# Day 11: Number-Based Programs – Part 1

## 1. Write a program to check whether a number is even or odd.

#include <stdio.h>  
int main() {  
 int num;  
 printf("Enter a number: ");  
 scanf("%d", &num);  
 if (num % 2 == 0)  
 printf("Even\n");  
 else  
 printf("Odd\n");  
 return 0;  
}

## 2. Write a program to check if a number is prime.

#include <stdio.h>  
int main() {  
 int num, i, flag = 0;  
 printf("Enter a number: ");  
 scanf("%d", &num);  
 for (i = 2; i <= num / 2; ++i) {  
 if (num % i == 0) {  
 flag = 1;  
 break;  
 }  
 }  
 if (num <= 1) flag = 1;  
 printf(flag ? "Not Prime\n" : "Prime\n");  
 return 0;  
}

## 3. Write a program to print all prime numbers in a given range.

#include <stdio.h>  
int isPrime(int n) {  
 if (n <= 1) return 0;  
 for (int i = 2; i <= n / 2; i++)  
 if (n % i == 0) return 0;  
 return 1;  
}  
int main() {  
 int start, end;  
 printf("Enter range: ");  
 scanf("%d %d", &start, &end);  
 for (int i = start; i <= end; i++)  
 if (isPrime(i)) printf("%d ", i);  
 return 0;  
}

## 4. Write a program to check whether a number is a palindrome.

#include <stdio.h>  
int main() {  
 int num, rev = 0, temp, rem;  
 printf("Enter a number: ");  
 scanf("%d", &num);  
 temp = num;  
 while (temp != 0) {  
 rem = temp % 10;  
 rev = rev \* 10 + rem;  
 temp /= 10;  
 }  
 printf(num == rev ? "Palindrome\n" : "Not Palindrome\n");  
 return 0;  
}

## 5. Write a program to check whether a number is an Armstrong number.

#include <stdio.h>  
#include <math.h>  
int main() {  
 int num, temp, rem, sum = 0, n = 0;  
 printf("Enter a number: ");  
 scanf("%d", &num);  
 temp = num;  
 while (temp != 0) {  
 temp /= 10;  
 n++;  
 }  
 temp = num;  
 while (temp != 0) {  
 rem = temp % 10;  
 sum += pow(rem, n);  
 temp /= 10;  
 }  
 printf(num == sum ? "Armstrong\n" : "Not Armstrong\n");  
 return 0;  
}

## 6. Write a program to check whether a number is a Strong number.

#include <stdio.h>  
int factorial(int n) {  
 int fact = 1;  
 for (int i = 1; i <= n; i++) fact \*= i;  
 return fact;  
}  
int main() {  
 int num, temp, rem, sum = 0;  
 printf("Enter a number: ");  
 scanf("%d", &num);  
 temp = num;  
 while (temp != 0) {  
 rem = temp % 10;  
 sum += factorial(rem);  
 temp /= 10;  
 }  
 printf(num == sum ? "Strong number\n" : "Not Strong number\n");  
 return 0;  
}

## 7. Write a program to check whether a number is a Perfect number.

#include <stdio.h>  
int main() {  
 int num, sum = 0;  
 printf("Enter a number: ");  
 scanf("%d", &num);  
 for (int i = 1; i < num; i++) {  
 if (num % i == 0) sum += i;  
 }  
 printf(num == sum ? "Perfect number\n" : "Not Perfect number\n");  
 return 0;  
}

## 8. Write a program to reverse a number.

#include <stdio.h>  
int main() {  
 int num, rev = 0, rem;  
 printf("Enter a number: ");  
 scanf("%d", &num);  
 while (num != 0) {  
 rem = num % 10;  
 rev = rev \* 10 + rem;  
 num /= 10;  
 }  
 printf("Reversed number: %d\n", rev);  
 return 0;  
}

## 9. Write a program to find the sum of digits of a number.

#include <stdio.h>  
int main() {  
 int num, sum = 0;  
 printf("Enter a number: ");  
 scanf("%d", &num);  
 while (num != 0) {  
 sum += num % 10;  
 num /= 10;  
 }  
 printf("Sum of digits: %d\n", sum);  
 return 0;  
}

## 10. Write a program to calculate the factorial of a number.

#include <stdio.h>  
int main() {  
 int num, fact = 1;  
 printf("Enter a number: ");  
 scanf("%d", &num);  
 for (int i = 1; i <= num; i++) {  
 fact \*= i;  
 }  
 printf("Factorial: %d\n", fact);  
 return 0;  
}