = 5, b = 12;

    // Declare and initialize pointers
    int *pa = &a, *pb = &b;

    // Output: Print initial values using pointers
    printf("*pa = %d, *pb = %d\n", *pa, *pb);

    // Swap values using pointers
    int temp = *pa;
    *pa = *pb;
    *pb = temp;

    // Output: Print values after swapping
    printf("*pa = %d, *pb = %d\n", *pa, *pb);

    return 0;
}

```

```
manish@fedora: ~/vs-code/bca-programming-repo/C/pointe
$ ./question4
*pa = 5, *pb = 12
*pa = 12, *pb = 5

manish@fedora: ~/vs-code/bca-programming-repo/C/pointe
$
```

```c
/*Program that lower case letter to upper case and upper case to lower case
by passing of pointer to function*/
#include <stdio.h>

void conversion(char *c)
{
    if (*c >= 'a' && *c <= 'z')
        *c -= 32;
    else if (*c >= 'A' && *c <= 'Z')
        *c += 32;
}

int main()
{
    char ch;

    // Input: Prompt the user to enter a character
    printf("Enter a character: ");
    scanf("%c", &ch);

    // Call the conversion function
    conversion(&ch);

    // Output: Display the corresponding character
    printf("The corresponding character is %c.\n", ch);

    return 0;
}

```

```
manish@fedora: ~/vs-code/bca-programming-repo/C/pointe
$ ./question5
Enter a character: d
The corresponding character is D.

manish@fedora: ~/vs-code/bca-programming-repo/C/pointe
$ ./question5
Enter a character: W
The corresponding character is w.
```

```c
// Program to demonstrate the relationship between arrays and pointer.
#include <stdio.h>

int main()
{
    // Declare an array of integers
    int numbers[] = {10, 20, 30, 40, 50};

    // Declare a pointer to an integer
    int *ptr;

    // Point the pointer to the first element of the array
    ptr = numbers;

    // Access array elements using the pointer
    printf("Array elements using pointer:\n");
    for (int i = 0; i < 5; i++)
    {
        printf("Element %d: %d\n", i + 1, *ptr);
        // Move the pointer to the next element in the array
        ptr++;
    }

    // Note: After the loop, the pointer has moved beyond the end of the array

    return 0;
}

```

```
manish@fedora: ~/vs-code/bca-programming-repo/C/pointe
$ ./question6
Array elements using pointer:
Element 1: 10
Element 2: 20
Element 3: 30
Element 4: 40
Element 5: 50

manish@fedora: ~/vs-code/bca-programming-repo/C/pointe
```

```c
// Program to display all the elements of two dimensional array using pointer.
#include <stdio.h>
int main()
{
    int a[2][3] = {{10, 20, 30}, {40, 22, 125}};
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            printf("%d\t", *(*(a + i) + j));
        }
        printf("\n");
    }
    return 0;
}
```

```
manish@fedora: ~/vs-code/bca-programming-repo/C/pointe
$ ./question7
10      20      30
40      22      125

manish@fedora: ~/vs-code/bca-programming-repo/C/pointe
$
```

```c
// Program to add two m*n matrices using pointer.
#include <stdio.h>

void addMatrices(int m, int n, int mat1[][n], int mat2[][n], int result[][n])
{
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            *(*(result + i) + j) = *(*(mat1 + i) + j) + *(*(mat2 + i) + j);
        }
    }
}

void displayMatrix(int m, int n, int mat[][n])
{
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            printf("%d ", *(*(mat + i) + j));
        }
        printf("\n");
    }
}

int main()
{
    int m, n;

    // Input matrix dimensions
    printf("Enter number of rows (m) and columns (n) for matrices: ");
    scanf("%d %d", &m, &n);

    int mat1[m][n], mat2[m][n], result[m][n];

    // Input elements of the first matrix
    printf("Enter elements of the first matrix:\n");
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            scanf("%d", (*(mat1 + i) + j));
        }
    }

    // Input elements of the second matrix
    printf("Enter elements of the second matrix:\n");
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            scanf("%d", (*(mat2 + i) + j));
        }
    }

    // Add matrices
    addMatrices(m, n, mat1, mat2, result);

    // Display matrices and result
    printf("\nMatrix 1:\n");
    displayMatrix(m, n, mat1);

    printf("\nMatrix 2:\n");
    displayMatrix(m, n, mat2);

    printf("\nSum of Matrices:\n");
    displayMatrix(m, n, result);

    return 0;
}

```

```
manish@fedora: ~/vs-code/bca-programming-repo/C/pointer main!
$ ./question8
Enter number of rows (m) and columns (n) for matrices: 2 3
Enter elements of the first matrix:
1 2 3 4 5 6
Enter elements of the second matrix:
10 12 11 15 6 2

Matrix 1:
1 2 3
4 5 6

Matrix 2:
10 12 11
15 6 2

Sum of Matrices:
11 14 14
19 11 8
```

```c
// Program to demonstrate the relationship between string and pointer.
#include <stdio.h>
#include <string.h>
int main()
{
    char *namaste = "NAMASKAR SIR";
    // char namaste[20] =  "NAMASKAR SIR";
    char name[40];
    printf("Enter your name: ");
    scanf("%[^\n]s", name);
    puts(namaste);
    printf("Namaskar %s Sir", name);
    return 0;
}
```

```
manish@fedora: ~/vs-code/bca-programming-repo/C/pointer main!
$ ./question9
Enter your name: Manish
NAMASKAR SIR
Namaskar Manish Sir%

manish@fedora: ~/vs-code/bca-programming-repo/C/pointer main!
$
```