Q9. Accept an integer number and when the program is executed print the binary, octal and hexadecimal equivalent of the given number.

Sample Output:

Enter Number : 20

Given Number :20

Binary equivalent :10100

Octal equivalent :24

#include <stdio.h>

// Function to print the binary equivalent using bitwise operations

void printBinary(int n) {

printf("Binary equivalent: ");

// Iterate over each bit (assuming a 32-bit integer)

for (int i = 31; i >= 0; i--) {

int bit = (n >> i) & 1; // Right shift and mask with 1 to get the i-th bit

printf("%d", bit);

}

printf("\n");

}

// Recursive function to print the octal equivalent

void printOctal(int n) {

if (n == 0) {

return;

}

printOctal(n / 8); // Recursive call with n divided by 8

printf("%d", n % 8); // Print the remainder when divided by 8

}

// Recursive function to print the hexadecimal equivalent

void printHexadecimal(int n) {

if (n == 0) {

return;

}

printHexadecimal(n / 16); // Recursive call with n divided by 16

int remainder = n % 16;

// Print the corresponding hexadecimal digit

if (remainder < 10) {

printf("%d", remainder);

} else {

printf("%c", 'A' + (remainder - 10));

}

}

int main() {

int number;

// Accept an integer number from the user

printf("Enter Number: ");

scanf("%d", &number);

// Print the given number

printf("Given Number: %d\n", number);

// Print the binary equivalent

printBinary(number);

// Print the octal equivalent

printf("Octal equivalent: ");

if (number == 0) {

printf("0");

} else {

printOctal(number);

}

printf("\n");

// Print the hexadecimal equivalent

printf("Hexadecimal equivalent: ");

if (number == 0) {

printf("0");

} else {

printHexadecimal(number);

}

printf("\n");

return 0;

}