C-Programming Assignments

Q1. Write a C program to print the given patterns:

1.

\*

\* \*

\* \* \*

\* \* \* \*

Code:

#include <stdio.h>

int main() {

int rows = 4; // number of rows in the pattern

for(int i = 1; i <= rows; i++) {

for(int j = 1; j <= i; j++) {

printf("\*\t"); // print asterisk and tab character

}

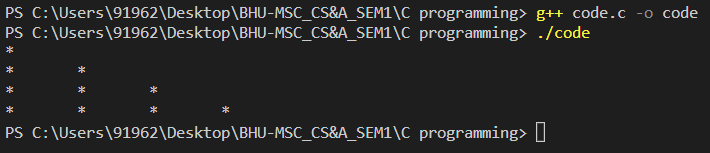
printf("\n"); // move to the next line

}

return 0;

}

Output:



2. A full pyramid of stars.

|  |  | \* |  |  |
| --- | --- | --- | --- | --- |
|  | \* | \* | \* |  |
| \* | \* | \* | \* | \* |

#include <stdio.h>

int main() {

int rows = 3; // number of rows in the pyramid

for(int i = 1; i <= rows; i++) {

for(int j = 1; j <= rows - i; j++) {

printf("\t"); // print tab character for spacing

}

for(int k = 1; k <= 2\*i-1; k++) {

printf("\*\t"); // print asterisk and tab character

}

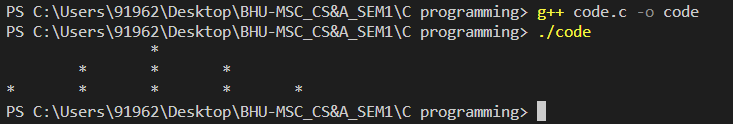
printf("\n"); // move to the next line

}

return 0;

}

Output:



3. A hollow pyramid of stars.

|  |  |  | \* |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | \* |  | \* |  |  |
|  | \* |  |  |  | \* |  |
| \* | \* | \* | \* | \* | \* | \* |

#include <stdio.h>

int main() {

int rows = 4; // number of rows in the pyramid

for(int i = 1; i <= rows; i++) {

for(int j = 1; j <= rows - i; j++) {

printf("\t"); // print tab character for left-side spacing

}

for(int k = 1; k <= 2\*i-1; k++) {

if(k == 1 || k == 2\*i-1 || i == rows) {

printf("\*\t"); // print asterisk for first and last position or for last row

} else {

printf("\t"); // otherwise print tab character for hollow pyramid

}

}

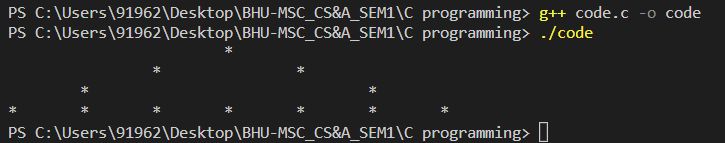
printf("\n"); // move to the next line

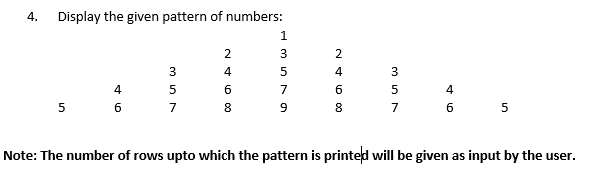
}

return 0;

}

Output:





#include <stdio.h>

int main() {

int rows;

printf("Enter the number of rows: ");

scanf("%d", &rows); // take input from the user

int num = 1; // start with number 1

for(int i = 1; i <= rows; i++) {

for(int j = 1; j <= rows - i; j++) {

printf("\t"); // print tab character for left-side spacing

}

for(int k = 1; k <= 2\*i-1; k++) {

printf("%d\t", num); // print the number and tab character

if(k < i) {

num++; // increment the number up to middle column

} else {

num--; // decrement the number after middle column

}

}

num += 2; // set the number for the next row

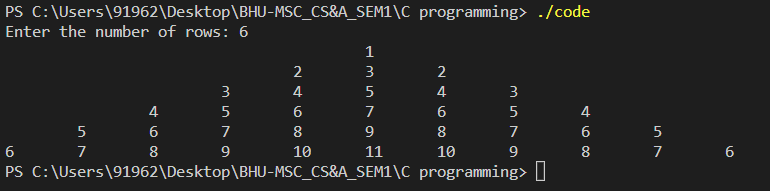
printf("\n"); // move to the next line

}

return 0;

}

Output:



Q2. Find the ASCII value of a given character.

#include <stdio.h>

int main() {

char c;

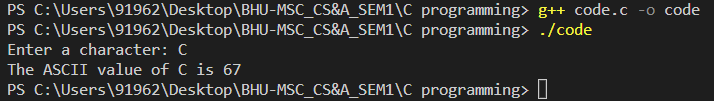
printf("Enter a character: ");

scanf("%c", &c); // take input from the user

printf("The ASCII value of %c is %d", c, c); // print the ASCII value of the character

return 0;

}



Q3. Find sum of N Natural Numbers.

#include <stdio.h>

int main() {

int n, sum = 0;

printf("Enter a positive integer: ");

scanf("%d", &n); // take input from the user

for(int i = 1; i <= n; i++) {

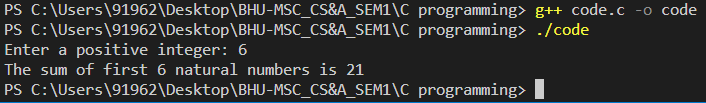
sum += i; // add each number from 1 to N to the sum variable

}

printf("The sum of first %d natural numbers is %d", n, sum); // display the sum

return 0;

}



Q4. Check Prime Number.

#include <stdio.h>

#include <stdbool.h>

bool isPrime(int n) {

if(n <= 1) {

return false; // 1 is not a prime number

}

for(int i = 2; i <= n/2; i++) {

if(n % i == 0) {

return false; // n is divisible by a number other than 1 and itself

}

}

return true; // n is a prime number

}

int main() {

int n;

printf("Enter a positive integer: ");

scanf("%d", &n); // take input from the user

if(isPrime(n)) {

printf("%d is a prime number", n); // display the message if n is prime

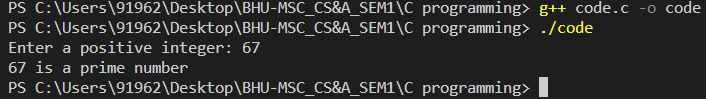
} else {

printf("%d is not a prime number", n); // display the message if n is not prime

}

return 0;

}



Q5. Check Armstrong Number.

#include <stdio.h>

#include <math.h>

int isArmstrong(int n) {

int originalNumber, remainder, result = 0, nDigits = 0;

originalNumber = n;

// count the number of digits in the given number

while(originalNumber != 0) {

originalNumber /= 10;

nDigits++;

}

originalNumber = n;

// find the sum of the cube of each digit

while(originalNumber != 0) {

remainder = originalNumber % 10;

result += pow(remainder, nDigits);

originalNumber /= 10;

}

// check if the sum is equal to the original number

if(result == n) {

return 1; // the number is an Armstrong number

} else {

return 0; // the number is not an Armstrong number

}

}

int main() {

int n;

printf("Enter a positive integer: ");

scanf("%d", &n); // take input from the user

if(isArmstrong(n)) {

printf("%d is an Armstrong number", n); // display the message if n is an Armstrong number

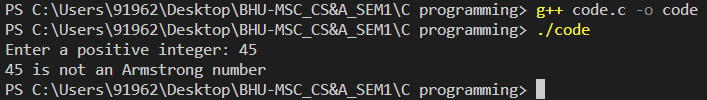
} else {

printf("%d is not an Armstrong number", n); // display the message if n is not an Armstrong number

}

return 0;

}



Q6. Print Fibonacci Series upto N.

#include <stdio.h>

void printFibonacci(int n) {

int first = 0, second = 1, next;

printf("Fibonacci series up to %d:\n", n);

printf("%d %d ", first, second);

next = first + second;

while(next <= n) {

printf("%d ", next);

first = second;

second = next;

next = first + second;

}

}

int main() {

int n;

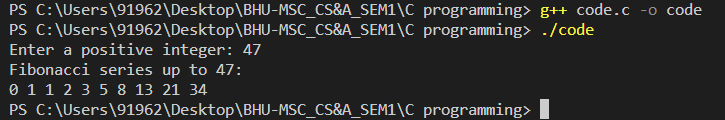
printf("Enter a positive integer: ");

scanf("%d", &n); // take input from the user

printFibonacci(n); // call the function to print the Fibonacci series up to n

return 0;

}



Q7. Check Perfect Number

#include <stdio.h>

int isPerfect(int n) {

int i, sum = 0;

// calculate the sum of proper divisors of n

for(i = 1; i <= n/2; i++) {

if(n % i == 0) {

sum += i;

}

}

// check if the sum of proper divisors is equal to n

if(sum == n) {

return 1; // the number is a Perfect number

} else {

return 0; // the number is not a Perfect number

}

}

int main() {

int n;

printf("Enter a positive integer: ");

scanf("%d", &n); // take input from the user

if(isPerfect(n)) {

printf("%d is a Perfect number", n); // display the message if n is a Perfect number

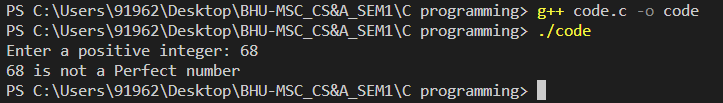
} else {

printf("%d is not a Perfect number", n); // display the message if n is not a Perfect number

}

return 0;

}



Q8. Swap two numbers without using a third variable.

#include <stdio.h>

int main() {

int a, b;

printf("Enter the first number: ");

scanf("%d", &a); // take input of first number

printf("Enter the second number: ");

scanf("%d", &b); // take input of second number

printf("Before swapping: a = %d, b = %d\n", a, b);

a = a + b;

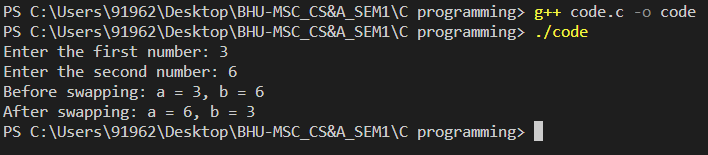
b = a - b;

a = a - b;

printf("After swapping: a = %d, b = %d\n", a, b);

return 0;

}



Q9. Count the digits of a number given input by a user.

#include <stdio.h>

int main() {

int n, count = 0;

printf("Enter a number: ");

scanf("%d", &n); // take input from the user

while(n != 0) {

count++; // increment the count of digits

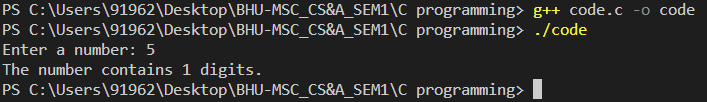
n /= 10; // remove the last digit

}

printf("The number contains %d digits.", count); // display the number of digits

return 0;

}



Q10. Calculate A^b without using any library function

#include <stdio.h>

int main() {

int a, b, result = 1;

printf("Enter the value of a: ");

scanf("%d", &a); // take input of a

printf("Enter the value of b: ");

scanf("%d", &b); // take input of b

while(b != 0) {

result \*= a; // multiply the result with a

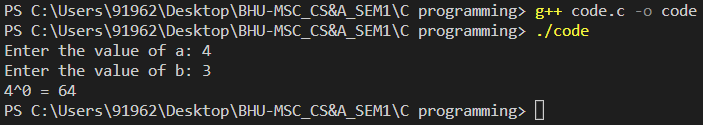
b--; // decrement the value of b

}

printf("%d^%d = %d", a, b, result); // display the result

return 0;

}



Q11. Reverse a number given input by a user.

#include <stdio.h>

int main() {

int n, reversed = 0;

printf("Enter a number: ");

scanf("%d", &n); // take input from the user

while(n != 0) {

reversed = reversed \* 10 + n % 10; // reverse the number

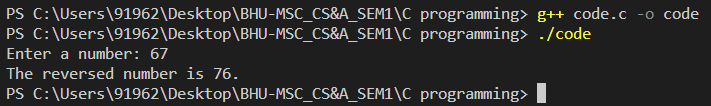
n /= 10; // remove the last digit

}

printf("The reversed number is %d.", reversed); // display the reversed number

return 0;

}



Q12. Find out the sum of all odd and even numbers present inside an array list of about 10 integer elements.

#include <stdio.h>

int main() {

int arr[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; // initialize the array

int i, sum\_even = 0, sum\_odd = 0;

for(i = 0; i < 10; i++) {

if(arr[i] % 2 == 0) {

sum\_even += arr[i]; // add even numbers to sum\_even

} else {

sum\_odd += arr[i]; // add odd numbers to sum\_odd

}

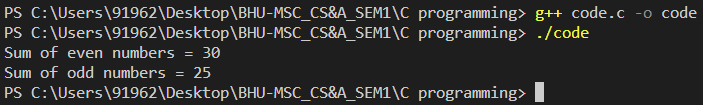
}

printf("Sum of even numbers = %d\n", sum\_even); // display the sum of even numbers

printf("Sum of odd numbers = %d\n", sum\_odd); // display the sum of odd numbers

return 0;

}



Q13. Calculate the average of elements of the array

#include <stdio.h>

int main() {

int arr[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; // initialize the array

int i, sum = 0;

float avg;

for(i = 0; i < 10; i++) {

sum += arr[i]; // add the elements of the array

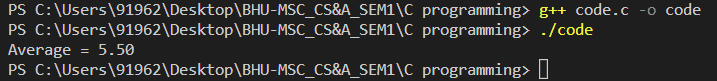
}

avg = (float)sum / 10; // calculate the average

printf("Average = %.2f\n", avg); // display the average with two decimal places

return 0;

}



Q14. Check whether a particular integer number is present in the list of 20 numbers in array. Also find the number of occurrences of such integer.

#include <stdio.h>

int main() {

int arr[20] = {3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3, 2, 3, 8, 4}; // initialize the array

int n, i, count = 0;

printf("Enter an integer: ");

scanf("%d", &n); // read the integer to be searched from user input

for(i = 0; i < 20; i++) {

if(arr[i] == n) {

count++; // increment count if the integer is found in the array

}

}

if(count > 0) {

printf("%d is present in the array %d times.\n", n, count); // display the number of occurrences

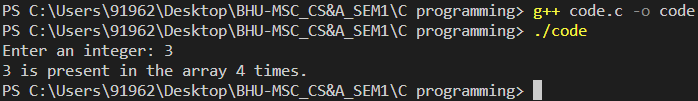
} else {

printf("%d is not present in the array.\n", n); // display a message if the integer is not found

}

return 0;

}



Q15. Add values of two arrays and update in the first one

#include <stdio.h>

int main() {

int arr1[5] = {1, 2, 3, 4, 5};

int arr2[5] = {10, 20, 30, 40, 50};

int i;

for(i = 0; i < 5; i++) {

arr1[i] = arr1[i] + arr2[i]; // add the values of the two arrays and update the values in arr1

}

printf("Updated values of arr1: ");

for(i = 0; i < 5; i++) {

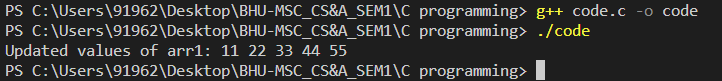
printf("%d ", arr1[i]); // display the updated values of arr1

}

printf("\n");

return 0;

}



Q16. Convert a decimal number to binary using array.

#include <stdio.h>

int main() {

int decimal, i, j;

int binary[32]; // to store the binary digits

printf("Enter a decimal number: ");

scanf("%d", &decimal);

i = 0;

while(decimal > 0) {

binary[i] = decimal % 2; // get the remainder when dividing by 2 (binary)

decimal = decimal / 2; // divide by 2

i++;

}

printf("Binary equivalent: ");

for(j = i - 1; j >= 0; j--) {

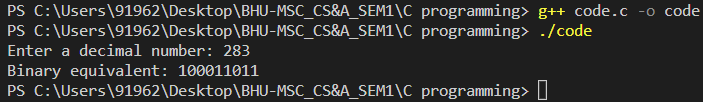
printf("%d", binary[j]); // display the binary digits in reverse order

}

printf("\n");

return 0;

}



Q17. Delete an element from a given array and display the updated array

#include <stdio.h>

int main() {

int arr[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

int n = 10; // size of the array

int i, j, k, x;

printf("Enter the element to delete: ");

scanf("%d", &x);

// find the location of the element to be deleted

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

if(arr[i] == x) {

break;

}

}

// shift the elements to the left starting from the location of the element to be deleted

for(j = i; j < n - 1; j++) {

arr[j] = arr[j + 1];

}

n--; // reduce the size of the array

printf("Updated array: ");

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

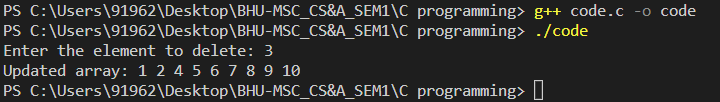
printf("%d ", arr[k]); // display the updated array

}

printf("\n");

return 0;

}



Q18. Read and Display few elements in a two dimensional array.

#include <stdio.h>

int main() {

int arr[2][2]; // a 3x3 array

int i, j;

// read elements of the array

for(i = 0; i < 2; i++) {

for(j = 0; j < 2; j++) {

printf("Enter element [%d][%d]: ", i, j);

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

}

}

// display first two rows and two columns of the array

printf("The array elements are:\n");

for(i = 0; i < 2; i++) {

for(j = 0; j < 2; j++) {

printf("%d ", arr[i][j]);

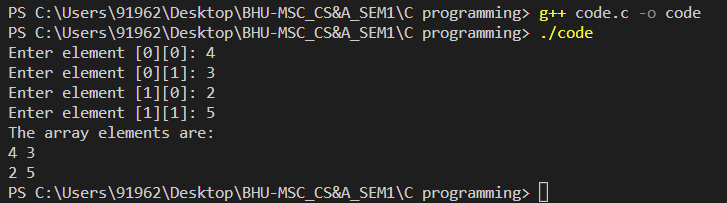
}

printf("\n");

}

return 0;

}



Q19. Perform Matrix addition

#include <stdio.h>

int main() {

int A[3][3], B[3][3], C[3][3];

int i, j;

// read elements of matrix A

printf("Enter elements of matrix A:\n");

for(i = 0; i < 3; i++) {

for(j = 0; j < 3; j++) {

printf("Enter element [%d][%d]: ", i, j);

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

}

}

// read elements of matrix B

printf("Enter elements of matrix B:\n");

for(i = 0; i < 3; i++) {

for(j = 0; j < 3; j++) {

printf("Enter element [%d][%d]: ", i, j);

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

}

}

// perform matrix addition

for(i = 0; i < 3; i++) {

for(j = 0; j < 3; j++) {

C[i][j] = A[i][j] + B[i][j];

}

}

// display the result matrix

printf("The result of matrix addition is:\n");

for(i = 0; i < 3; i++) {

for(j = 0; j < 3; j++) {

printf("%d ", C[i][j]);

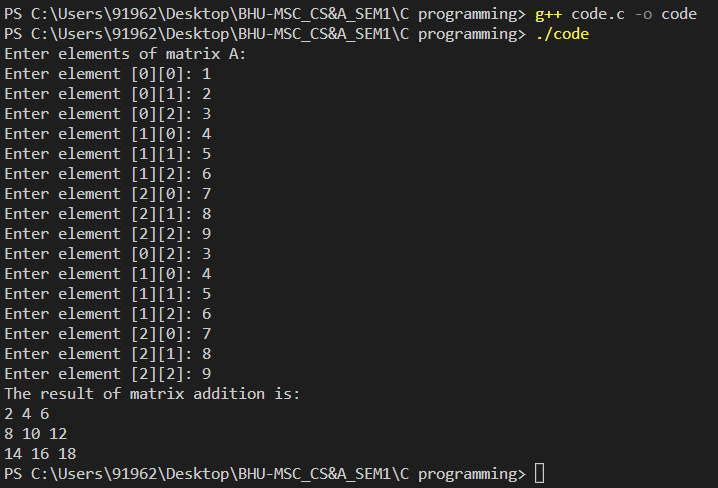
}

printf("\n");

}

return 0;

}



Q20. Write a program in C to find sum of values rows an columns of a Matrix separately.

#include <stdio.h>

int main() {

int rows, columns;

printf("Enter the number of rows in the matrix: ");

scanf("%d", &rows);

printf("Enter the number of columns in the matrix: ");

scanf("%d", &columns);

int matrix[rows][columns];

// Taking input for the matrix

printf("Enter the elements of the matrix:\n");

for (int i = 0; i < rows; i++) {

for (int j = 0; j < columns; j++) {

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

}

}

// Finding the sum of each row

for (int i = 0; i < rows; i++) {

int row\_sum = 0;

for (int j = 0; j < columns; j++) {

row\_sum += matrix[i][j];

}

printf("The sum of elements in row %d is %d\n", i+1, row\_sum);

}

// Finding the sum of each column

for (int i = 0; i < columns; i++) {

int column\_sum = 0;

for (int j = 0; j < rows; j++) {

column\_sum += matrix[j][i];

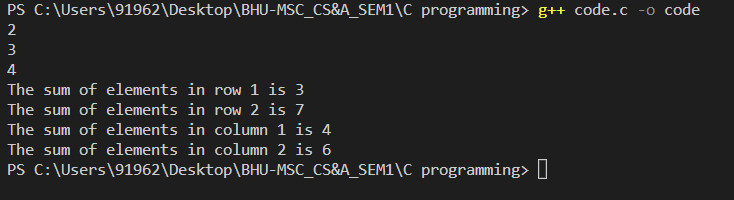
}

printf("The sum of elements in column %d is %d\n", i+1, column\_sum);

}

return 0;

}



Q21. Write a program in C to print or display the lower triangular of a given matrix.

#include <stdio.h>

int main() {

int rows, columns;

printf("Enter the number of rows and columns of the matrix: ");

scanf("%d %d", &rows, &columns);

int matrix[rows][columns];

printf("Enter the elements of the matrix:\n");

for (int i = 0; i < rows; i++) {

for (int j = 0; j < columns; j++) {

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

}

}

printf("The lower triangular matrix is:\n");

for (int i = 0; i < rows; i++) {

for (int j = 0; j < columns; j++) {

if (i >= j) {

printf("%d ", matrix[i][j]);

} else {

printf(" ");

}

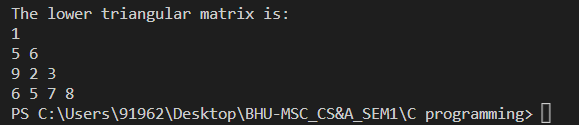
}

printf("\n");

}

return 0;

}



Q22. Read any String and display it.

#include <stdio.h>

int main() {

char str[100];

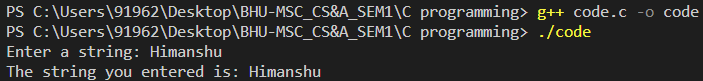
printf("Enter a string: ");

gets(str);

printf("The string you entered is: %s", str);

return 0;

}



Q23. Input a string and display the length of the string without using strlen() function

#include <stdio.h>

int main() {

char str[100];

int length = 0;

printf("Enter a string: ");

scanf("%[^\n]", str);

while (str[length] != '\0') {

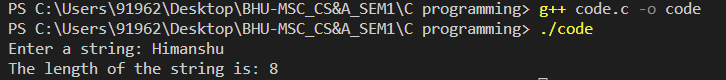
length++;

}

printf("The length of the string is: %d", length);

return 0;

}



Q24. Compare two strings without using strcmp() function

#include <stdio.h>

int main() {

char str1[100], str2[100];

int i = 0, flag = 0;

printf("Enter the first string: ");

scanf("%s", str1);

printf("Enter the second string: ");

scanf("%s", str2);

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

if (str1[i] != str2[i]) {

flag = 1;

break;

}

i++;

}

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

printf("The two strings are equal");

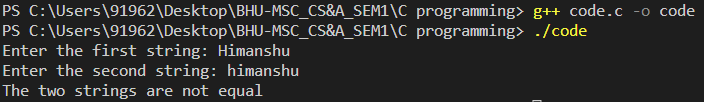
} else {

printf("The two strings are not equal");

}

return 0;

}



Q25. Copy one string into another without using strcpy() function

#include <stdio.h>

int main() {

char str1[100], str2[100];

int i = 0;

printf("Enter a string: ");

scanf("%s", str1);

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

str2[i] = str1[i];

i++;

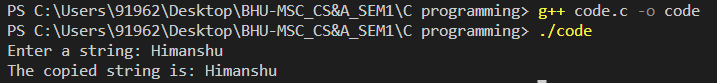
}

str2[i] = '\0';

printf("The copied string is: %s", str2);

return 0;

}



Q26. Concatenate one string into another without using strcat() function

#include <stdio.h>

int main() {

char str1[100], str2[100], result[200];

int i = 0, j = 0;

printf("Enter the first string: ");

scanf("%s", str1);

printf("Enter the second string: ");

scanf("%s", str2);

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

result[j] = str1[i];

i++;

j++;

}

i = 0;

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

result[j] = str2[i];

i++;

j++;

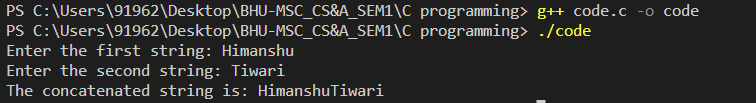
}

result[j] = '\0';

printf("The concatenated string is: %s", result);

return 0;

}



Q27. Check whether an input string is palindrome or not.

#include <stdio.h>

#include <string.h>

int main() {

char str[100];

int i, j, flag = 1;

printf("Enter a string: ");

scanf("%s", str);

j = strlen(str) - 1;

for (i = 0; i < j; i++, j--) {

if (str[i] != str[j]) {

flag = 0;

break;

}

}

if (flag == 1) {

printf("%s is a palindrome", str);

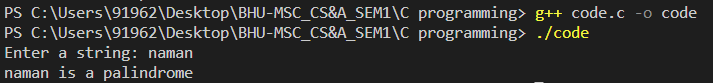
} else {

printf("%s is not a palindrome", str);

}

return 0;

}



Q28. Count and display the frequency of a character in the string

#include <stdio.h>

#include <string.h>

int main() {

char str[100], ch;

int i, freq = 0;

printf("Enter a string: ");

scanf("%s", str);

printf("Enter a character to find its frequency: ");

scanf(" %c", &ch);

for (i = 0; str[i] != '\0'; i++) {

if (str[i] == ch)

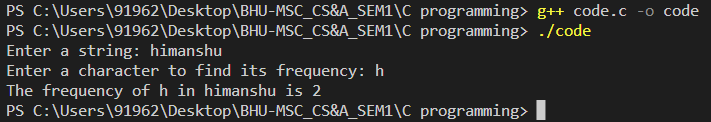
freq++;

}

printf("The frequency of %c in %s is %d", ch, str, freq);

return 0;

}



Q29. Count the number of vowels and consonants present in the string.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

char str[100];

int i, vowels = 0, consonants = 0;

printf("Enter a string: ");

scanf("%s", str);

for (i = 0; i < strlen(str); i++) {

if (isalpha(str[i])) {

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

vowels++;

} else {

consonants++;

}

}

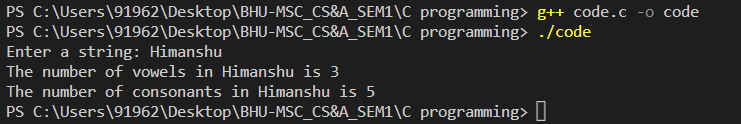
}

printf("The number of vowels in %s is %d\n", str, vowels);

printf("The number of consonants in %s is %d", str, consonants);

return 0;

}



Q30. Show the implementation of string library functions:   
a) strlen()   
b) strcat()   
c) strcpy()   
d) strcmp()   
e) strncat()  
f) strncmp()

#include <stdio.h>

#include <string.h>

int my\_strlen(char \*str) {

int len = 0;

while (\*str != '\0') {

len++;

str++;

}

return len;

}

char \*my\_strcat(char \*dest, char \*src) {

char \*ptr = dest + strlen(dest);

while (\*src != '\0') {

\*ptr++ = \*src++;

}

\*ptr = '\0';

return dest;

}

char \*my\_strcpy(char \*dest, char \*src) {

char \*ptr = dest;

while (\*src != '\0') {

\*ptr++ = \*src++;

}

\*ptr = '\0';

return dest;

}

int my\_strcmp(char \*str1, char \*str2) {

while (\*str1 != '\0' || \*str2 != '\0') {

if (\*str1 != \*str2) {

return \*str1 - \*str2;

}

str1++;

str2++;

}

return 0;

}

char \*my\_strncat(char \*dest, char \*src, int n) {

char \*ptr = dest + strlen(dest);

while (\*src != '\0' && n--) {

\*ptr++ = \*src++;

}

\*ptr = '\0';

return dest;

}

int my\_strncmp(char \*str1, char \*str2, int n) {

while ((\*str1 != '\0' || \*str2 != '\0') && n--) {

if (\*str1 != \*str2) {

return \*str1 - \*str2;

}

str1++;

str2++;

}

return 0;

}

int main() {

char str1[100], str2[100], str3[100];

int n;

printf("Enter a string: ");

scanf("%s", str1);

printf("Length of string: %d\n", my\_strlen(str1));

printf("Enter another string: ");

scanf("%s", str2);

printf("Concatenated string: %s\n", my\_strcat(str1, str2));

printf("Copy of string: %s\n", my\_strcpy(str3, str1));

printf("Enter a third string: ");

scanf("%s", str3);

printf("Comparison of strings: %d\n", my\_strcmp(str2, str3));

printf("Enter a fourth string: ");

scanf("%s", str3);

printf("Enter the number of characters to concatenate: ");

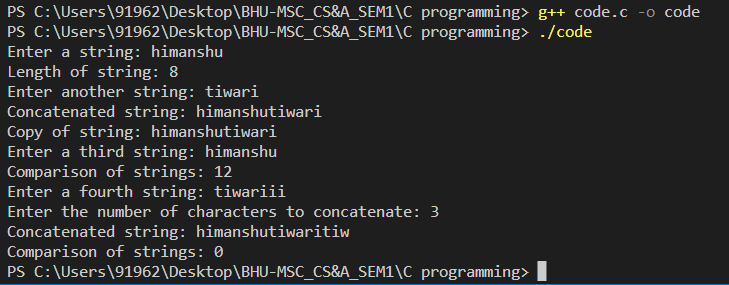
scanf("%d", &n);

printf("Concatenated string: %s\n", my\_strncat(str1, str3, n));

printf("Comparison of strings: %d\n", my\_strncmp(str2, str3, n));

return 0;

}



Q31. Program to create, initialize, assign and access a pointer variable

#include <stdio.h>

int main() {

int num = 10;

int \*ptr; // Pointer variable declaration

ptr = &num; // Pointer assignment

printf("Value of num: %d\n", num);

printf("Address of num: %p\n", &num);

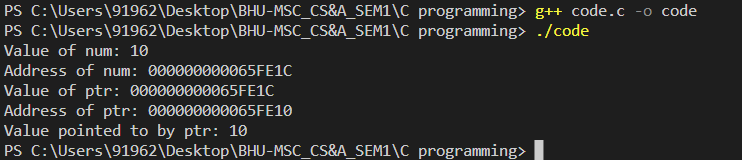
printf("Value of ptr: %p\n", ptr);

printf("Address of ptr: %p\n", &ptr);

printf("Value pointed to by ptr: %d\n", \*ptr);

return 0;

}



Q32. Program to swap two numbers using pointers.

#include <stdio.h>

void swap(int \*a, int \*b);

int main() {

int x = 10, y = 20;

printf("Before swapping: x = %d, y = %d\n", x, y);

// Call the swap function with the addresses of x and y

swap(&x, &y);

printf("After swapping: x = %d, y = %d\n", x, y);

return 0;

}

// Define the swap function that takes two integer pointers as arguments

void swap(int \*a, int \*b) {

int temp;

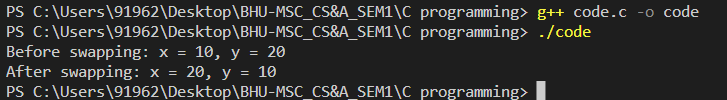
// Swap the values pointed to by the two pointers

temp = \*a;

\*a = \*b;

\*b = temp;

}



Q33. Program to print a string using pointer.

#include <stdio.h>

int main() {

char str[] = "Hello, World!";

char \*ptr = str;

// Loop through the string using the pointer

while (\*ptr != '\0') {

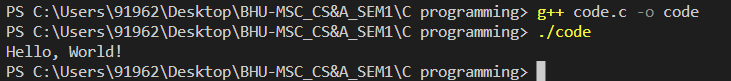
printf("%c", \*ptr);

ptr++;

}

return 0;

}



Q34.Program to count vowels and consonants in a string using pointer

#include <stdio.h>

#include <ctype.h>

int main() {

char str[100];

int vowels = 0, consonants = 0;

char \*ptr;

printf("Enter a string: ");

fgets(str, 100, stdin);

ptr = str;

while (\*ptr != '\0') {

if (isalpha(\*ptr)) {

if (tolower(\*ptr) == 'a' || tolower(\*ptr) == 'e' || tolower(\*ptr) == 'i' || tolower(\*ptr) == 'o' || tolower(\*ptr) == 'u') {

vowels++;

} else {

consonants++;

}

}

ptr++;

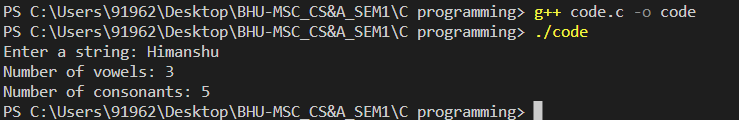
}

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

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

return 0;

}



Q35. Program to read array elements and print with addresses

#include <stdio.h>

int main() {

int arr[10];

int i;

int \*ptr;

// Read array elements from the user

printf("Enter 10 integer values:\n");

for (i = 0; i < 10; i++) {

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

}

ptr = arr;

// Print array elements with their addresses

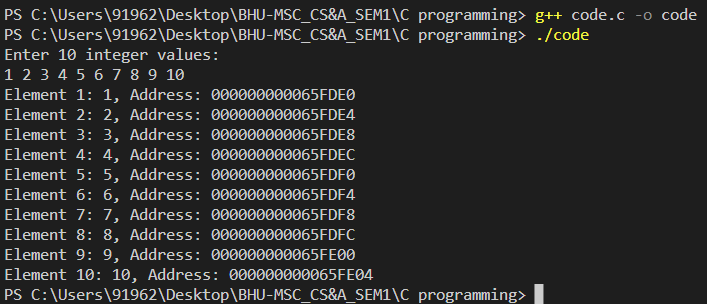
for (i = 0; i < 10; i++) {

printf("Element %d: %d, Address: %p\n", i+1, \*(ptr+i), ptr+i);

}

return 0;

}



Q36. Modify value stored in another variable using pointer in C.

#include <stdio.h>

int main() {

int a = 10;

int \*ptr;

printf("Value of a before modification: %d\n", a);

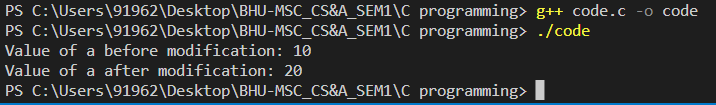
ptr = &a;

\*ptr = 20;

printf("Value of a after modification: %d\n", a);

return 0;

}



Q37.Calculate the length of the string using a pointer.

#include <stdio.h>

int main() {

char str[100];

char \*ptr;

int length = 0;

printf("Enter a string: ");

scanf("%s", str);

ptr = str;

// Use pointer arithmetic to traverse the string

while (\*ptr != '\0') {

length++;

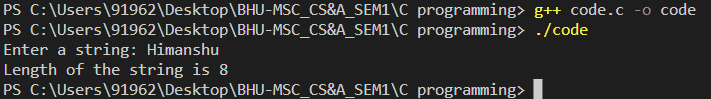
ptr++;

}

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

return 0;

}



Q38.Compute the sum of all elements in an array using pointers

#include <stdio.h>

int main() {

int arr[100];

int \*ptr;

int sum = 0;

int n, i;

printf("Enter the number of elements in the array: ");

scanf("%d", &n);

printf("Enter the elements of the array: ");

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

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

}

ptr = arr;

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

sum += \*ptr;

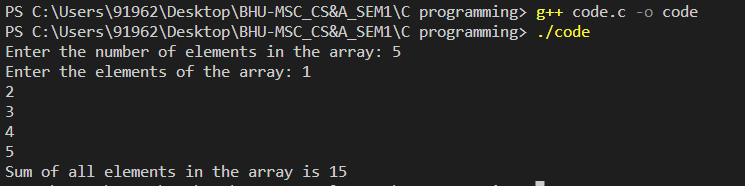
ptr++;

}

printf("Sum of all elements in the array is %d\n", sum);

return 0;

}



Q39. Show the difference between call by value and call by reference

#include <stdio.h>

// Function prototype for call by value

void call\_by\_value(int num);

// Function prototype for call by reference

void call\_by\_reference(int \*num);

int main() {

int num = 10;

// Call function by value

printf("Before call by value: num = %d\n", num);

call\_by\_value(num);

printf("After call by value: num = %d\n\n", num);

// Call function by reference

printf("Before call by reference: num = %d\n", num);

call\_by\_reference(&num);

printf("After call by reference: num = %d\n", num);

return 0;

}

// Function definition for call by value

void call\_by\_value(int num) {

num = 20;

printf("Inside call by value: num = %d\n", num);

}

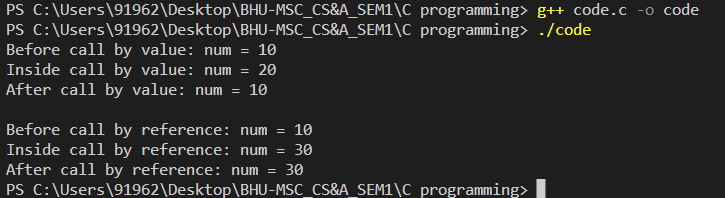
// Function definition for call by reference

void call\_by\_reference(int \*num) {

\*num = 30;

printf("Inside call by reference: num = %d\n", \*num);

}



Q40. Find the sum of all the elements in an array using Dynamic Memory Allocation.

#include <stdio.h>

#include <stdlib.h>

int main() {

int \*arr, n, i, sum = 0;

printf("Enter the number of elements in the array: ");

scanf("%d", &n);

// Allocate memory for the array dynamically

arr = (int\*)malloc(n \* sizeof(int));

if (arr == NULL) {

printf("Memory allocation failed!");

exit(0);

}

printf("Enter the elements of the array: ");

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

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

}

// Calculate the sum of all elements in the array

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

sum += arr[i];

}

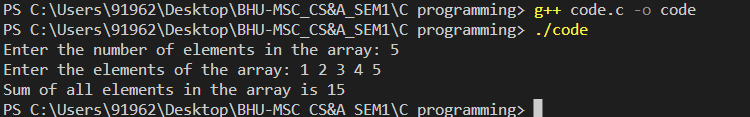
printf("Sum of all elements in the array is %d\n", sum);

// Free the dynamically allocated memory

free(arr);

return 0;

}



Q41. Convert a decimal number to octal and vice versa using a user defined function

#include <stdio.h>

#include <math.h>

// Function to convert decimal to octal

void dec\_to\_oct(int num) {

int octal[100], i = 0;

while (num != 0) {

octal[i] = num % 8;

num /= 8;

i++;

}

printf("Octal equivalent: ");

for (int j = i - 1; j >= 0; j--) {

printf("%d", octal[j]);

}

printf("\n");

}

// Function to convert octal to decimal

void oct\_to\_dec(int num) {

int decimal = 0, i = 0;

while (num != 0) {

decimal += (num % 10) \* pow(8, i);

num /= 10;

i++;

}

printf("Decimal equivalent: %d\n", decimal);

}

int main() {

int choice, num;

printf("Enter 1 to convert decimal to octal, or 2 to convert octal to decimal: ");

scanf("%d", &choice);

if (choice == 1) {

printf("Enter the decimal number: ");

scanf("%d", &num);

dec\_to\_oct(num);

} else if (choice == 2) {

printf("Enter the octal number: ");

scanf("%d", &num);

oct\_to\_dec(num);

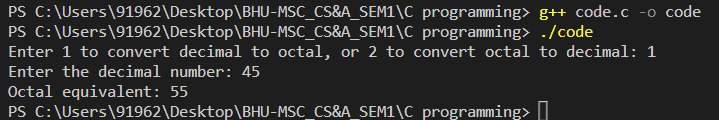
} else {

printf("Invalid choice!");

}

return 0;

}



Q42. Consider any 5 questions of your choice from assignments 1-5 and implement using user defined function

1. Program to count the number of words in a string:

#include <stdio.h>

#include <string.h>

int count\_words(char str[]) {

int count = 0, i;

for (i = 0; str[i] != '\0'; i++) {

if (str[i] == ' ' || str[i] == '\n' || str[i] == '\t') {

count++;

}

}

return count + 1;

}

int main() {

char str[100];

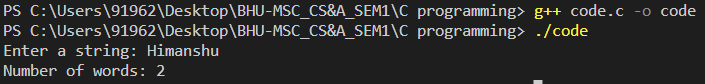
printf("Enter a string: ");

fgets(str, 100, stdin);

printf("Number of words: %d\n", count\_words(str));

return 0;

}



1. Program to find the maximum element in an array:

#include <stdio.h>

int max\_element(int arr[], int size) {

int max = arr[0];

for (int i = 1; i < size; i++) {

if (arr[i] > max) {

max = arr[i];

}

}

return max;

}

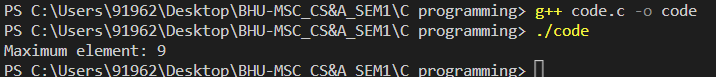
int main() {

int arr[5] = {5, 2, 9, 1, 7};

printf("Maximum element: %d\n", max\_element(arr, 5));

return 0;

}



1. Program to concatenate two strings:

#include <stdio.h>

#include <string.h>

void my\_strcat(char str1[], char str2[]) {

int i = strlen(str1), j = 0;

while (str2[j] != '\0') {

str1[i] = str2[j];

i++;

j++;

}

str1[i] = '\0';

}

int main() {

char str1[50] = "Hello, ";

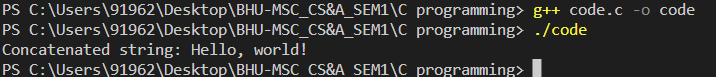
char str2[] = "world!";

my\_strcat(str1, str2);

printf("Concatenated string: %s\n", str1);

return 0;

}



1. Program to calculate the factorial of a number:

#include <stdio.h>

int factorial(int n) {

if (n == 0) {

return 1;

} else {

return n \* factorial(n - 1);

}

}

int main() {

int n;

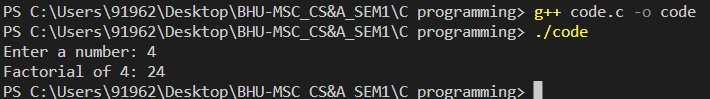
printf("Enter a number: ");

scanf("%d", &n);

printf("Factorial of %d: %d\n", n, factorial(n));

return 0;

}



1. Program to find the sum of all elements in an array:

#include <stdio.h>

int sum\_array(int arr[], int size) {

int sum = 0;

for (int i = 0; i < size; i++) {

sum += arr[i];

}

return sum;

}

int main() {

int arr[5] = {1, 2, 3, 4, 5};

printf("Sum of array: %d\n", sum\_array(arr, 5));

return 0;

}

