1.1 Write a program to assist in the design of a hydroelectric dam. Prompt the user for the height
of the dam and for the number of cubic meters of water that are projected to flow from the top
to the bottom of the dam each second. Predict how many megawatts (1MW = 106W) of power
will be produced if 90% of the work done on the water by gravity is converted to electrical
energy. Note that the mass of one cubic meter of water is 1000 kg. Use 9.80 meters/second2
as the gravitational constant g . Be sure to use meaningful names for both the gravitational
constant and the 90% efficiency constant. For one run, use a height of 170 m and flow of
1.30 × 103 m3/s. The relevant formula (w = work, m = mass, g = gravity, h = height) is: w
= mgh.

```
#include <stdio.h>

int main() {
    double height, flow, efficiency = 0.9, g = 9.80, power;
    printf("Enter height of the dam (m): ");
    scanf("%lf", &height);
    printf("Enter water flow rate (m^3/s): ");
    scanf("%lf", &flow);

    double mass = flow * 1000; // Mass of water (kg/s)
    power = efficiency * mass * g * height / 1e6; // Convert W to MW
    printf("The power produced is %.2f MW\n", power);

    return 0;
}
```

1.2 Metro City Planners proposes that a community conserve its water supply by replacing all the
community’ s toilets with low-flush models that use only 2 liters per flush. Assume that there
is about 1 toilet for every 3 persons, that existing toilets use an average of 15 liters per flush,
that a toilet is flushed on average 14 times per day, and that the cost to install each new toilet
is $150. Write a program that would estimate the magnitude (liters/day) of the water saved
and the total cost to install new toilets based on the community’ s population.

```
#include <stdio.h>

int main() {
    int population;
```

```
    printf("Enter the population: ");
    scanf("%d", &population);

    int toilets = population / 3;
    double flushes = 14.0, old_flush = 15.0, new_flush = 2.0,
cost_per_toilet = 150.0;
    double water_saved = toilets * flushes * (old_flush - new_flush);
    double total_cost = toilets * cost_per_toilet;

    printf("Water saved: %.2f liters/day\n", water_saved);
    printf("Total cost for new toilets: $%.2f\n", total_cost);

    return 0;
}
```

1.3 Write a program that calculates the acceleration (m/s2) of a jet fighter launched from an
aircraft-carrier catapult, given the jet' s takeoff speed in km/hr and the distance (meters) over
which the catapult accelerates the jet from rest to takeoff. Assume constant acceleration.
Also calculate the time (seconds) for the fighter to be accelerated to takeoff speed. When you
prompt the user, be sure to indicate the units for each input. For one run, use a takeoff speed
of 278 km/hr and a distance of 94 meters. Relevant formulas (v = velocity, a = acceleration,
t = time, s = distance)
v = at
s =1/2 at2

```
#include <stdio.h>

int main() {
    double speed_kmh, distance, speed_ms, acceleration, time;
    printf("Enter takeoff speed (km/hr): ");
    scanf("%lf", &speed_kmh);
    printf("Enter acceleration distance (m): ");
    scanf("%lf", &distance);

    speed_ms = speed_kmh / 3.6; // Convert km/hr to m/s
    acceleration = speed_ms * speed_ms / (2 * distance);
    time = speed_ms / acceleration;

    printf("Acceleration: %.2f m/s^2\n", acceleration);
    printf("Time: %.2f seconds\n", time);

    return 0;
}
```

1.4 You have saved $500 to use as a down payment on a car. Before beginning your car shopping,

you decide to write a program to help you figure out what your monthly payment will be, given
the car’ s purchase price, the monthly interest rate, and the time period over which you will
pay back the loan. The formula for calculating your payment is
payment =
iP
1 − (1 + i)−n
where P = principal (the amount you borrow)
i = monthly interest rate
 112 of the annual rate)
n = total number of payments
Your program should prompt the user for the purchase price, the down payment, the annual
interest rate and the total number of payments (usually 36, 48, or 60). It should then display
the amount borrowed and the monthly payment including a dollar sign and two decimal places.

```
#include <stdio.h>
#include <math.h>

int main() {
    double price, down_payment, annual_rate, principal, monthly_rate,
payment;
    int num_payments;
    printf("Enter purchase price: ");
    scanf("%lf", &price);
    printf("Enter down payment: ");
    scanf("%lf", &down_payment);
    printf("Enter annual interest rate (%%): ");
    scanf("%lf", &annual_rate);
    printf("Enter number of payments: ");
    scanf("%d", &num_payments);

    principal = price - down_payment;
    monthly_rate = annual_rate / 12 / 100;
    payment = (monthly_rate * principal) / (1 - pow(1 + monthly_rate,
-num_payments));

    printf("Amount borrowed: $%.2f\n", principal);
    printf("Monthly payment: $%.2f\n", payment);

    return 0;
}
```

1.5 A cyclist coasting on a level road slows from a speed of 10 mi/hr to 2.5 mi/hr in one minute.
Write a computer program that calculates the cyclist’ s constant rate of acceleration and determines

how long the cyclist will take to come to rest, given an initial speed of 10 mi/hr.

```
#include <stdio.h>

void display_instructions() {
    printf("This program calculates acceleration and time to stop for a cyclist.\n");
}

int main() {
    double v_initial = 10.0, v_final = 2.5, time_to_slow = 60.0, acceleration, time_to_stop;
    display_instructions();

    acceleration = (v_final - v_initial) / time_to_slow; // m/s^2
    time_to_stop = -v_initial / acceleration;

    printf("Acceleration: %.2f m/s^2\n", acceleration);
    printf("Time to stop: %.2f seconds\n", time_to_stop);

    return 0;
}
```

1.6 Write a program to take a depth (in kilometers) inside the earth as input data; compute and display the temperature at this depth in degrees Celsius and degrees Fahrenheit. The relevant formulas are

Celsius = 10 (depth) + 20 (Celsius temperature at depth in km)

Fahrenheit = 1.8 (Celsius) + 32

Include two functions in your program. Function celsius at depth should compute and return the Celsius temperature at a depth measured in kilometers. Function fahrenheit should convert a Celsius temperature to Fahrenheit.

```
#include <stdio.h>

double celsius_at_depth(double depth) {
    return 10 * depth + 20;
}

double fahrenheit(double celsius) {
    return 1.8 * celsius + 32;
}

int main() {
    double depth, temp_c, temp_f;
    printf("Enter depth inside Earth (km): ");
    scanf("%lf", &depth);
```

```
    temp_c = celsius_at_depth(depth);
    temp_f = fahrenheit(temp_c);

    printf("Temperature at %.2f km depth:\n", depth);
    printf("%.2f °C\n", temp_c);
    printf("%.2f °F\n", temp_f);

    return 0;
}
```