Question 45

Q: Write a Program to perform Linear search on an array.

Source Code

//Program to perform Linear search on an array.

#include<stdio.h>

int main(){

    int elm,arr[50],i,srch,flag;

    printf("How Many Elements you want to Enter: ");

    scanf("%d", &elm);

    for(i=0;i<elm;i++){

        printf("Enter Element %d: ", i+1);

        scanf("%d", &arr[i]);

    }

    printf("Enter the Integer to Search: ");

    scanf("%d",&srch);

    for(i=0;i<elm;i++){

        if(arr[i] == srch){

            flag = 1;

            break;

        }

    }

    if(flag == 1){

        printf("%d found at %d position", srch,i+1);

    }

    else{

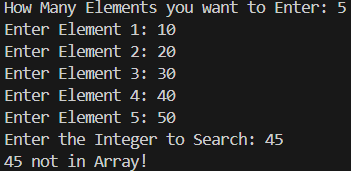
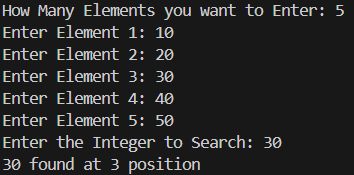
        printf("%d not in Array!", srch);

    }

    return 0;

}

Output



Question 46

Q: Program to generate reverse array for a given array.

Source Code

//Program to generate reverse array for a given array.

#include<stdio.h>

int main(){

    int arr[50],arrDup[50],elm,i;

    printf("How Many Elements you want to Enter: ");

    scanf("%d", &elm);

    for(i=0;i<elm;i++){

        printf("Enter Element %d: ", i+1);

        scanf("%d", &arr[i]);

    }

    for(i=elm;i>=1;i--){

        arrDup[elm-i] = arr[i-1];

    }

    for(i=0;i<elm;i++){

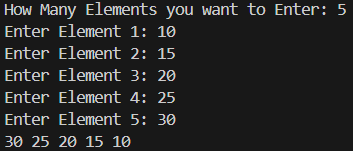
        printf("%d ", arrDup[i]);

    }

    return 0;

}

Output



Question 47

Q: Program to perform Matrix Operations (switch-case):

Addition, Subtraction, Multiplication and Transpose.

Source Code

/\*Program to perform Matrix Operations (switch-case):

Addition, Subtraction, Multiplication and Transpose\*/

#include<stdio.h>

void inputMatrix(int n[3][3]){

    int i, j;

    printf("Enter a 3x3 Matrix:\n");

    // Input values in the matrix

    for(i = 0; i < 3; i++){

        for(j = 0; j < 3; j++){

            printf("Enter number @ row %d column %d: ", i + 1, j + 1);

            scanf("%d", &n[i][j]);

        }

    }

}

void printMatrix(int a[3][3]){

    // Print the matrix

    int i, j;

    printf("Matrix:\n");

    for(i = 0; i < 3; i++){

        for(j = 0; j < 3; j++){

            printf("%d ", a[i][j]);

        }

        printf("\n");

    }

}

void addMatrix(int a[3][3], int b[3][3], int result[3][3]){

    // Adding two matrices

    int i, j;

    for(i = 0; i < 3; i++){

        for(j = 0; j < 3; j++){

            result[i][j] = a[i][j] + b[i][j];

        }

    }

}

void subtractMatrix(int a[3][3], int b[3][3], int result[3][3]){

    // Subtracting two matrices

    int i, j;

    for(i = 0; i < 3; i++){

        for(j = 0; j < 3; j++){

            result[i][j] = a[i][j] - b[i][j];

        }

    }

}

void multiplyMatrix(int a[3][3], int b[3][3], int result[3][3]){

    // Multiplying two matrices

    int i, j, k;

    for(i = 0; i < 3; i++){

        for(j = 0; j < 3; j++){

            result[i][j] = 0;

            for(k = 0; k < 3; k++){

                result[i][j] += a[i][k] \* b[k][j];

            }

        }

    }

}

void transposeMatrix(int a[3][3], int result[3][3]){

    // Transposing the matrix

    int i, j;

    for(i = 0; i < 3; i++){

        for(j = 0; j < 3; j++){

            result[j][i] = a[i][j];

        }

    }

}

int main(){

    int a[3][3], b[3][3], result[3][3];

    char choice;

    printf("Choose operation:\n");

    printf("Addition (+)\nSubtraction (-)\nMultiplication (x)\nTranspose (t)\n");

    scanf(" %c", &choice);  //space before %c to ignore newline character from previous input

    switch (choice){

        case '+':

            printf("Input first matrix:\n");

            inputMatrix(a);

            printf("Input second matrix:\n");

            inputMatrix(b);

            addMatrix(a, b, result);

            printf("Result of matrix addition:\n");

            printMatrix(result);

            break;

        case '-':

            printf("Input first matrix:\n");

            inputMatrix(a);

            printf("Input second matrix:\n");

            inputMatrix(b);

            subtractMatrix(a, b, result);

            printf("Result of matrix subtraction:\n");

            printMatrix(result);

            break;

        case 'x':

            printf("Input first matrix:\n");

            inputMatrix(a);

            printf("Input second matrix:\n");

            inputMatrix(b);

            multiplyMatrix(a, b, result);

            printf("Result of matrix multiplication:\n");

            printMatrix(result);

            break;

        case 't':

            printf("Input matrix to transpose:\n");

            inputMatrix(a);

            transposeMatrix(a, result);

            printf("Transpose of the matrix:\n");

            printMatrix(result);

            break;

        default:

            printf("Invalid operation!\n");

            break;

    }

    return 0;

}

Output

|  |  |
| --- | --- |
| Choose operation:  Addition (+)  Subtraction (-)  Multiplication (x)  Transpose (t)  +  Input first matrix:  Enter a 3x3 Matrix:  Enter number @ row 1 column 1: 1  Enter number @ row 1 column 2: 2  Enter number @ row 1 column 3: 3  Enter number @ row 2 column 1: 4  Enter number @ row 2 column 2: 5  Enter number @ row 2 column 3: 6  Enter number @ row 3 column 1: 7  Enter number @ row 3 column 2: 8  Enter number @ row 3 column 3: 9  Input second matrix:  Enter a 3x3 Matrix:  Enter number @ row 1 column 1: 9  Enter number @ row 1 column 2: 8  Enter number @ row 1 column 3: 7  Enter number @ row 2 column 1: 6  Enter number @ row 2 column 2: 5  Enter number @ row 2 column 3: 4  Enter number @ row 3 column 1: 3  Enter number @ row 3 column 2: 2  Enter number @ row 3 column 3: 1  Result of matrix addition:  Matrix:  10 10 10  10 10 10  10 10 10 | Choose operation:  Addition (+)  Subtraction (-)  Multiplication (x)  Transpose (t)  x  Input first matrix:  Enter a 3x3 Matrix:  Enter number @ row 1 column 1: 1  Enter number @ row 1 column 2: 2  Enter number @ row 1 column 3: 3  Enter number @ row 2 column 1: 4  Enter number @ row 2 column 2: 5  Enter number @ row 2 column 3: 6  Enter number @ row 3 column 1: 7  Enter number @ row 3 column 2: 8  Enter number @ row 3 column 3: 9  Input second matrix:  Enter a 3x3 Matrix:  Enter number @ row 1 column 1: 1  Enter number @ row 1 column 2: 2  Enter number @ row 1 column 3: 3  Enter number @ row 2 column 1: 4  Enter number @ row 2 column 2: 5  Enter number @ row 2 column 3: 6  Enter number @ row 3 column 1: 7  Enter number @ row 3 column 2: 8  Enter number @ row 3 column 3: 9  Result of matrix multiplication:  Matrix:  30 36 42  66 81 96  102 126 150 |
| Choose operation:  Addition (+) Subtraction (-) Multiplication (x) Transpose (t): t  Input matrix to transpose:  Enter a 3x3 Matrix:  Enter number @ row 1 column 1: 1 Enter number @ row 1 column 2: 2  Enter number @ row 1 column 3: 3 Enter number @ row 2 column 1: 4  Enter number @ row 2 column 2: 5 Enter number @ row 2 column 3: 6  Enter number @ row 3 column 1: 7 Enter number @ row 3 column 2: 8  Enter number @ row 3 column 3: 9  Transpose of the matrix:  Matrix:  1 4 7  2 5 8  3 6 9 | |

Question 48

Q: Program to read character array using getchar() in do-while loop and print it.

Find its length and number of vowels (Case-sensitive)

Source Code

/\*Program to read character array using getchar() in do-while loop and print it.

Find its length and number of vowels (Case-sensitive)\*/

#include<stdio.h>

int main() {

    char str[100],ch;

    int length = 0,vowels = 0,i = 0;

    printf("Enter a string (end input with a newline): ");

    do {

        ch = getchar();

        str[i++] = ch;

    }while(ch != '\n' && i < 100);

    str[i-1] = '\0';

    // Loop to calculate length and number of vowels

    for(i = 0; str[i] != '\0'; i++) {

        length++;

        if(str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U' ||

           str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u') {

            vowels++;

        }

    }

    printf("You entered: %s\n", str);

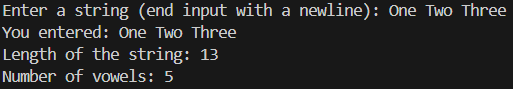
    printf("Length of the string: %d\n", length);

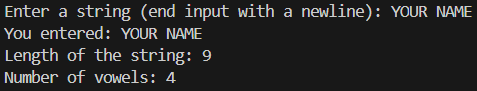
    printf("Number of vowels: %d\n", vowels);

    return 0;

}

Output





Question 49

Q: Program to find reverse of a string (without inbuilt function).

Source Code

//Program to find reverse of a string (without inbuilt function).

#include<stdio.h>

int main(){

    char str[100],strRev[100],ch;

    int length = 0,vowels = 0,i = 0,len;

    printf("Enter a string: ");

    do {

        ch = getchar();

        str[i++] = ch;

    }while(ch != '\n' && i < 100);

    str[i-1] = '\0';

    for(len=0;str[len]!='\0';len++); // to calculate length of string array

    i=0;

    while(str[i]!='\0'){

        strRev[i] = str[len-1];

        i++;

        len--;

    }

    strRev[i] = '\0';

    for(i=0;strRev[i]!='\0';i++){

        printf("%c", strRev[i]);

    }

    return 0;

}

Output





Question 50

Q: Program to Compare and Concatenate two string (without inbuilt function).

Source Code

//Program to compare and concatenate two string (without inbuilt function).

#include<stdio.h>

int main(){

    char str1[50],str2[50];

    printf("\nEnter Two Strings: ");

    scanf("%s %s", str1, str2);

    char fullStr[150];

    int i,j;

    //Concatenating Strings

    for(i=0; str1[i]!='\0'; i++){

        fullStr[i] = str1[i];

    }

    fullStr[i] = ' ';

    for(j=0; str2[j]!='\0'; j++){

        fullStr[i+j+1] = str2[j];

    }

    fullStr[i+j+1] = ' ';

    printf("Concatenated String is:\n");

    printf("%s", fullStr);

    //Comparing Strings

    i=0;

    while(str1[i] == str2[i] && str1[i] != '\0' && str2[i] !='\0'){

        i++;

    }

    if(str1[i]=='\0' && str2[i]=='\0'){

        printf("\nBoth the Strings have Equal Values!");

    }

    else{

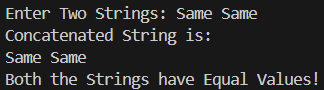
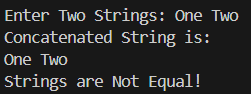
        printf("\nStrings are Not Equal!");

    }

    return 0;

}

Output



Question 51

Q: Program to copy a string to another string (without inbuilt function.)

Source Code

//Program to copy a string to another string (without inbuilt function.)

#include<stdio.h>

int main(){

    char str[50], strCopy[50];

    int i;

    printf("Enter String: ");

    scanf("%s", str);

    for(i=0;str[i]!='\0';i++){

        strCopy[i] = str[i];

    }

    printf("The Copied String is %s.\n", strCopy);

    return 0;

}

Output





Question 52

Q: Program to show the use of string function: strcpy(), strcat(), strcmp(), strlen().

Source Code

//Program to show the use of string function: strcpy(), strcat(), strcmp(), strlen().

#include<stdio.h>

#include<string.h>  // Include the string.h library for string functions

int main() {

    char str1[50], str2[50], str3[50];

    // strcpy() - Copies the contents of one string into another

    printf("Enter the first string: ");

    gets(str1);

    strcpy(str3, str1);

    printf("After copying, str3 = %s\n", str3);

    // strcat() - Concatenates (appends) one string at the end of another

    printf("Enter the second string: ");

    gets(str2);

    strcat(str1, str2);

    printf("After concatenation, str1 = %s\n", str1);

    // strcmp() - Compares two strings lexicographically

    if(strcmp(str1, str3) == 0)

        printf("str1 and str3 are the same.\n");

    else

        printf("str1 and str3 are different.\n");

    // strlen() - Finds the length of a string

    printf("Length of str1: %lu\n", strlen(str1));

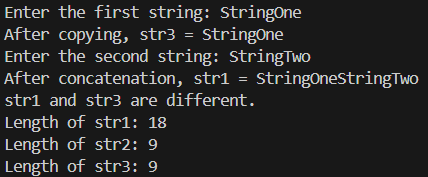
    printf("Length of str2: %lu\n", strlen(str2));

    printf("Length of str3: %lu\n", strlen(str3));

    return 0;

}

Output



Question 53

Q: Program to find if a given string is palindrome or not.

Source Code

//Program to find if a given string is palindrome or not.

#include<stdio.h>

#define palindromeYes printf("%s is a Palindrome Number!", str)

#define palindromeNo printf("%s is NOT a Palindrome Number!", str)

int main(){

    char str[50];

    int len,i, isPalindrome;

    printf("Enter String: ");

    scanf("%s", str);

    for(len=0;str[len]!='\0';len++);

    for(i=0;i<(len/2);i++){

        if(str[i] != str[len-1-i]){

            isPalindrome = 0;

            break;

        }

        else{

            isPalindrome = 1;

        }

    }

    (isPalindrome)? palindromeYes : palindromeNo;

    return 0;

}

Output





