Assignment 1

Q1#include <stdio.h>

#include <math.h>

// Function to check if a number is an Armstrong number

int isArmstrong(int number) {

int originalNumber = number;

int sum = 0;

int digits = 0;

while (number != 0) {

digits++;

number /= 10;

}

number = originalNumber;

while (number != 0) {

int digit = number % 10;

sum += pow(digit, digits);

number /= 10;

}

return sum == originalNumber;

}

int main() {

int number;

// Input from the user

printf("Enter a number: ");

scanf("%d", &number);

// Check and display the result

if (isArmstrong(number)) {

printf("%d is an Armstrong number.\n", number);

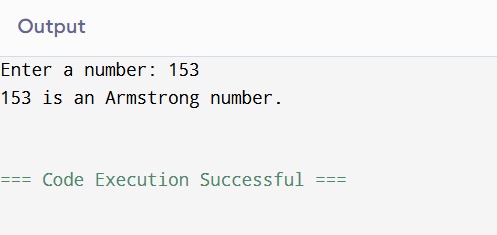
} else {

printf("%d is not an Armstrong number.\n", number);

}

return 0;

output:



Q2

#include <stdio.h>

int main() {

int a, b, t;

printf("Enter two integers: ");

scanf("%d %d", &a, &b);

int original\_a = a, original\_b = b;

while (b != 0) {

t = b;

b = a % b;

a = t;

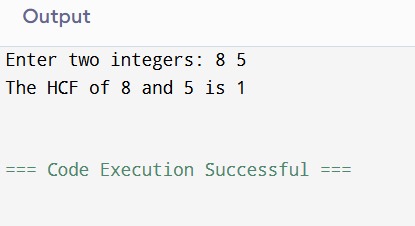
}

printf("The HCF of %d and %d is %d\n", original\_a, original\_b, a);

return 0;

}

Output:



Q3

#include<stdio.h>

int main() {

int num1,num2,sum;

printf("Enter any two numbers \n");

scanf("%d %d", &num1, &num2);

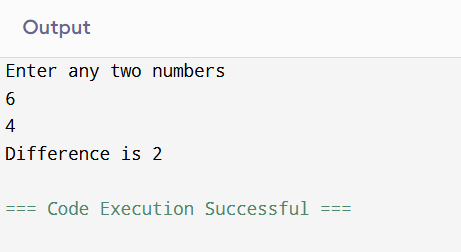
sum=num1+~num2+1;

printf("Difference is %d", sum);

return 0;

}

Output:



Q4:

#include <stdio.h>

int main() {

int num1, num2, temp;

printf("Enter two integers: ");

scanf("%d %d", &num1, &num2);

// Method 1: Using a temporary variable

temp = num1;

num1 = num2;

num2 = temp;

printf("Method 1: Using temporary variable => num1 = %d, num2 = %d\n", num1, num2);

printf("Enter two integers again: ");

scanf("%d %d", &num1, &num2);

// Method 2: Using arithmetic operations

num1 = num1 + num2;

num2 = num1 - num2;

num1 = num1 - num2;

printf("Method 2: Using arithmetic operations => num1 = %d, num2 = %d\n", num1, num2);

printf("Enter two integers again: ");

scanf("%d %d", &num1, &num2);

// Method 3: Using bitwise XOR

num1 = num1 ^ num2;

num2 = num1 ^ num2;

num1 = num1 ^ num2;

printf("Method 3: Using bitwise XOR => num1 = %d, num2 = %d\n", num1, num2);

printf("Enter two integers again: ");

scanf("%d %d", &num1, &num2);

// Method 4: Using pointers

int \*pnum1 = &num1, \*pnum2 = &num2;

\*pnum1 = \*pnum1 + \*pnum2;

\*pnum2 = \*pnum1 - \*pnum2;

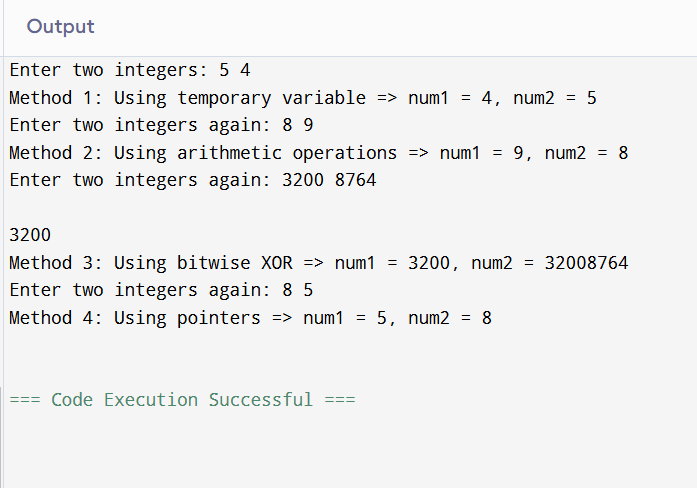
\*pnum1 = \*pnum1 - \*pnum2;

printf("Method 4: Using pointers => num1 = %d, num2 = %d\n", num1, num2);

return 0;

}

Output:



Q5:

#include <stdio.h>

int main() {

int num, sum = 0;

printf("Enter a number: ");

scanf("%d", &num);

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

if (num % i == 0) {

sum += i;

}

}

if (sum == num) {

printf("%d is a Perfect Number.\n", num);

} else {

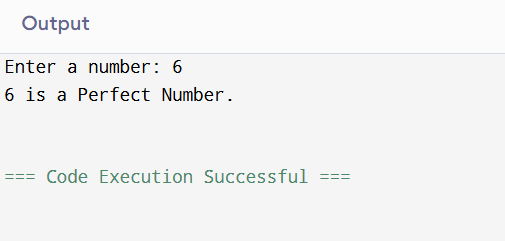
printf("%d is not a Perfect Number.\n", num);

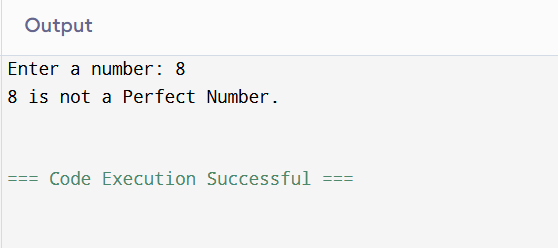
}

return 0;

}

Output:





Q6: #include<stdio.h>

int main() {

int x, y;

printf("Enter the x and y coordinate");

scanf("%d %d", &x, &y);

if (x == 0 && y == 0) {

printf("The point (0, 0) is at the origin.\n");

}

else if (x == 0) {

printf("The point lies on the Y-axis.\n");

}

else if (y == 0) {

printf("The point lies on the X-axis.\n");

}

else if (x > 0 && y > 0) {

printf("The point lies in Quadrant I.\n");

}

else if (x < 0 && y > 0) {

printf("The point lies in Quadrant II.\n");

}

else if (x < 0 && y < 0) {

printf("The point lies in Quadrant III.\n");

}

else if (x > 0 && y < 0) {

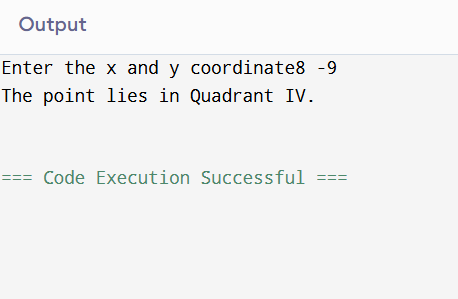
printf("The point lies in Quadrant IV.\n");

}

return 0;

}

Output:



Q7:

#include <stdio.h>

#include <math.h>

int binaryToDecimal(int binary) {

int decimal = 0, i = 0, remainder;

while (binary != 0) {

remainder = binary % 10;

decimal += remainder \* pow(2, i);

binary /= 10;

i++;

}

return decimal;

}

void decimalToBinary(int decimal) {

int binary[32];

int index = 0;

if (decimal == 0) {

printf("Binary: 0\n");

return;

}

while (decimal > 0) {

binary[index] = decimal % 2;

decimal = decimal / 2;

index++;

}

printf("Binary: ");

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

printf("%d", binary[i]);

}

printf("\n");

}

int main() {

int choice, number;

printf("Choose an option:\n");

printf("1. Binary to Decimal\n");

printf("2. Decimal to Binary\n");

printf("Enter your choice (1 or 2): ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter a binary number: ");

scanf("%d", &number);

printf("Decimal: %d\n", binaryToDecimal(number));

break;

case 2:

printf("Enter a decimal number: ");

scanf("%d", &number);

decimalToBinary(number);

break;

default:

printf("Invalid choice!\n");

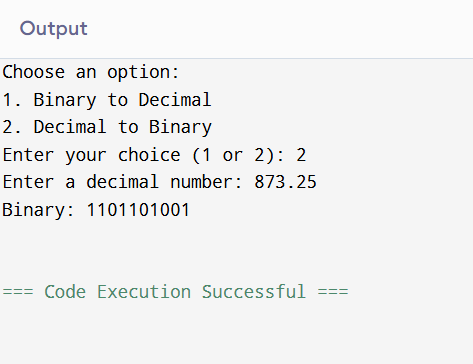
break;

}

return 0;

}

Output:



Q8: #include <stdio.h>

int main() {

int i, j;

for (i = 1; i <= 5; i++) {

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

if ((i + j) % 2 == 0) {

printf("1");

} else {

printf("0");

}

}

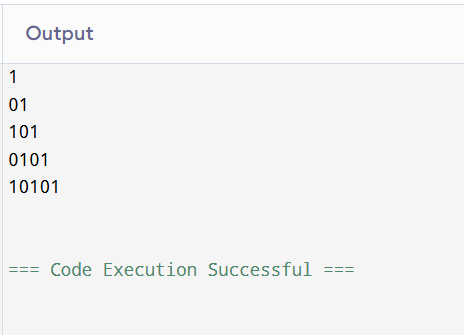
printf("\n");

}

return 0;

}

Output:



Q9:

#include <stdio.h>

int main() {

int i, j, k, n = 5;

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

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

printf("%d", j % 2);

}

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

printf(" ");

}

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

printf("%d", j % 2);

}

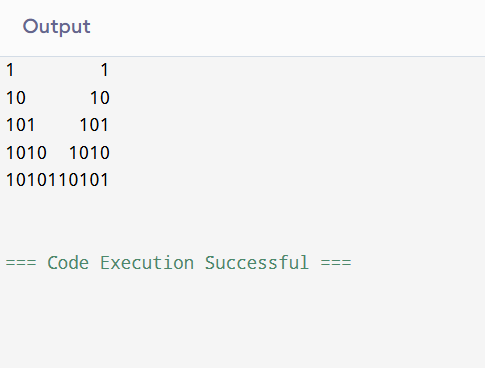
printf("\n");

}

return 0;

}

Output:



Q10:

#include <stdio.h>

int main() {

int rows, i, j;

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

scanf("%d", &rows);

int triangle[rows][rows];

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

triangle[i][0] = triangle[i][i] = 1;

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

triangle[i][j] = triangle[i - 1][j - 1] + triangle[i - 1][j];

}

}

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

for (j = 0; j < rows - i - 1; j++) {

printf(" ");

}

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

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

}

printf("\n");

}

return 0;

output:

