1. Write a C program to display the cube of the number upto N.

#include <stdio.h>

void displayCubes(int n) {

printf("Cubes of numbers up to %d are:\n", n);

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

printf("%d^3 = %d\n", i, i \* i \* i);

}

}

int main() {

int N;

// Input N from user

printf("Enter a value for N: ");

scanf("%d", &N);

// Display cubes

displayCubes(N);

return 0;

}

2. Write a C program to display the N terms of odd numbers and

their sum.

#include <stdio.h>

void displayOddNumbers(int n) {

int sum = 0;

printf("First %d odd numbers are:\n", n);

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

printf("%d ", i);

sum += i;

}

printf("\nSum of first %d odd numbers is: %d\n", n, sum);

}

int main() {

int N;

// Input N from user

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

scanf("%d", &N);

// Display odd numbers and their sum

displayOddNumbers(N);

return 0;

}

3. Write a C program to count number of digits in a number.

#include <stdio.h>

int countDigits(int number) {

int count = 0;

while (number != 0) {

number /= 10;

count++;

}

return count;

}

int main() {

int num;

// Input number from user

printf("Enter a number: ");

scanf("%d", &num);

// Count and print the number of digits

printf("Number of digits in %d is: %d\n", num, countDigits(num));

return 0;

}

4. Write a C program to check whether a number is Armstrong

number or not.

#include <stdio.h>

#include <math.h>

int isArmstrong(int number) {

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

originalNumber = number;

while (originalNumber != 0) {

originalNumber /= 10;

++n;

}

originalNumber = number;

while (originalNumber != 0) {

remainder = originalNumber % 10;

result += pow(remainder, n);

originalNumber /= 10;

}

return (result == number);

}

int main() {

int num;

// Input number from user

printf("Enter a number: ");

scanf("%d", &num);

// Check and print if the number is Armstrong

if (isArmstrong(num)) {

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

} else {

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

}

return 0;

}

5. Write a C program to check whether a number is Palindrome or

not.

#include <stdio.h>

int isPalindrome(int number) {

int originalNumber, reversedNumber = 0, remainder;

originalNumber = number;

while (originalNumber != 0) {

remainder = originalNumber % 10;

reversedNumber = reversedNumber \* 10 + remainder;

originalNumber /= 10;

}

return (number == reversedNumber);

}

int main() {

int num;

// Input number from user

printf("Enter a number: ");

scanf("%d", &num);

// Check and print if the number is Palindrome

if (isPalindrome(num)) {

printf("%d is a Palindrome number.\n", num);

} else {

printf("%d is not a Palindrome number.\n", num);

}

return 0;

}