Q.1) Write a c function to accept the number and check that number is prime or not.

```
#include<stdio.h>
int isPrime(int number)
{
    if (number <= 1)
   {
        return 0; // Not prime
    }
    for (int i = 2; i < number; i++)
   {
        if (number % i == 0) {
            return 0; // Not prime
        }
    }
    return 1; // Prime
}

int main()
 {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);

    if (isPrime(num))
    {
        printf("%d is a prime number.\n", num);
    } else
  {
        printf("%d is not a prime number.\n", num);
    }

    return 0;
}
```

Q.2) Write a c function to accept the limit and display Fibonacci series upto the limit.

```
#include <stdio.h>
void fibonacci_series(int limit)
{
    int a = 0, b = 1, next;
   if (limit <= 0)
    {
        printf("Please enter a positive number.\n");
        return;
    }
    printf("Fibonacci series up to %d:\n", limit);
    while (a <= limit)
 {
        printf("%d ", a);
        next = a + b;
        a = b;
        b = next;
    }
    printf("\n");
}
int main()
 {
    int limit;
    printf("Enter the limit: ");
    scanf("%d", &limit);
    fibonacci_series(limit);
    return 0;
}
```

Q.3) Write a c function which takes hours, minutes and seconds as parameters and an integer ‘s’ and increments the time by ‘s’ seconds. Accept time and seconds in main and Display the new time in main.

```
#include <stdio.h>
void increment_time(int *hours, int *minutes, int *seconds, int s)
{
    int total_seconds;
    // Convert everything to total seconds
    total_seconds = (*hours * 3600) + (*minutes * 60) + *seconds + s;
    *hours = (total_seconds / 3600) % 24;
    *minutes = (total_seconds / 60) % 60;
    *seconds = total_seconds % 60;
}
int main()
{
    int hours, minutes, seconds, increment;
    printf("Enter time (hours minutes seconds): ");
    scanf("%d %d %d", &hours, &minutes, &seconds);
    printf("Enter seconds to increment: ");
    scanf("%d", &increment);
    increment_time(&hours, &minutes, &seconds, increment);
    printf("New time: %02d:%02d:%02d\n", hours, minutes, seconds);
   return 0;
}
```

Q.4) Write a c program to calculate the GCD of two numbers using pointer functions.

```
#include <stdio.h>
void gcd(int *a, int *b)
{
   while (*b != 0)
 {
        int temp = *b;
        *b = *a % *b;
        *a = temp;
    }
}
int main()
{
    int num1, num2;
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);
    if (num1 == num2)
```

```
  {
        printf("GCD is: %d\n", num1);
    }
 else
 {
        gcd(&num1, &num2);
        printf("GCD is: %d\n", num1);
    }
   return 0;
}
```

Q.5)Write a c function which display all armstrong numbers in between 1 to 500.

```
#include <stdio.h>
void displayArmstrongNumbers()
 {
    int num, sum, remainder;
  for (num = 1; num <= 500; num++)
  {
        sum = 0;
        int temp = num;
        while (temp != 0)
       {
            remainder = temp % 10;
            sum += remainder * remainder * remainder;
            temp /= 10;
        }
        if (sum == num)
        {
            printf("%d\n", num);
        }
   }
}
int main()
{
    displayArmstrongNumbers();
    return 0;
}
```

Q.6) Write a c function which accepts a number and three flags as parameters. If the number is even, set the first flag to 1. If the number is prime, set the second flag to1.If the number is divisible by 3or 7, set the third flag to 1. In main, accept an integer and use this function to check if it is even, prime and divisible by 3 or 7. (Hint : pass the addresses of flags to the function)

```
#include <stdio.h>
#include <math.h>
void checkNumber(int num, int *flag1, int *flag2, int *flag3)
  {
    if (num % 2 == 0)
   {
        *flag1 = 1;
    }
    int isPrime = 1;
    if (num < 2)
   {
        isPrime = 0;
    } else {
        for (int i = 2; i <= sqrt(num); i++)
     {
            if (num % i == 0) {
                isPrime = 0;
                break;
            }
        }
    }
    if (isPrime)
   {
        *flag2 = 1;
    }
    if (num % 3 == 0 || num % 7 == 0)
   {
        *flag3 = 1;
    }
}
int main()
 {
    int num;
    int flag1 = 0, flag2 = 0, flag3 = 0;
    printf("Enter an integer: ");
    scanf("%d", &num);
    checkNumber(num, &flag1, &flag2, &flag3);
    printf("Flag1 (Even): %d\n", flag1);
    printf("Flag2 (Prime): %d\n", flag2);
    printf("Flag3 (Divisible by 3 or 7): %d\n", flag3);
    return 0;
}
```

Q.7)Write a c function which takes distance in kilometer, centimeter and millimeter as parameters as an integer d and increments the distance by d millimeters. Accept distance and d in main and Display the new distance in main using the above function.

```
#include <stdio.h>
void incrementDistance(int *km, int *cm, int *mm, int d)
 {
    *mm += d;
    if (*mm >= 10)
   {
        *cm += *mm / 10;
        *mm = *mm % 10;
    }
    if (*cm >= 100)
   {
        *km += *cm / 100;
        *cm = *cm % 100;
    }
}
int main()
{
    int km, cm, mm, d;
    printf("Enter the distance in km, cm, mm: ");
    scanf("%d %d %d", &km, &cm, &mm);
    printf("Enter the increment in millimeters: ");
    scanf("%d", &d);
    incrementDistance(&km, &cm, &mm, d);
    printf("New distance: %d km %d cm %d mm\n", km, cm, mm);
    return 0;
}
```