print-letusc.txt

```
// This file contains all C source code files from letusc/ and tuition-c/ directories
// Generated by GitHub Copilot


// File: letusc/luc001.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc001.c

/* Temperature of a city in fahrenheit degrees is input through
the keyboard. WAP to convert this temperature into Centigrade
degrees. */
/* Author - Amit Dutta, Date - 13 sep, 2025 */
/* Chapter 1; Page 19; Question NO.: F(a) */

#include<stdio.h>
int main() {
    float f, c;
    printf("Enter the temperature of city in Fahrenheit : ");
    scanf("%f", &f);
    // formula (F - 32) 5/9
    c = ((f - 32) * 5) / 9;
    printf("\nTemperature in centigrade : %f", c);
    return 0;
}

/* --- End of letusc/luc001.c --- */

// File: letusc/luc002.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc002.c

/* The length and breadth of a rectangle and radius of a circle
are input through the keyboard. Write a program to calculate the
area and perimeter of the rectangle, and the area and circumference
of the circle. */
/* Author - Amit Dutta, Date - 16th SEP, 2025 */
/* Let Us C; Page - 19; Chap- 1; QNo.: F(b) */

#include <stdio.h>
#include <math.h>
int main()
{
    double len, bre, r, area_r, per, area_c, cir;
    printf("Enter the length and breadth of the rectangle : ");
    scanf("%lf %lf", &len, &bre);
    printf("Enter the Radius of the circle : ");
    scanf("%lf", &r);
    area_r = len * bre;
    per = 2 * (len + bre);
    area_c = M_PI * r * r;
    cir = 2 * M_PI * r;
    printf("\nArea of Rectangle       : %lf"
           "\nPerimeter of Rectangle  : %lf"
           "\nArea of Circle          : %lf"
           "\nCircumference of Circle : %lf",
           area_r, per, area_c, cir);
    return 0;
}

/* --- End of letusc/luc002.c --- */

// File: letusc/luc003.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc003.c

/* Paper of size AO has dimensions 1189 mm x 841 mm.
Each subsequent size A(n) is defined as A(n-1) cut in
half, parallel to its shorter sides. Thus, paper of
size A1 would have dimensions 841 mm x 594 mm. Write
a program to calculate and print paper sizes A0,ï¿½
A1,ï¿½A2,ï¿½ ... ï¿½A8. */
/* Author - Amit Dutta, Date - 16th SEP, 2025 */
/* Let Us C; Page - 19; Chap- 1; QNo.: F(c) */

#include<stdio.h>
#include<math.h>
int main() {
    double s_long = 1189.0, s_short = 841.0, temp;
    int i;
    printf("A0 Dimension : %g mm x %g mm", floor(s_long), floor(s_short));
    for (i = 1; i <= 8; i++) {
        temp = s_long;
```

```
        s_long = s_short;
        s_short = temp / 2;
        printf("\nA%d Dimension : %g mm x %g mm", i, floor(s_long), floor(s_short));
    }
    return 0;
}

/* --- End of letusc/luc003.c --- */

// File: letusc/luc004.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc004.c

/* If a five digit number is input through the keyboard,
write a program to calculate the sum of it's digit.
(Hint : Use the modulus operator %) */
/* Author - Amit Dutta, Date - 28th SEP, 2025 */
/* Let Us C; Page - 37; Chap- 2; QNo.: G(a) */

#include <stdio.h>
int main()
{
    int inp, result = 0, i, temp;
    printf("Enter a five digit number : ");
    scanf("%d", &inp);
    if (inp < 10000 || inp > 99999)
    {
        printf("\nPlease enter a valid five digit number.");
        return 1;
    }
    temp = inp;
    for (i = 1; i <= 5; i++)
    {
        result = result + (inp % 10);
        inp = inp / 10;
    }
    printf("\nSum of the digit '%d' is : %d", temp, result);
    return 0;
}
/* --- End of letusc/luc004.c --- */

// File: letusc/luc005.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc005.c

/* Write a program to recive Cartesian
co-ordinates (x, y) of a point and convert
them into Polar co-ordinates (r, phi)
    Hint : r = sqrt (x^2 + y^2) and phi = tan^-1 (y/x) */
/* Author - Amit Dutta, Date - 28th SEP, 2025 */
/* Let Us C; Page - 37; Chap- 2; QNo.: G(b) */

#include <stdio.h>
#include <math.h>
int main()
{
    double x, y, r, phi;
    printf("Enter the Cartesian Co-Ordinates in this format 'x, y' : ");
    scanf("%lf, %lf", &x, &y);
    r = sqrt(pow(x, 2) + pow(y, 2));
    phi = atan2(y, x);
    printf("\nPolar Co-Ordinates are : (%g, %g Rad)", r, phi);
    return 0;
}
/* --- End of letusc/luc005.c --- */

// File: letusc/luc006.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc006.c

/* Write a program to receive values of latitude (L1, L2) and longitude
(G1, G2), in degrees, of two places on the earth and output the distance
(D) between them in nautical miles. The formula for distance in nautical
miles is :
    D = 3963 cos^-1(sin L1 sin L2 + cos L1 cos L2 * cos(G2 - G1))
*/
/* Author - Amit Dutta, Date - 29th SEP, 2025 */
/* Let Us C, Chap - 2, Page - 37, Qn no.: G(c) */

#include <stdio.h>
#include <math.h>
int main()
{
    double l1, l2, g1, g2, d;
    printf("Enter the Latitude in 'L1, L2' format : ");
```

```
    scanf("%lf, %lf", &l1, &l2);
    printf("Enter the Longitude in 'G1, G2' format : ");
    scanf("%lf, %lf", &g1, &g2);
    // Converting degree to radian because function from math.h use radian not degree
    l1 = l1 * (M_PI / 180);
    l2 = l2 * (M_PI / 180);
    g1 = g1 * (M_PI / 180);
    g2 = g2 * (M_PI / 180);
    d = 3963 * acos(sin(l1) * sin(l2) + cos(l1) * cos(l2) * cos(g2 - g1));
    printf("Distance in nautical miles : %g", d);
    return 0;
}
/* --- End of letusc/luc006.c --- */

// File: letusc/luc007.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc007.c

/* Wind-chill factor is the felt air temperature on exposed skin due to wind.
The wind-chill temperature is always lower than the air temperature, and is
calculated as per the following formula.
    wcf = 35.74 + 0.6215t + (0.4275t - 35.75) * v^0.16
Where t is temperature and v is wind velocity. Write a program to receive
values of t and v and calcualate wind-chill factor (wcf). */
/* Author - Amit Dutta, Date - 30th SEP, 2025 */
/* Let Us C, Chap - 2, Page - 37, Qn No.: G(d) */

#include <stdio.h>
#include <math.h>
int main()
{
    double t, v, wcf;
    printf("Enter the temperature and velociy of the wind : ");
    scanf("%lf %lf", &t, &v);
    wcf = 35.74 + (0.6215 * t) + (((0.4275 * t) - 35.75) * pow(v, 0.16));
    printf("\nWind-chill factor (wcf) : %g", wcf);
    return 0;
}
/* --- End of letusc/luc007.c --- */

// File: letusc/luc008.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc008.c

/* If value of an angle is input through the keyboard,
write a program to print all its trigonometric ratios. */
/* Author - Amit Dutta, Date - 30th SEP, 2025 */
/* Let Us C, Chap - 2, Page - 37, Qn No.: G(e) */

#include <stdio.h>
#include <math.h>
int main()
{
    double inp, rsin, rcos, rtan, rcosec, rsec, rcot;
    printf("Enter the Angle in degree : ");
    scanf("%lf", &inp);
    inp = inp * (M_PI / 180); //changing degree to radian
    rsin = sin(inp);
    rcos = cos(inp);
    rtan = tan(inp);
    rcosec = 1 / rsin;
    rsec = 1 / rcos;
    rcot = 1 / rtan;
    printf("\nTrigonometric ratios :-"
           "\nsin   = %g "
           "\ncos   = %g"
           "\ntan   = %g"
           "\ncosec = %g"
           "\nsec   = %g"
           "\ncot   = %g",
           rsin, rcos, rtan, rcosec, rsec, rcot);
    return 0;
}
/* --- End of letusc/luc008.c --- */

// File: letusc/luc009.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc009.c

/* Two numbers are input through the keyboard into two locations
C and D. Write a program to interchange the contents of C and D. */
/* Author - Amit Dutta, Date - 30th SEP, 2025 */
/* Let Us C, Chap - 2, Page - 37, Qn No.: G(d) */

#include <stdio.h>
```

```
int main()
{
    double a, b;
    printf("Enter the two number A and B : ");
    scanf("%lf %lf", &a, &b);
    printf("\nBefore Interchange : ");
    printf("\nA = %g (Memory location = %p)", a, &a);
    printf("\nB = %g (Memory location = %p)", b, &b);
    a = a + b;
    b = a - b;
    a = a - b;
    printf("\nAfter Interchange :");
    printf("\nA = %g (Memory location = %p)", a, &a);
    printf("\nB = %g (Memory location = %p)", b, &b);
    return 0;
}
/* --- End of letusc/luc009.c --- */

// File: letusc/luc010.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc010.c

/* A five-digit number is entered through the keyboard. Write a program
to obtain the reversed number and to etermine whether the original and reversed
numbers are equal or not. */
/* Author - Amit Dutta, Date - 1st OCT, 2025 */
/* Let Us C, Chap - 3, Page - 53, Qn No.: f(a) */

#include <stdio.h>
int main()
{
    int num, rev, temp, i;
    printf("Please enter the number : ");
    scanf("%d", &num);
    if (num < 10000 || num > 99999)
    {
        printf("\nPlease enter a five digit number.");
        return 1;
    }
    temp = num;
    for (i = 1; i <= 5; i++)
    {
        rev = (rev * 10) + num % 10;
        num = num / 10;
    }
    printf("\nReverse of the Input number : %d", rev);
    if (rev == temp)
        printf("\nOriginal and reversed numbers are equal.");
    else
        printf("\nOrigianl and reversed numbers are not equal.");
    return 0;
}
/* --- End of letusc/luc010.c --- */

// File: letusc/luc011.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc011.c

/* If ages of Ram, Shyam and Ajay are input through the keyboard,
write a program to determine the youngest of the three. */
/* Author - Amit Dutta, Date - 1st OCT, 2025 */
/* Let Us C, Chap- 3, Page - 53, Qn No.: f(b) */

#include <stdio.h>
int main()
{
    int ram, shyam, ajay;
    printf("Please enter the age of Ram, Shyam and Ajay : ");
    scanf("%d %d %d", &ram, &shyam, &ajay);
    if (ram == shyam || ram == ajay || shyam == ajay)
        printf("\nThree must have different age.");
    if (ram < shyam && ram < ajay)
        printf("\nRam is the youngest. Age : %d", ram);
    if (shyam < ram && shyam < ajay)
        printf("\nShyam is the youngest. Age : %d", shyam);
    if (ajay < ram && ajay < shyam)
        printf("\nAjay is the youngest. Age : %d", ajay);
    return 0;
}
/* --- End of letusc/luc011.c --- */

// File: letusc/luc012.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc012.c
```

```c
/* Write a program to check whether a triangle is valid or not,
if three angles of the triangle are entered through the keyboard.
A triangle is valid if the sum of all the three angles is equal to 180 degrees. */
/* Author - Amit Dutta, Date - 1st OCT, 2025 */
/* Let Us C, Chap- 3, Page - 53, Qn No.: f(c) */

#include <stdio.h>
int main()
{
    double angle1, angle2, angle3, sum;
    printf("Enter the value of the three angle of the triangle : ");
    scanf("%lf %lf %lf", &angle1, &angle2, &angle3);
    sum = angle1 + angle2 + angle3;
    if (sum == 180.0)
        printf("\nThis Triangle is a valid one.");
    else
        printf("\nThis Triangle is not valid. Sum of the angles : %g", sum);
    return 0;
}
/* --- End of letusc/luc012.c --- */

// File: letusc/luc013.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc013.c

/* Write a program to find the absolute value
of a number entered through the keyboard. */
/* Author - Amit Dutta, Date - 1st OCT, 2025 */
/* Let Us C, Chap- 3, Page - 53, Qn No.: f(d) */

#include <stdio.h>
#include <math.h>
int main()
{
    double num;
    printf("Enter the number : ");
    scanf("%lf", &num);
    num = abs(num);
    printf("\nAbsolute value of the input is : %g", num);
    return 0;
}
/* --- End of letusc/luc013.c --- */

// File: letusc/luc014.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc014.c

/* Given the length and breadth of a rectangle, write a program to find
whether the area of the rectangle is greater than it's perimeter.
For example, the area of the rectangle with length = 5 and breadth = 4
is greater than its perimeter. */
/* Author - Amit Dutta, Date - 1st OCT, 2025 */
/* Let Us C, Chap- 3, Page - 53, Qn No.: f(e) */

#include <stdio.h>
int main()
{
    double len, bre, area, peri;
    printf("Enter the length and breadth of the rectangle : ");
    scanf("%lf %lf", &len, &bre);
    area = len * bre;
    peri = 2 * (len + bre);
    if (area > peri)
        printf("\nThe area of the rectangle is greater than it's perimeter.");
    else if (area < peri)
        printf("\nThe area of the rectangle is lower than it's perimeter.");
    else
        printf("\nThe area of the rectangle is same as it's perimeter.");
    return 0;
}
/* --- End of letusc/luc014.c --- */

// File: letusc/luc015.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc015.c

/* Given three points (x1, y1), (x2, y2), and (x3, y3),
write a program to check if the three poins fall on one straight line. */
/* Author - Amit Dutta, Date - 1st OCT, 2025 */
/* Let Us C, Chap- 3, Page - 53, Qn No.: f(f) */

#include <stdio.h>
#include <math.h>
#define EPSILON 0.0001
// Define a small tolerance value (EPSILON) for safe floating-point comparison
```

```
// This is critical because of minor rounding errors in computer arithmetic.
int main()
{
    double x1, x2, x3, y1, y2, y3, area;
    printf("Enter the point A(x1, y1) : ");
    scanf("%lf %lf", &x1, &y1);
    printf("Enter the point B(x2, y2) : ");
    scanf("%lf %lf", &x2, &y2);
    printf("Enter the point C(x3, y3) : ");
    scanf("%lf %lf", &x3, &y3);
    area = 0.5 * ((x1 * (y2 - y3)) + (x2 * (y3 - y1)) + (x3 * (y1 - y2)));
    if (fabs(area) < EPSILON) // abs() for integer, fabs() for float, double
        printf("\nA(%g, %g), B(%g, %g) and C(%g, %g) points fall on one straight line.", x1, y1, x2, y2, x3, y3);
    else
        printf("\nA(%g, %g), B(%g, %g) and C(%g, %g) points doesn't fall on one straight line.", x1, y1, x2, y2, x3, y3);
    return 0;
}
/* --- End of letusc/luc015.c --- */

// File: letusc/luc016.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc016.c

/* Given the coordiantes (x, y) of center of a circle and its radius,
write a program that will determine whether a point lies inside the circle,
on the circle or outside the circle. (Hint : Use sqrt() and pow() functions.) */
/* Author - Amit Dutta, Date - 1st OCT, 2025 */
/* Let Us C, Chap- 3, Page - 53, Qn No.: f(g) */

#include <stdio.h>
#include <math.h>
// Define a small tolerance value (EPSILON) for reliable floating-point comparison
#define EPSILON 0.0001

int main()
{
    double h, k;
    double R;
    double x, y;
    double distance_sq;
    printf("Enter the center coordinates (h, k) : ");
    scanf("%lf %lf", &h, &k);
    printf("Enter the radius (R) : ");
    scanf("%lf", &R);
    printf("Enter the point P coordinates (x, y) : ");
    scanf("%lf %lf", &x, &y);
    distance_sq = pow(x - h, 2) + pow(y - k, 2);
    double radius_sq = R * R;
    // Case 1: On the circle (D^2 = R^2) - Use EPSILON for safety!
    if (fabs(distance_sq - radius_sq) < EPSILON)
    {
        printf("The point P(%g, %g) lies ON THE CIRCLE.\n", x, y);
    }
    // Case 2: Inside the circle (D^2 < R^2)
    else if (distance_sq < radius_sq)
    {
        printf("The point P(%g, %g) lies INSIDE THE CIRCLE.\n", x, y);
    }
    // Case 3: Outside the circle (D^2 > R^2)
    else
    {
        printf("The point P(%g, %g) lies OUTSIDE THE CIRCLE.\n", x, y);
    }
    return 0;
}
/* --- End of letusc/luc016.c --- */

// File: letusc/luc017.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc017.c

/* Given a point (x, y), write a program to find out if it lies on X-axis, Y-axis or origin. */
/* Author - Amit Dutta, Date - 1st OCT, 2025 */
/* Let Us C, Chap- 3, Page - 53, Qn No.: f(h) */

#include <stdio.h>
#include <math.h>
#define EPSILON 0.00001

int main()
{
    double x, y;
    printf("Enter the point P(x, y) : ");
    scanf("%lf %lf", &x, &y);
```

```
    if (fabs(x) < EPSILON && fabs(y) < EPSILON)
        printf("\nPoint P(%g, %g) lies on the origin.", x, y);
    else if (fabs(x) < EPSILON)
        printf("\nPoint P(%g, %g) lies on the Y-Axis.", x, y);
    else if (fabs(y) < EPSILON)
        printf("\nPoint P(%G, %g) lies on the X-Axis.", x, y);
    else
        printf("\nThe point P(%g, %g) lies in a QUADRANT (not on an axis or the origin).", x, y);
    return 0;
}
/* --- End of letusc/luc017.c --- */

// File: letusc/luc018.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc018.c

/* According to Gregorian calender, it was Monday on the date 01/01/01.
if any year is input through the keyboard write a program to find out
what is the day on 1st January of this year. */
/* Author - Amit Dutta, Date - 1st OCT, 2025 */
/* Let Us C, Chap- 3, Page - 53, Qn No.: f(i) */

#include <stdio.h>

int is_leap(int year) {
    if ((year % 400 == 0) || (year % 4 == 0 && year % 100 != 0)) {
        return 1;
    }
    return 0;
}

int main() {
    long long year; // long long for year input if years far in the future/past are tested
    int i;
    long long total_days = 0;
    int day_index;

    // Day names array for output
    const char *day_names[] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};

    printf("Enter the year (e.g., 2025): ");
    if (scanf("%lld", &year) != 1 || year < 1) {
        printf("Invalid year input. Please enter a positive integer year (>= 1).\n");
        return 1;
    }
    int years_passed = year - 1;
    long long leap_years = years_passed / 4 - years_passed / 100 + years_passed / 400;
    total_days = years_passed * 365 + leap_years;
    day_index = (total_days + 1) % 7;
    printf("\nOn January 1st, %lld, the day was: %s.\n", year, day_names[day_index]);
    return 0;
}
/* --- End of letusc/luc018.c --- */

// File: letusc/luc018-logic.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc018-logic.c

/* According to Gregorian calender, it was Monday on the date 01/01/01.
Write a program to find out what is the day on 1st January of any input year. */
/* Author - Amit Dutta, Date - 1st OCT, 2025 */
/* Let Us C, Chap- 3, Page - 53, Qn No.: f(i) */

#include <stdio.h>

/**
 * @brief Determines if a given year is a leap year.
 * * The rule: A year is a leap year if it is divisible by 4, UNLESS it is
 * divisible by 100 but NOT by 400.
 * * @param year The year to check.
 * @return 1 if it is a leap year, 0 otherwise.
 */
int is_leap(int year) {
    // Check if divisible by 400 OR (divisible by 4 AND not divisible by 100)
    if ((year % 400 == 0) || (year % 4 == 0 && year % 100 != 0)) {
        return 1;
    }
    return 0;
}

/**
 * @brief Calculates the day of the week for January 1st of the given year.
 * * The base date is 01/01/01, which was a Monday (index 1).
 * * Day Mapping: 0:Sunday, 1:Monday, 2:Tuesday, 3:Wednesday, 4:Thursday, 5:Friday, 6:Saturday
```

```
 */
int main() {
    long long year; // Use long long for year input if years far in the future/past are tested
    int i;
    long long total_days = 0;
    int day_index;

    // Day names array for output
    const char *day_names[] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};

    printf("Enter the year (e.g., 2025): ");
    if (scanf("%lld", &year) ≠ 1 || year < 1) {
        printf("Invalid year input. Please enter a positive integer year ( ≥ 1).\n");
        return 1;
    }

    // --- Core Logic: Calculate Total Days ---

    // We only need to consider the years that have *passed* before the target year.
    // So, we count days from the end of year 0 up to the end of year (year - 1).
    int years_passed = year - 1;

    // 1. Calculate the number of leap days up to the end of year (year - 1)
    // Formula based on Gregorian calendar rules for years Y-1:
    // (Y-1)/4 - (Y-1)/100 + (Y-1)/400
    long long leap_years = years_passed / 4 - years_passed / 100 + years_passed / 400;

    // 2. Total days = (Number of years * 365) + (Number of leap years)
    // Note: The loop method (below) is more intuitive but the formula is faster.
    // We will use the direct formula for efficiency.
    total_days = years_passed * 365 + leap_years;

    // --- Alternate Loop Method (for conceptual simplicity) ---
    /*
    for (i = 1; i < year; i++) {
        total_days += 365;
        if (is_leap(i)) {
            total_days += 1; // Add 1 for the leap day
        }
    }
    */

    // --- Determine the Day of the Week ---

    // Since 01/01/01 was Monday (index 1), we use the following setup:
    // Index 1 corresponds to Monday.
    // The calculation gives the number of days *past* the Monday start (01/01/01).
    // The modulo operation gives the remainder (0-6).

    // 0 days elapsed (Year 1): total_days=0. (0 + 1) % 7 = 1 (Monday). Correct.
    // 365 days elapsed (Year 2): total_days=365. (365 + 1) % 7 = 2 (Tuesday). Correct. (365 mod 7 = 1, 1+1 = 2)

    day_index = (total_days + 1) % 7;

    // Correct the Day Index to match the array (0:Sun, 1:Mon, ... , 6:Sat)
    // The +1 adjusts for the Monday starting point (index 1).

    printf("\nOn January 1st, %lld, the day was: **%s**.\n", year, day_names[day_index]);

    return 0;
}
/* --- End of letusc/luc018-logic.c --- */

// File: letusc/luc019.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc019.c

/* If the length of three sides of a triangle are entered through the
keyboard, write a program to check whether the triangle is an isosceles,
an equilateral, a scalene or a right-angled triangle. */
/* Author - Amit Dutta, Date - 3rd OCT, 2025 */
/* Let Us C, Chap- 4, Page - 71, Qn No.: D(a) */

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

#define EPSILON 0.000001
/* EPSILON is used to compare double values for approximate equality,
mitigating floating-point precision errors. */

int main()
{
    double side1, side2, side3;
```

```
    printf("Please enter the length of three side of the triangle : ");
    scanf("%lf %lf %lf", &side1, &side2, &side3);
    if (side1 < 1 || side2 < 1 || side3 < 1)
    {
        printf("\nLength of a side of triangle should be a positive integer.");
        return 1;
    }
    if (side1 + side2 < side3 || side1 + side3 < side2 || side2 + side3 < side1)
    {
        printf("\nEntered triangle is not VALID.");
        return 1;
    }
    // isosceles check
    if (side1 == side2 || side2 == side3 || side3 == side1)
        printf("\nEntered triangle is a ISOSCELES triangle.");
    else
        printf("\nEntered triangle is NOT a ISOSCELES triangle.");
    // equilateral check
    if (side1 == side2 && side2 == side3)
        printf("\nEntered triangle is a EQUILATERAL triangle.");
    else
        printf("\nEntered triangle is NOT a EQUILATERAL triangle.");
    // scalene check
    if (side1 ≠ side2 && side2 ≠ side3)
        printf("\nEntered triangle is a SCALENE triangle.");
    else
        printf("\nEntered triangle is NOT a SCALENE triangle.");
    // right-angle check
    if (fabs(((side1 * side1) + (side2 * side2)) - (side3 * side3)) < EPSILON ||
        fabs(((side2 * side2) + (side3 * side3)) - (side1 * side1)) < EPSILON ||
        fabs(((side3 * side3) + (side1 * side1)) - (side2 * side2)) < EPSILON)
        printf("\nEntered triangle is a RIGHT-ANGLED triangle.");
    else
        printf("\nEntered triangle is NOT a RIGHT-ANGLED triangle.");
    return 0;
}
/* --- End of letusc/luc019.c --- */

// File: letusc/luc020.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc020.c

/* In digital world colors are specified in Red-Green-Blue (RGB) format,
with values of R, G, B varying on an integer scale from 0 to 255. In print
publishing the colors are mentioned in Cyan-Magenta-Yellow-Black (CMYK) format,
with values of C, M, Y, and K varying on a real scale from 0.0 to 1.0.
Write a program that converts RGB color to CMYK color as per the following formulae:
    White   = Max(Red/255, Green/255, Blue/255)
    Cyan    = (White-Red/255) / White
    Magenta = (White-Green/255) / White
    Yellow  = (White-Blue/255) / White
    Black   = 1 - White
Note that if the RGB values are all 0, then the CMY values are all 0 and the K value is 1. */
/* Author - Amit Dutta, Date - 4th OCT, 2025 */
/* Let Us C, Chap- 4, Page - 71, Qn No.: D(b) */

#include <stdio.h>

// declaring function
double get_white(double red, double green, double blue)
{
    double max;
    max = red / 255;
    if (max < (green / 255))
        max = green / 255;
    if (max < (blue / 255))
        max = blue / 255;
    return max;
}

int main()
{
    double r, g, b, w, c = 0, m = 0, y = 0, k = 0;
    printf("Enter the RGB color code in 'R G B' format : ");
    scanf("%lf %lf %lf", &r, &g, &b);

    // checking for invalid input (negetive input)
    if (r < 0 || g < 0 || b < 0)
    {
        printf("\nRGB color code can not be a negetive number.");
        return 1;
    }
```

```
    // checking for invalid input (out of range color code)
    if (r > 255 || g > 255 || b > 255)
    {
        printf("\nRGB color code can be maximum (255, 255, 255).");
        return 1;
    }

    // converting RGB color code to CMYK color code
    if (r == 0 && g == 0 && b == 0)
        k = 1;
    else
    {
        w = get_white(r, g, b);
        c = (w - (r / 255)) / w;
        m = (w - (g / 255)) / w;
        y = (w - (b / 255)) / w;
        k = 1 - w;
    }

    printf("\nRGB color (%g, %g, %g) equivalent to CMYK color (%g, %g, %g, %g).", r, g, b, c, m, y, k);
    return 0;
}
/* --- End of letusc/luc020.c --- */

// File: letusc/luc021.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc021.c

/* A certain grade of steel is graded according to the following conditions:
    (i) Hardness must be greater than 50
    (ii) Carbon content must be less than 0.7
    (iii) Tensile strength must be greater than 5600

The grades are as follows:
    Grade is 10 if all three conditions are met
    Grade is 9 if conditions (i) and (ii) are met
    Grade is 8 if conditions (ii) and (iii) are met
    Grade is 7 if conditions (i) and (iii) are met
    Grade is 6 if only one condition is met
    Grade is 5 if none of the conditions are met

Write a program, which will require the user to give values of hardness,
carbon content and tensile strength of the steel under consideration and output
the grade of the steel. */

/* Author - Amit Dutta, Date - 4th OCT, 2025 */
/* Let Us C, Chap- 4, Page - 71, Qn No.: D(c) */

#include <stdio.h>
int main()
{
    double hardness, carbon_content, tensile_strength;
    printf("Enter the details of the steel below - \n");
    printf("1. Hardness         : ");
    scanf("%lf", &hardness);
    printf("2. Carbon Content   : ");
    scanf("%lf", &carbon_content);
    printf("3. Tensile Strength : ");
    scanf("%lf", &tensile_strength);

    // storing how many conditions are met as boolean result
    int condition_met, condition1, condition2, condition3;
    condition1 = hardness > 50;
    condition2 = carbon_content < 0.7;
    condition3 = tensile_strength > 5600;
    condition_met = condition1 + condition2 + condition3;

    // now grading according the result
    int grade;
    if (condition_met == 3)
        grade = 10;
    else if (condition_met == 2)
    {
        if (condition1 && condition2)
            grade = 9;
        else if (condition2 && condition3)
            grade = 8;
        else if (condition1 && condition3)
            grade = 7;
    }
    else if (condition_met == 1)
        grade = 6;
    else
```

```
        grade = 5;

    // printing the result
    printf("\n————————————— Result —————————————");
    printf("\n1. Hardness         : Condition %s", condition1 ? "MET" : "DID NOT MET");
    printf("\n2. Carbon Content   : Condition %s", condition2 ? "MET" : "DID NOT MET");
    printf("\n3. Tensile Strength : Condition %s", condition3 ? "MET" : "DID NOT MET");
    printf("\nTotal Condition Met : %d", condition_met);
    printf("\n\nGrade : %d\n\n", grade);
    return 0;
}

/* I did not used this long variable names. I used very short just the first letter of the word.
After writting the whole program, I just renamed the valiables. This is possible in Visual Stdio Code. */
/* --- End of letusc/luc021.c --- */

// File: letusc/luc022.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc022.c

/* The Body Mass Index (BMI) is defined as ratio of  weight of the
person (in Kilograms) to square of the height (in meters).
Write a program that receives weight and height, calculate the BMI, and reports
the BMI catagory as per the following table.
    BMI Catagory            BMI
    Starvation              < 15
    Anorexic                15.1 to 17.5
    Underweight             17.6 to 18.5
    Ideal                   18.6 to 24.9
    Overweight              25 to 25.9
    Obese                   30 to 39.9
    Morbidly Obese          ⩾ 40
*/
/* Author - Amit Dutta, Date - 4th OCT, 2025 */
/* Let Us C, Chap- 4, Page - 72, Qn No.: D(d) */

#include <stdio.h>

int main()
{
    double weight, height, bmi;
    printf("Enter your Weight (in Kilograms) and Height (in Meters) : ");
    scanf("%lf %lf", &weight, &height);
    bmi = weight / (height * height);
    printf("\nCalculated BMI : %g", bmi);
    if (bmi < 15)
        printf("\nBMI Catagory : Starvation");
    else if (bmi ⩾ 15.1 && bmi ⩽ 17.5)
        printf("\nBMI Catagory : Anorexic");
    else if (bmi ⩾ 17.6 && bmi ⩽ 18.5)
        printf("\nBMI Catagory : Underweight");
    else if (bmi ⩾ 18.6 && bmi ⩽ 24.9)
        printf("\nBMI Catagory : Ideal");
    else if (bmi ⩾ 25 && bmi ⩽ 25.9)
        printf("\nBMI Catagory : Overweight");
    else if (bmi ⩾ 30 && bmi ⩽ 39.9)
        printf("\nBMI Catagory : Obese");
    else if (bmi ⩾ 40)
        printf("\nBMI Catagory : Morbidly Obese");
    else
        printf("\nBMI Catagory : Unclassified");
    return 0;
}
/* --- End of letusc/luc022.c --- */

// File: letusc/luc023.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc023.c

/* Using conditional operators determine :
    (1) Whether the character entered through the keyboard is a
        lower case alphabet or not.
    (2) Whether a character entered through the keyboard is a special
        symbol or not. */
/* Author - Amit Dutta, Date - 5th OCT, 2025 */
/* Let Us C, Chap- 4, Page - 72, Qn No.: E(a) */

#include <stdio.h>
int main()
{
    char inp;
    printf("Enter the character : ");
    scanf("%c", &inp);
    printf("\nInput Character '%c' is %s a LOWER CASE ALPHABET.", inp,
```

```
        (inp ⩾ 'a' && inp ⩽ 'z') ? "" : "NOT");
    printf("\nInput Character '%c' is %s a SPECIAL SYMBOL.", inp,
        (inp ⩾ 'a' && inp ⩽ 'z' || inp ⩾ 'A' && inp ⩽ 'Z'
            || inp ⩾ '0' && inp ⩽ '9') ? "NOT" : "");
    return 0;
}
/* --- End of letusc/luc023.c --- */

// File: letusc/luc024.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc024.c

/* Write a program using conditional operators to determine whether
a year entered through the keyboard is a leap year or not. */
/* Author - Amit Dutta, Date - 5th OCT, 2025 */
/* Let Us C, Chap- 4, Page - 72, Qn No.: E(b) */

#include <stdio.h>
int main()
{
    int year;
    printf("Enter the year : ");
    scanf("%d", &year);
    printf("\nYear %d is %s a Leapyear.", year, (year % 400 == 0 || year % 4 == 0 && year % 100 ≠ 0) ? "" : "NOT");
    return 0;
}
/* --- End of letusc/luc024.c --- */

// File: letusc/luc025.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc025.c

/* Write a program to find the greates of the three numbers entered
through the keyboard. Use conditional operators. */
/* Author - Amit Dutta, Date - 6th OCT, 2025 */
/* Let Us C, Chap- 4, Page - 72, Qn No.: E(c) */

#include <stdio.h>
int main()
{
    double num1, num2, num3, max;
    printf("Enter three number : ");
    scanf("%lf %lf %lf", &num1, &num2, &num3);
    printf("\nGreatest of the three number '%g', '%g' and '%g' is : '%g'", num1, num2, num3,
        (num1 > num2 && num1 > num3) ? num1 : ((num2 > num1 && num2 > num3) ? num2 : num3));
    return 0;
}
/* --- End of letusc/luc025.c --- */

// File: letusc/luc026.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc026.c

/* Write a program to recieve value of an angle in degreesand check
whether sum of squares of sine and cosine of this angle is equal to 1. */
/* Author - Amit Dutta, Date - 6th OCT, 2025 */
/* Let Us C, Chap- 4, Page - 73, Qn No.: E(d) */

#include <stdio.h>
#include <math.h>
#define EPSILON 0.0000001

int main()
{
    double angle, result;
    printf("Enter the angle value in degree : ");
    // checking if the input is other than number.
    if(scanf("%lf", &angle) ≠ 1) {
        printf("\nPlease enter a number.");
        return 1;
    }
    angle = angle * (M_PI / 180); // converting degree to radian
    result = pow(sin(angle), 2) + pow(cos(angle), 2);
    (fabs(result - 1.0) < EPSILON) ?
    printf("\nsum of squares of sine and cosine of this angle is equal to 1.") :
    printf("\nsum of squares of sine and cosine of this angle is NOT equal to 1.");
    return 0;
}
/* --- End of letusc/luc026.c --- */

// File: letusc/luc027.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc027.c

/* Rewrite the folowing program using conditional operations
    #include<stdio.h>
```

```
    int main()
    {
        float sal;
        printf("Enter the salary");
        scanf("%f", &sal);
        if(sal ⩾ 25000 && sal ⩽ 40000)
            printf("Manager\n");
        else
            if(sal ⩾ 15000 && sal < 25000)
                printf("Accountant\n");
            else
                printf("Clerk\n");
        return 0;
    }
*/
/* Author - Amit Dutta, Date - 6th OCT, 2025 */
/* Let Us C, Chap- 4, Page - 73, Qn No.: E(e) */

#include <stdio.h>
int main()
{
    float sal;
    printf("Enter the salary");
    scanf("%f", &sal);
    (sal ⩾ 25000 && sal ⩽ 40000) ? printf("Manager\n") :
    ((sal ⩾ 15000 && sal < 25000) ? printf("Accountant\n") : printf("Clerk\n"));
    return 0;
}
/* --- End of letusc/luc027.c --- */

// File: letusc/luc028.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc028.c

/* Write a program to print all the ASCII values and their equivalent
characters using a while loop. The ASCII may vary from 0 to 255. */
/* Author - Amit Dutta, Date - 7th OCT, 2025 */
/* Let Us C, Chap- 5, Page - 87, Qn No.: B(a) */

#include <stdio.h>
int main()
{
    int i = 0;
    printf("ASCII Value\tCharacter");
    printf("\n————————\t————————\n");
    while (i ⩽ 255)
    {
        printf("%d.\t%c\n\n", i, i);
        i++;
    }
    return 0;
}
/* --- End of letusc/luc028.c --- */

// File: letusc/luc029.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc029.c

/* Write a program to print out all Armstrong numbers between 100
and 500. If sum of cubes of each digit of the number is equal to the
number itself, then the number is called an Armstrong number. For
example, 153 = (1 * 1 * 1) + (5 * 5 * 5) + (3 * 3 * 3) */
/* Author - Amit Dutta, Date - 7th OCT, 2025 */
/* Let Us C, Chap- 5, Page - 87, Qn No.: B(b) */

#include <stdio.h>
#include <math.h>
int main()
{
    int num = 100, temp1, temp2, res;
    printf("Armstrong numbers between 100 and 500 :");
    while (num ⩽ 500)
    {
        temp1 = num;
        res = 0;
        while (temp1 ≠ 0)
        {
            temp2 = temp1 % 10;
            res = res + pow(temp2, 3);
            temp1 = temp1 / 10;
        }
        if (num == res)
            printf("  %d", num);
        num++;
```

```
    }
    return 0;
}
/* --- End of letusc/luc029.c --- */

// File: letusc/luc030.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc030.c

/* Write a program for a matchstick game being played between the
computer and a user. Your program should ensure that the computer
always wins. Rules for the game are as follows :
    - There are 21 matchsticks.
    - The computer asks the player to pick 1, 2, 3, or 4 matchsticks.
    - After the person picks, the computer does its picking.
    - Whoever is forced to pick up the last matchstick loses the game.
*/
/* Author - Amit Dutta, Date - 8th OCT, 2025 */
/* Let Us C, Chap- 5, Page - 87, Qn No.: B(c) */

/* My Plan: The total number of matchsticks is 21. To guarantee a win,
the computer ensures that after its turn, the number of remaining
matchsticks is always a multiple of 5 plus 1 (i.e., 16, 11, 6, 1).
This is achieved by making the sum of the player's pick and the
computer's pick equal to 5 in each round. This forces the player
to pick the final matchstick. */

#include <stdio.h>
int main()
{
    int remaining_matchsticks = 21, player_pick, computer_pick;
    printf("                  --- Matchstick Game ---\n");
    printf("Rules: There are 21 matchsticks. You can pick 1, 2, 3, or 4.\n");
    printf("Whoever is forced to pick the last matchstick loses the game.\n");
    while (remaining_matchsticks > 1)
    {
        // game start
        printf("\n----------------------------");
        printf("\nRemaining Matchsticks : %d", remaining_matchsticks);

        // player pick and checking input is valid or not
        printf("\nYour turn: Pick 1, 2, 3, or 4 matchsticks: ");
        if (scanf("%d", &player_pick) != 1)
        {
            printf("\n\tPlease enter a number.");
            while (getchar() != '\n')
                ;
            continue;
        }

        // checking player pick is valid or not
        if (player_pick < 1 || player_pick > 4)
        {
            printf("\n\tPlease enter a number among 1, 2, 3 and 4.");
            while (getchar() != '\n')
                ;
            continue;
        }

        if (player_pick > remaining_matchsticks)
        {
            printf("\nInvalid choice! There are not enough matchsticks left.");
            while (getchar() != '\n')
                ;
            continue;
        }

        // computer_picks
        computer_pick = 5 - player_pick;
        printf("\ncomputer_picks : %d", computer_pick);

        // remaining matchsticks
        remaining_matchsticks = remaining_matchsticks - (player_pick + computer_pick);
    }

    // game over
    printf("\n--------------------------------------\n");
    printf("Only 1 matchstick is left.\n");
    printf("You are forced to pick the last matchstick. You lose!\n");
    printf("The computer wins.\n");

    return 0;
}
```

```
/* --- End of letusc/luc030.c --- */

// File: letusc/luc031.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc031.c

/* Write a program to enter numbers till the user wants. At the end it
should display the count of positive, negative and zeros entered. */
/* Author - Amit Dutta, Date - 8th OCT, 2025 */
/* Let Us C, Chap- 5, Page - 87, Qn No.: B(d) */

#include <stdio.h>
int main()
{
    int choice = 1, num, positive_count = 0, negative_count = 0, zero_count = 0;
    while (choice == 1)
    {
        printf("\nEnter the number (Type any character and press Enter to finish.) : ");
        choice = scanf("%d", &num); // Checking whether the user has input any characters
        if (choice == 1)
        {
            printf("Number recorded : %d", num);
            if (num < 0)
                negative_count++;
            else if (num > 0)
                positive_count++;
            else if (num == 0)
                zero_count++;
        }
        else
        {
        // If the user inputs any characters, then choice = 0, it means he doesn't want to give any more input;
            choice = 0;
            printf("\nCharacter received. Stopping input...\n");
        }
    }
    // Display the final results
    printf("\n=======================================\n");
    printf("   Analysis Complete\n");
    printf("=======================================\n");
    printf("Positive numbers entered: %d\n", positive_count);
    printf("Negative numbers entered: %d\n", negative_count);
    printf("Zeroes entered:           %d\n", zero_count);
    printf("Total numbers recorded:   %d\n", positive_count + negative_count + zero_count);
    printf("=======================================\n");
}
/* --- End of letusc/luc031.c --- */

// File: letusc/luc031-logic.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc031-logic.c

/* Write a program to enter numbers till the user wants. At the end it
should display the count of positive, negative and zeros entered. */
/* Author - Amit Dutta, Date - 8th OCT, 2025 */
/* Let Us C, Chap- 5, Page - 87, Qn No.: B(d) */

/*
 * DETAILED PROGRAM PLAN:
 * 1. INITIALIZATION: Set three counters (positive_count, negative_count, zero_count) to zero.
 * 2. INPUT LOOP: Start an infinite loop to continuously request user input.
 * 3. READ INPUT: Use fgets() to read the user's input as a generic string (char array).
 * This allows the program to accept multi-digit numbers (e.g., "-500") or the command ('n').
 * 4. TERMINATION CHECK: Immediately check if the input string is exactly "n" using strcmp().
 * If true, the user wants to quit, so break the loop.
 * 5. VALIDATION & CONVERSION: If the input is not "n", use sscanf() to safely attempt to convert
 * the string into an integer.
 * 6. COUNTING LOGIC: If sscanf() successfully reads an integer (sscanf returns 1):
 * - If the number is > 0, increment positive_count.
 * - If the number is < 0, increment negative_count.
 * - If the number is == 0, increment zero_count.
 * 7. ERROR HANDLING: If the input is neither "n" nor a valid number (sscanf returns 0),
 * inform the user of invalid input and continue the loop.
 * 8. FINAL DISPLAY: After the loop terminates, print the final totals for positive, negative, and zero counts.
 */

#include <stdio.h>
#include <stdlib.h> // for strtol
#include <string.h> // for strcmp

// Maximum size of the input line
#define MAX_INPUT_LEN 15

int main() {
```

```
    // Initialize counters
    int positive_count = 0;
    int negative_count = 0;
    int zero_count = 0;

    // Buffer to store the user's input as a string (e.g., "123", "-50", or "n")
    char input_buffer[MAX_INPUT_LEN];
    int number;

    printf("--- Number Analyzer ---\n");
    printf("Enter numbers one by one. Type 'n' and press Enter to finish.\n\n");

    // Loop until the user enters 'n'
    while (1) {
        printf("Enter number or 'n': ");

        // Read the entire line of input into the buffer
        if (fgets(input_buffer, MAX_INPUT_LEN, stdin) == NULL) {
            // Handle EOF (end of file) or reading error
            break;
        }

        // Remove the trailing newline character from the input_buffer
        // The last character will be '\n' if the input was shorter than MAX_INPUT_LEN
        size_t len = strlen(input_buffer);
        if (len > 0 && input_buffer[len - 1] == '\n') {
            input_buffer[len - 1] = '\0';
        }

        // 1. Check for the termination condition
        if (strcmp(input_buffer, "n") == 0) {
            printf("\n'n' received. Stopping input...\n");
            break; // Exit the while loop
        }

        // 2. Attempt to convert the input string to an integer
        // sscanf attempts to read the string according to the format "%d" (decimal integer)
        // It returns 1 if a number was successfully read.
        int conversions = sscanf(input_buffer, "%d", &number);

        if (conversions == 1) {
            // Conversion was successful, now check the number's sign
            if (number > 0) {
                positive_count++;
            } else if (number < 0) {
                negative_count++;
            } else {
                zero_count++;
            }
            printf("  -> Number recorded: %d\n", number);
        } else {
            // Conversion failed. The input was neither 'n' nor a valid integer.
            printf("  -> Invalid input. Please enter a valid number or 'n'.\n");
        }
    }

    // Display the final results
    printf("\n========================================\n");
    printf("   Analysis Complete\n");
    printf("========================================\n");
    printf("Positive numbers entered: %d\n", positive_count);
    printf("Negative numbers entered: %d\n", negative_count);
    printf("Zeroes entered:           %d\n", zero_count);
    printf("Total numbers recorded:   %d\n", positive_count + negative_count + zero_count);
    printf("========================================\n");

    return 0;
}
/* --- End of letusc/luc031-logic.c --- */

// File: letusc/luc032.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc032.c

/* Write a program to recieve an integer and find its octal equivalent.
(Hint : To obtain octal equivalent of an integer, Divide it continuously
by 8 till dividend does not become zero, then write the remainders
obtained in reverse derection.) */
/* Author - Amit Dutta, Date - 8th OCT, 2025 */
/* Let Us C, Chap- 5, Page - 87, Qn No.: B(e) */

// using array
```

```c
#include <stdio.h>
int main()
{
    int octal[20], decimal, index = 0, temp, rem;
    printf("Enter the decimal number : ");
    scanf("%d", &decimal);
    temp = decimal;
    while (temp ≠ 0)
    {
        rem = temp % 8;
        temp = temp / 8;
        octal[index] = rem;
        index++;
    }
    printf("\nDeciaml %d to octal : ", decimal);
    while ((index - 1) ≥ 0)
    {
        printf("%d", octal[index - 1]);
        index--;
    }
    return 0;
}
/* --- End of letusc/luc032.c --- */

// File: letusc/luc033.c
// URL: https://github.com/notamitgamer/bsc/blob/main/letusc/luc033.c

/* Write a program to find the range of a set of numbers entered
through the keyboard. Range is the difference between the smallest
and biggest number in the list. */
/* Author - Amit Dutta, Date - 8th OCT, 2025 */
/* Let Us C, Chap- 5, Page - 87, Qn No.: B(f) */

#include <stdio.h>
int main()
{
    int choice = 1, set_of_numbers[30], num, index = -1;
    while (choice == 1)
    {
        printf("\nEnter the number (Type any character and press Enter to finish.) : ");
        choice = scanf("%d", &num); // Checking whether the user has input any characters
        if (choice ≠ 1)
        {
            // If the user inputs any characters, then choice = 0, it means he doesn't want to give any more input;
            choice = 0;
            printf("\nCharacter received. Stopping input...\n");
            break;
        }
        index++;
        set_of_numbers[index] = num;
    }
    int max = set_of_numbers[0], min = set_of_numbers[0];
    while (index ≥ 0)
    {
        if (max < set_of_numbers[index])
            max = set_of_numbers[index];
        if (min > set_of_numbers[index])
            min = set_of_numbers[index];
        index--;
    }
    int range = max - min;
    printf("\nBiggest number in the set  : %d", max);
    printf("\nSmallest number in the set : %d", min);
    printf("\nRange : %d", range);
    return 0;
}
/* --- End of letusc/luc033.c --- */
```