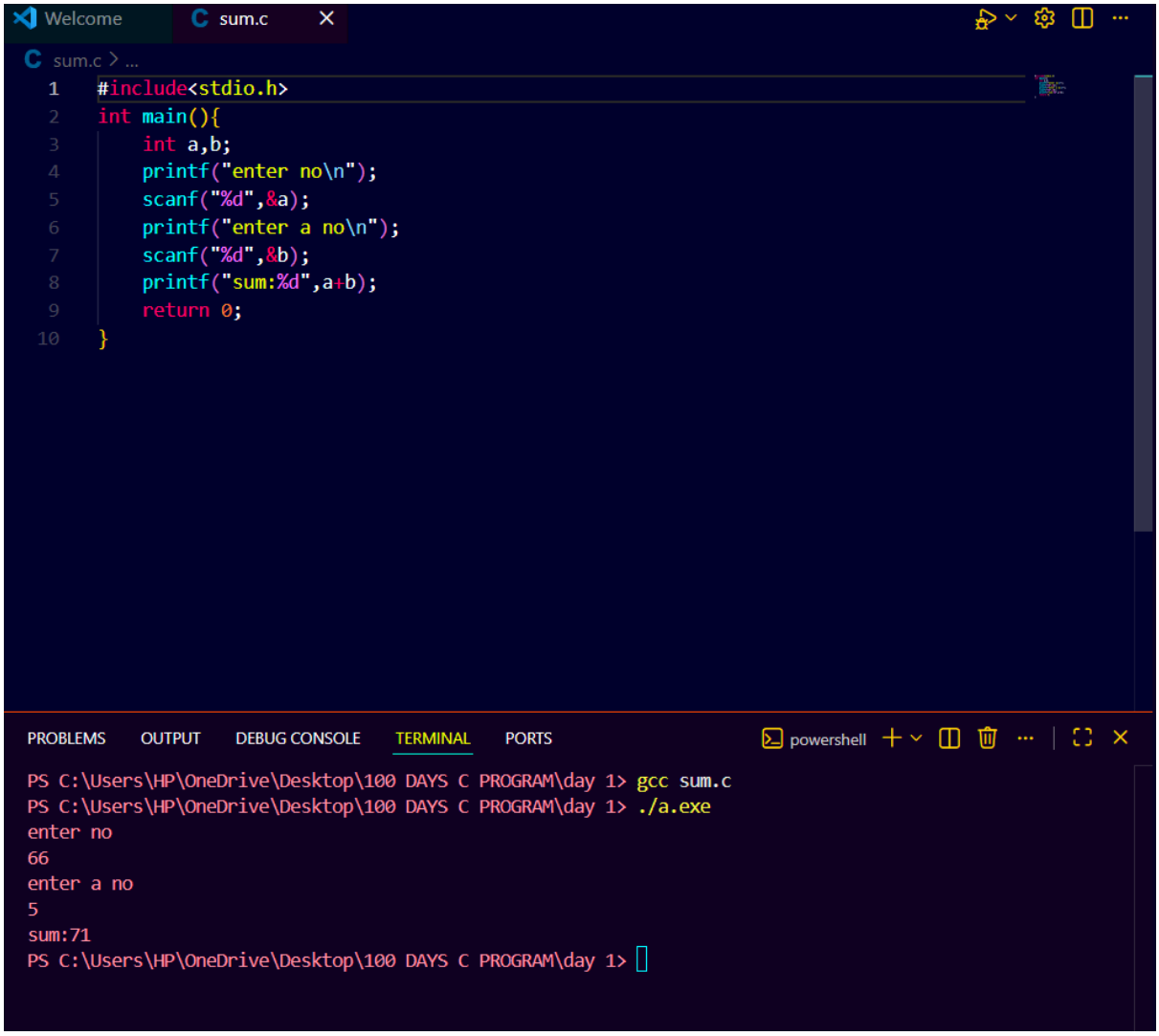
C -PROGRAMMING 100 DAYS CODE

NAME : VINIT KUMAR

SAP ID : 590029353

DAY -1

Q1 -Write a program to input two numbers and display their sum



Q2 -Write a program to input two numbers and display their sum, difference, product, and quotient

operation.c > ...

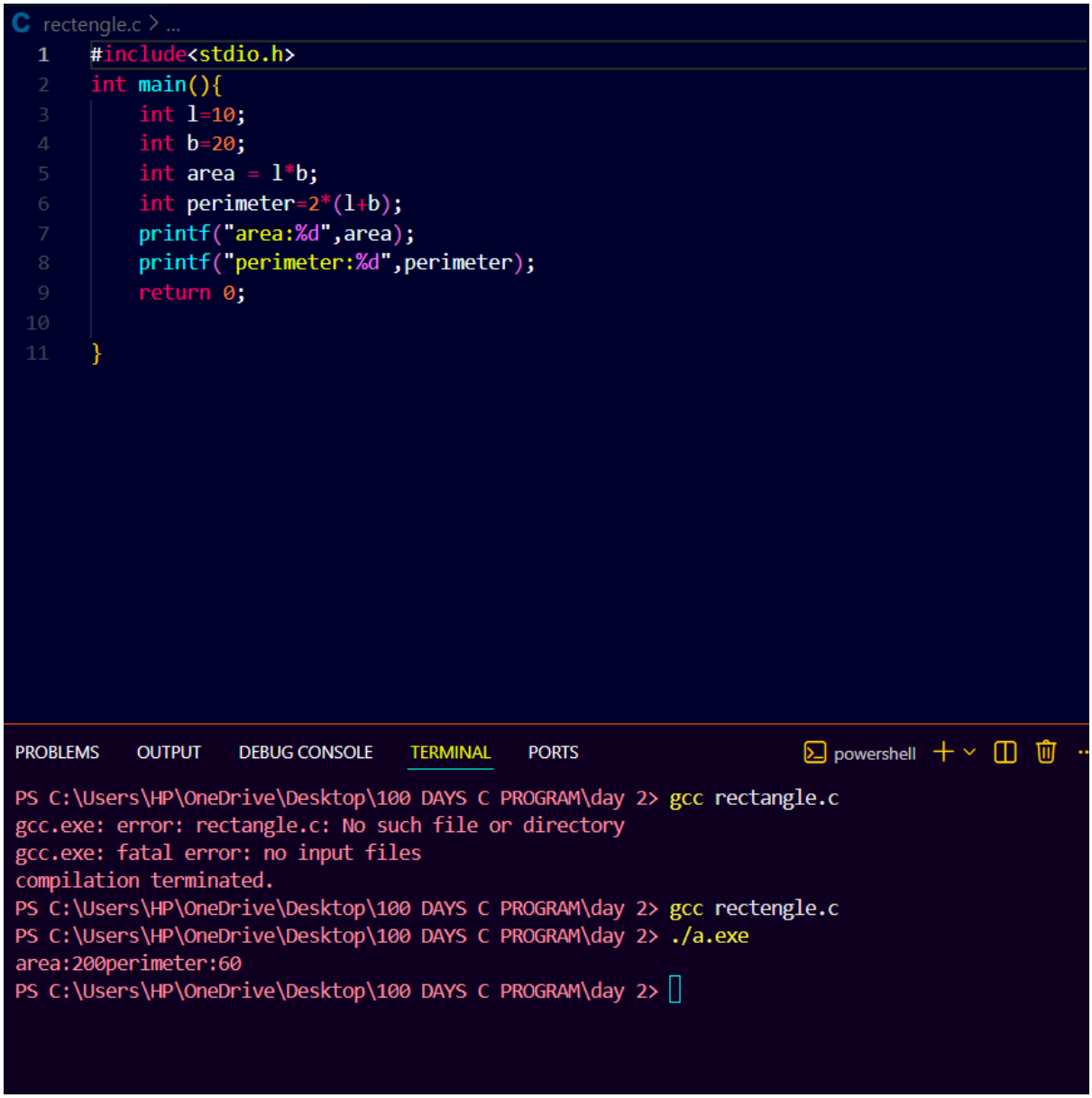
```c
#include<stdio.h>
int main(){
    int a,b;
    printf("enter a no\n");
    scanf("%d",&a);
    printf("enter a no\n");
    scanf("%d",&b);
    printf("sum:%d",a+b);
    printf("difference :%d",a-b);
    printf("product:%d",a*b);
    printf("quotient:%d",a/b);
    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 1> gcc operation.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 1> ./a.exe
enter a no
5
enter a no
3
sum:8difference :2product:15quotient:1
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 1>
```

DAY-2

Q3 -Write a program to calculate the area and perimeter of a rectangle given its length and breadth.



Q4 -Write a program to calculate the area and circumference of a circle given its radius.

DAY-3

Q5 -Write a program to convert temperature from Celsius to Fahrenheit.

convert.c > main()

```c
#include<stdio.h>
int main(){
    int c, f;
    f=9/5*(c)+32;
    printf("enter celcius\n");
    scanf("%d",&c);
    printf("fahrenheit:%d",f);
    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 3> gcc convert.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 3> ./a.exe
enter celcius
37
fahrenheit:32
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 3>
```

Q6 -Write a program to swap two numbers using a third variable.

Welcome | convert.c | swap.c

swap.c > main()

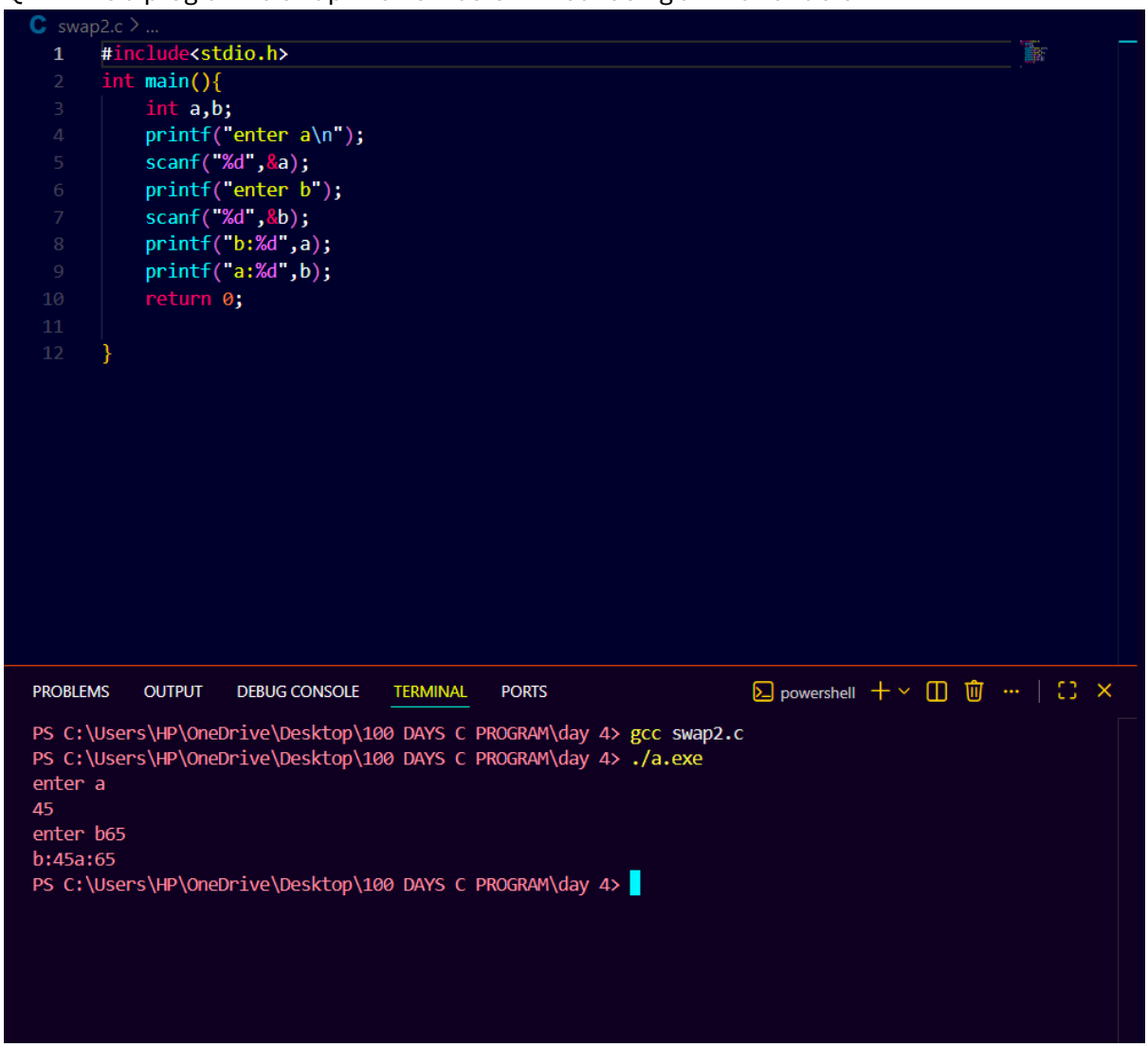
```c
#include<stdio.h>
int main (){
    int a,b,c;
    printf("enter a\n");
    scanf("%d",&a);
    printf("enter b\n");
    scanf("%d",&b);
    c=b;
    printf("a:%d",c);
    printf("b:%d", a);
    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 3> gcc swap.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 3> ./a.exe
enter a
66
enter b
45
a:45b:66
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 3>
```

DAY-4

Q7 -Write a program to swap two numbers without using a third variable.



Q8 -Write a program to find and display the sum of the first n natural numbers.

sum.c > ...

```c
#include<stdio.h>

int main(){
    int n;
    printf("enter no\n");
    scanf("%d",&n);
    int sum =(1+n)*(n/2);
    printf("sum of first n natural no:%d",sum);
    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 4> gcc sum.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 4> ./a.exe
enter no
55
sum of first n natural no:1512
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 4>
```

DAY-5

Q9-Write a program to calculate simple and compound interest for given principal, rate, and time.

interest.c > ...

```c
#include<stdio.h>
#include<math.h>
int main(){
    float p,r,t,si,ci;
    printf("Enter the p:\n");
    scanf("%f",&p);
    printf("Enter the r:\n");
    scanf("%f",&r);
    printf("Enter the t:\n");
    scanf("%f",&t);

    si = (p*r*t)/100;
    ci = p * pow((1 + r / 100), t)-p;
    printf("%.2f",si);
    printf("%.2f",ci);

    return 0;
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 5> GCC INTEREST.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 5> ./A.EXE
Enter the p:
500
Enter the r:
87
Enter the t:
5
2175.0010933.47
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 5>
```

Q10 -Write a program to input time in seconds and convert it to hours:minutes:seconds format.

time.c > main()

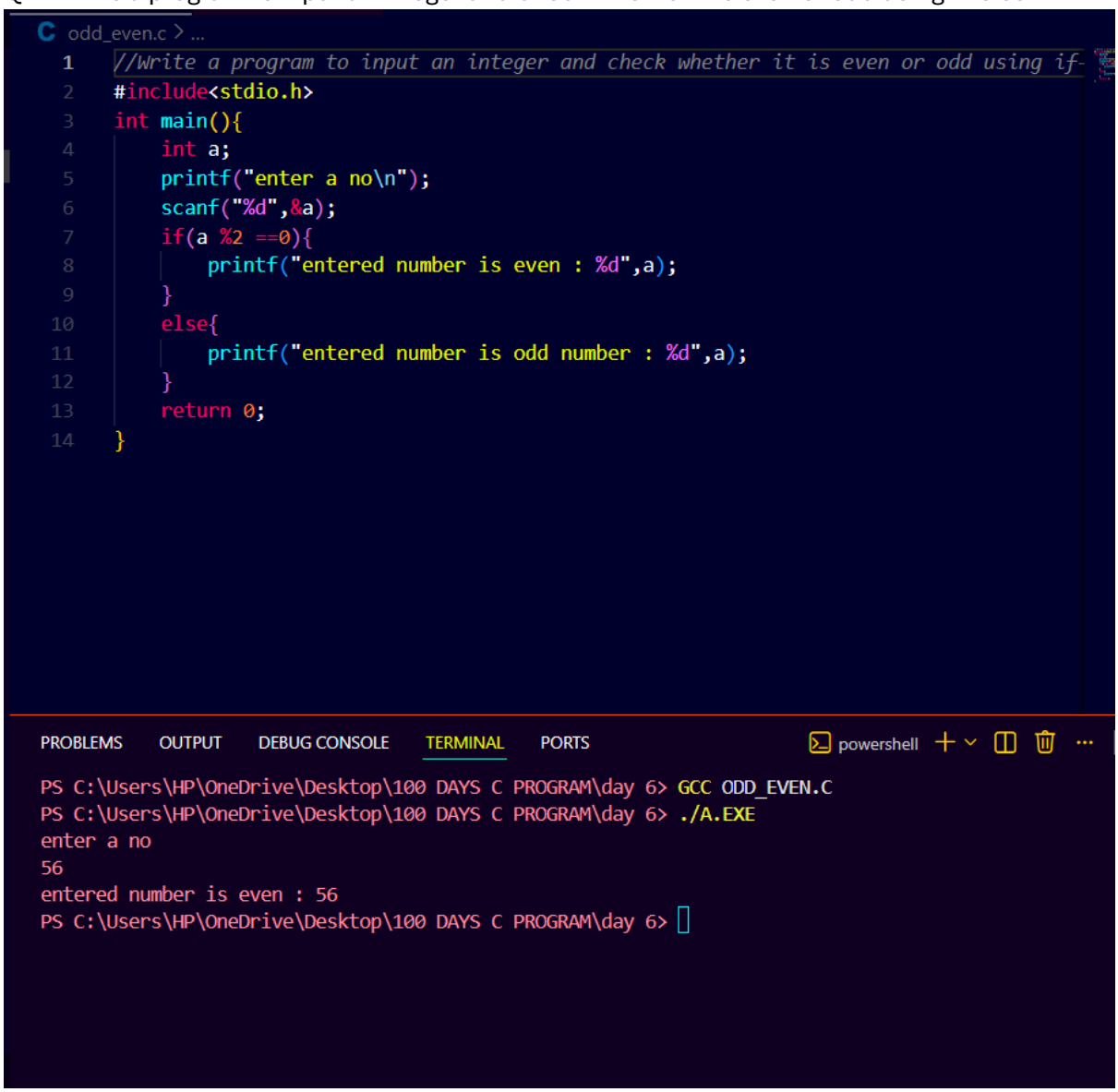
```c
#include<stdio.h>
int main(){
    int h,m,s,t;
    printf("enter second\n");
    scanf("%d",&t);
    h=t/3600;int a=t%3600;
     m=a/60;
     s=a%60;
    printf("%d:%d:%d",h,m,s);
    return 0;


}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 5> gcc time.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 5> ./a.exe
enter second
67868
18:51:8
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 5>
```

DAY-6

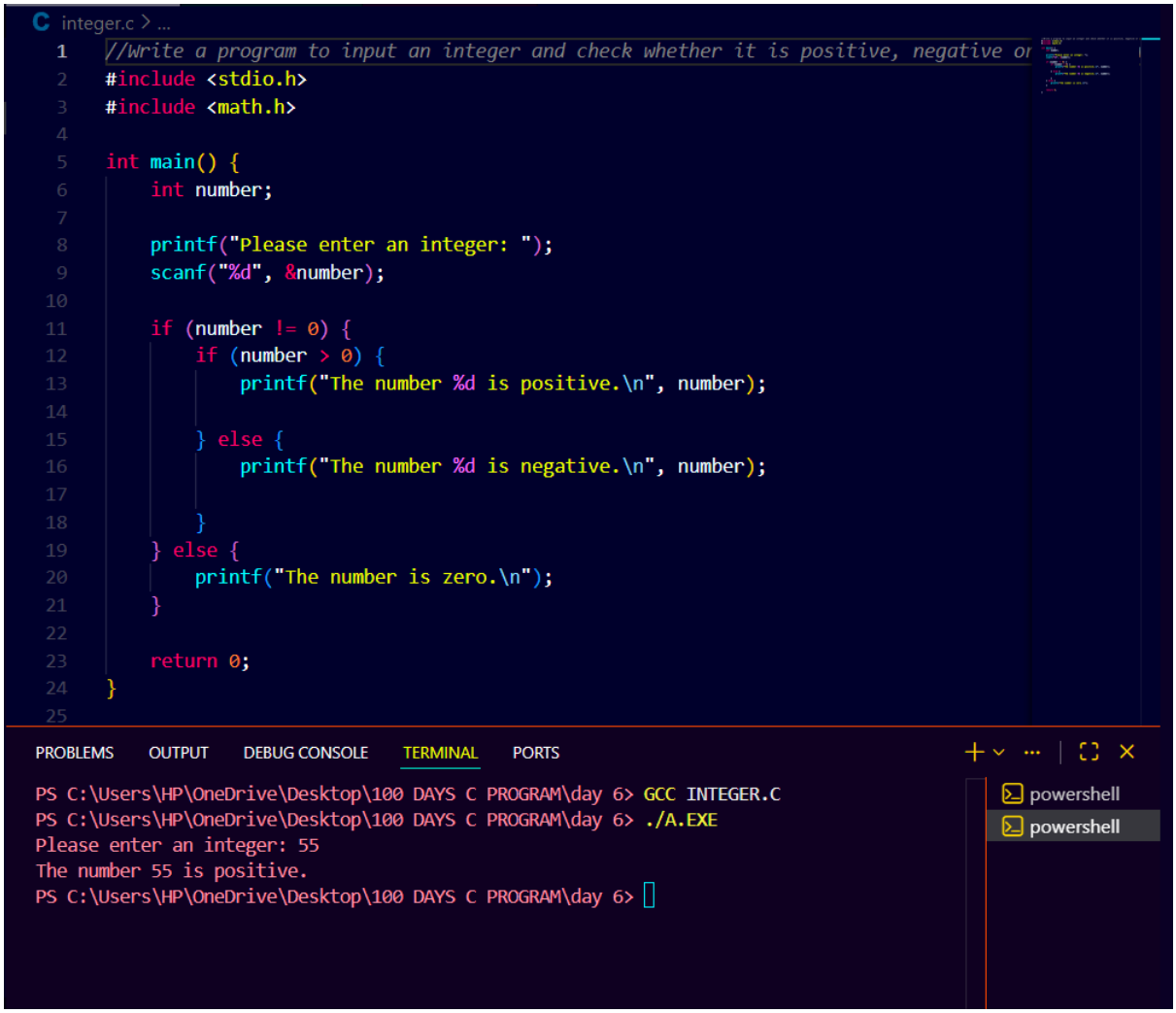
Q11-Write a program to input an integer and check whether it is even or odd using if–else



```c
//Write a program to input an integer and check whether it is even or odd using if-
#include<stdio.h>
int main(){
    int a;
    printf("enter a no\n");
    scanf("%d",&a);
    if(a %2 ==0){
        printf("entered number is even : %d",a);
    }
    else{
        printf("entered number is odd number : %d",a);
    }
    return 0;
}
```

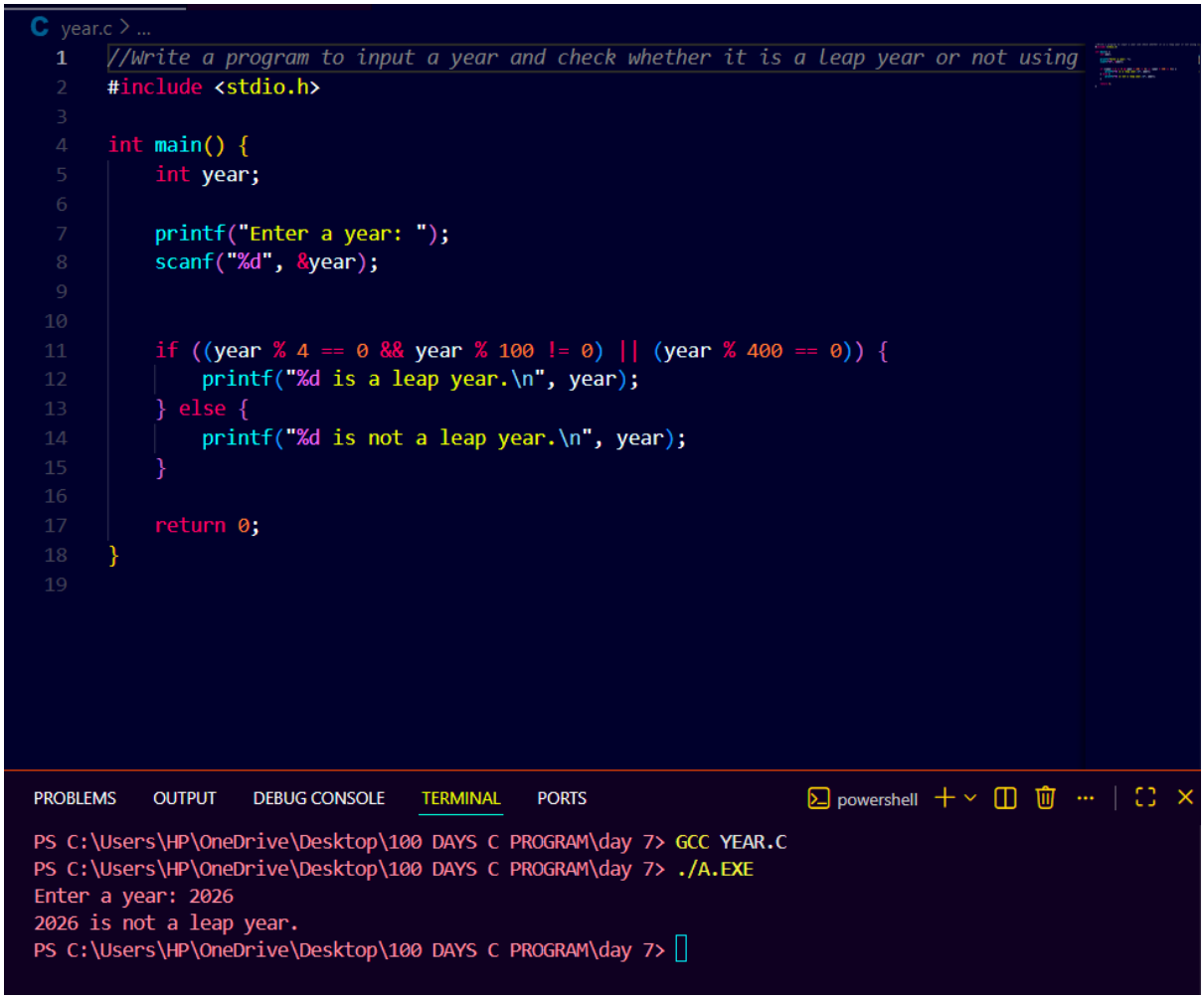
```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 6> GCC ODD_EVEN.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 6> ./A.EXE
enter a no
56
entered number is even : 56
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 6>
```

Q12 -Write a program to input an integer and check whether it is positive, negative or zero using nested if–else.



DAY-7

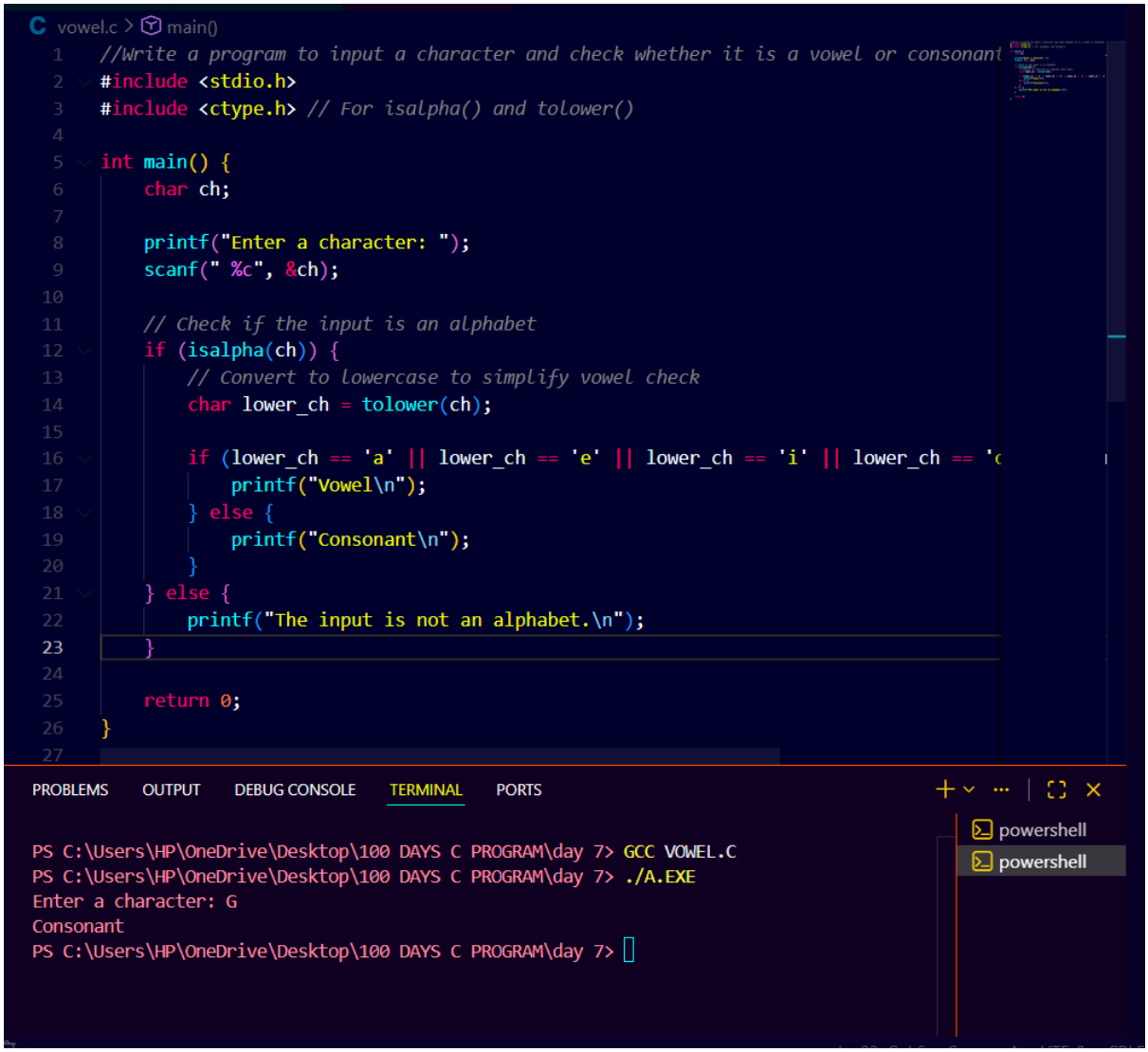
Q13 -Write a program to input a year and check whether it is a leap year or not using conditional statements.



```c
//Write a program to input a year and check whether it is a leap year or not using
#include <stdio.h>

int main() {
    int year;

    printf("Enter a year: ");
    scanf("%d", &year);


    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
        printf("%d is a leap year.\n", year);
    } else {
        printf("%d is not a leap year.\n", year);
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 7> GCC YEAR.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 7> ./A.EXE
Enter a year: 2026
2026 is not a leap year.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 7>
```

Q14-Write a program to input a character and check whether it is a vowel or consonant using if–else.



DAY-8

Q15 -Write a program to input a character and check whether it is an uppercase alphabet, lowercase alphabet, digit, or special character.

check.c > ...

```c

#include <stdio.h>
#include <ctype.h>

int main() {
    char ch;

    printf("--- Character Type Checker (C) ---\n");
    printf("Enter a single character: ");

    scanf(" %c", &ch);


    if (isupper(ch)) {
        printf("The character '%c' is an Uppercase Alphabet.\n", ch);
    }

    else if (islower(ch)) {
        printf("The character '%c' is a Lowercase Alphabet.\n", ch);
    }
    else if (isdigit(ch)) {
        printf("The character '%c' is a Digit.\n", ch);
    }

    else {
        printf("The character '%c' is a Special Character.\n", ch);
    }

    return 0;
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 8> GCC CHECK.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 8> ./A.EXE
--- Character Type Checker (C) ---
Enter a single character: H
The character 'H' is an Uppercase Alphabet.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 8>
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF { } C Signed out windows-gcc-x86

Q16 -Write a program to input three numbers and find the largest among them using if–else.

Welcome | check.c | largest_number.c

largest_number.c > ...

```c
//Write a program to input three numbers and find the largest among them using if-else.

#include <stdio.h>
int main(){
    int a ,b, c;
    printf("enter no\n");
    scanf("%d",&a);
    printf("enter number\n");
    scanf("%d",&b);
    printf("enter number\n");
    scanf("%d",&c);
    if( a > b && a>c){
        printf("%d is the largest number\n ",a);

    }
    else if ( b >a && b >c)
    {
        printf(" %d is the largest number", b);
    }
    else {
        printf(" %d is the largest number", c);
    }
    return 0;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 8> GCC LARGEST_NUMBER.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 8> ./A.EXE
enter no
7
enter number
6
enter number
5
7 is the largest number
```

DAY-9

Q17 -Write a program to find the roots of a quadratic equation and categorize them.

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

int main() {
    double a, b, c;
    double discriminant, root1, root2, realPart, imagPart;

    printf("--- Quadratic Equation Root Finder ---\n");
    printf("Enter coefficients a, b, and c (separated by spaces): ");


    scanf("%lf %lf %lf", &a, &b, &c);
discriminant = b * b - 4 * a * c;
if (discriminant > 0) {
        root1 = (-b + sqrt(discriminant)) / (2 * a);
        root2 = (-b - sqrt(discriminant)) / (2 * a);

        printf("Roots are real and different: %g, %g\n", root1, root2);
    }

    else if (discriminant == 0) {
        root1 = root2 = -b / (2 * a);
        printf("Roots are real and same: %g\n", root1);
    }

    else {


        realPart = -b / (2 * a);
        imagPart = sqrt(-discriminant) / (2 * a);
        printf("Roots are complex\n");


    }

    return 0;
}
```

Ln 3, Col 19 Spaces: 4 UTF-8

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS        powershell

PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 9> GCC ROOT.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 9> ./A.EXE
--- Quadratic Equation Root Finder ---
Enter coefficients a, b, and c (separated by spaces): 4
7
9
Roots are complex
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 9>
```

Q18 -Write a program that accepts a percentage (0-100) and assigns a grade based on the following .

grade.c > ...

```c
//Write a program that accepts a percentage (0-100) and assigns a grade based on th
#include<stdio.h>
int main (){
    int a;
    printf("enter percentage\n");
    scanf("%d",&a);
    if( a >= 90 && a <= 100){
        printf("GRADE-A %d",a);
    }
    else if ( a >=80 && a<=89)
    {
        printf("GRADE-B %d",a);
    }
    else if (a >=70 && a<= 79)
    {
        printf("GRADE - C %d",a);
    }
    else if ( a>=60 && a<=69)
    {
        printf("GRADE-D %d",a);
    }
    else if (a>=0 && a<=59)
    {
        printf("GRADE-F %d",a);
    }
    else{
        printf(" not valid percentage");
    }
    return 0;



}
```

Ln 1, Col 106 Spaces: 4 UTF-8

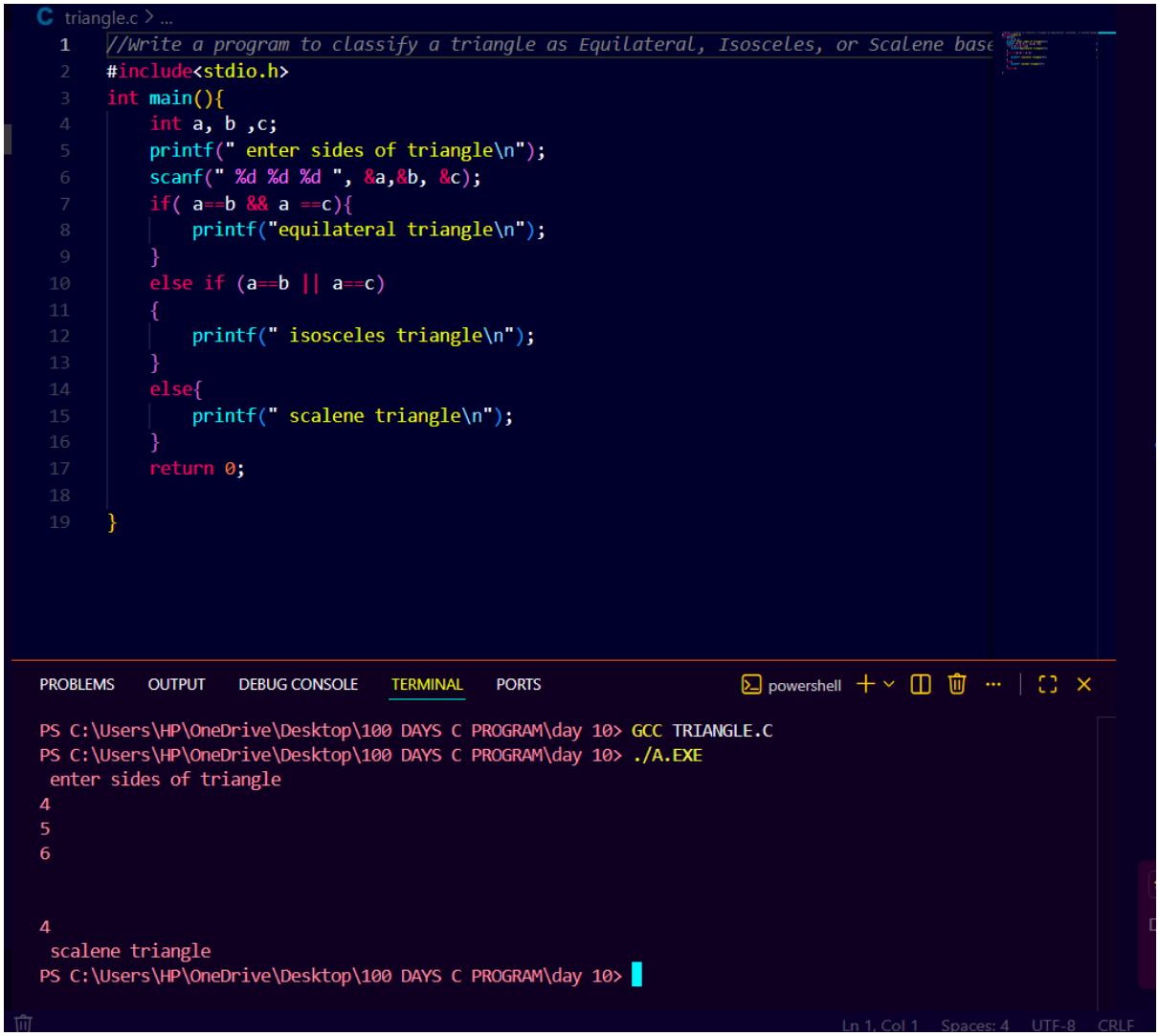
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 9> GCC GRADE.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 9> ./A.EXE
enter percentage
90
GRADE-A 90
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 9>
```

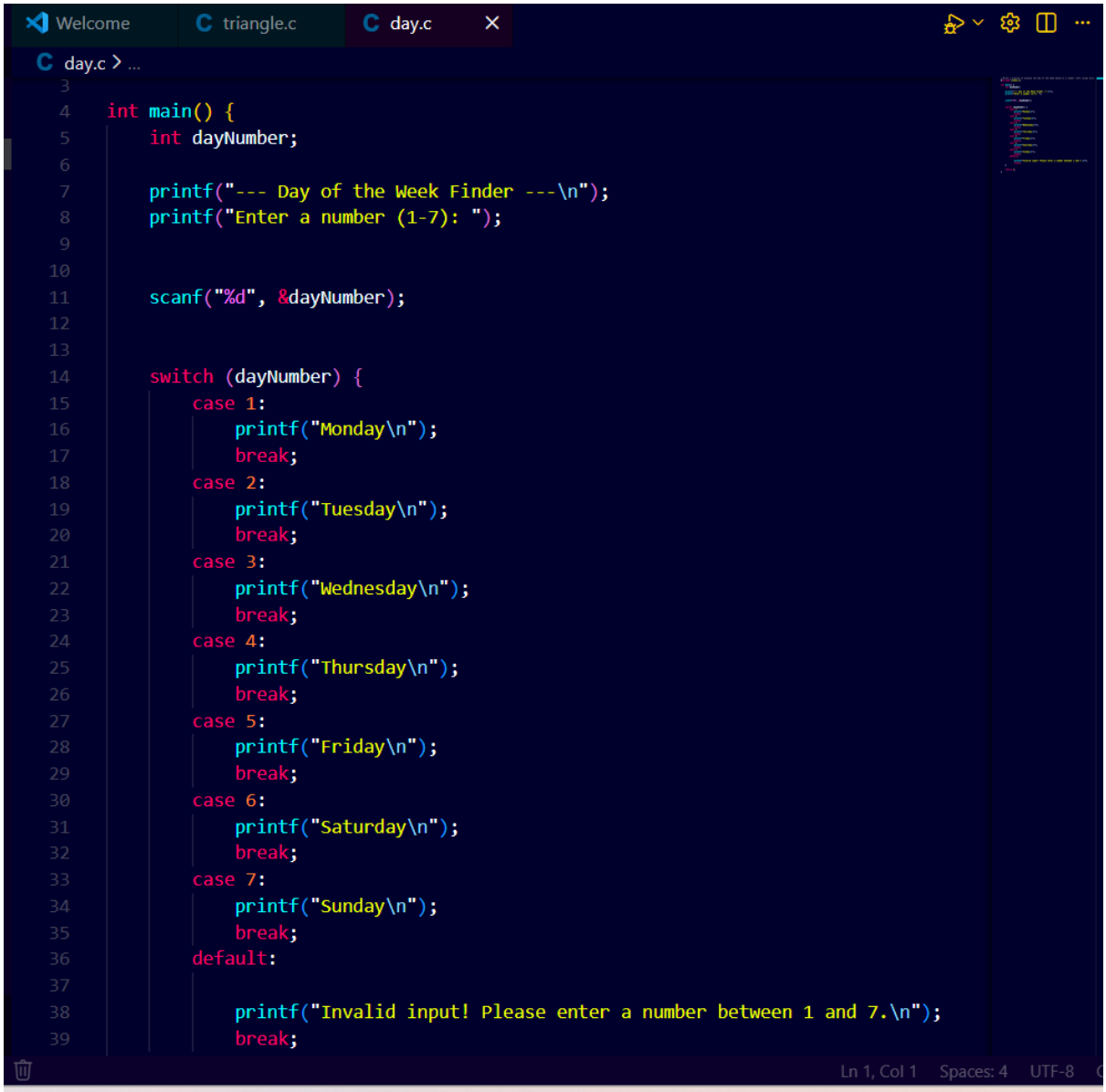
Ln 1, Col 106

DAY-10

Q19 -Write a program to classify a triangle as Equilateral, Isosceles, or Scalene based on its side lengths.



Q20 -Write a program to display the day of the week based on a number (1–7) using switch-case.

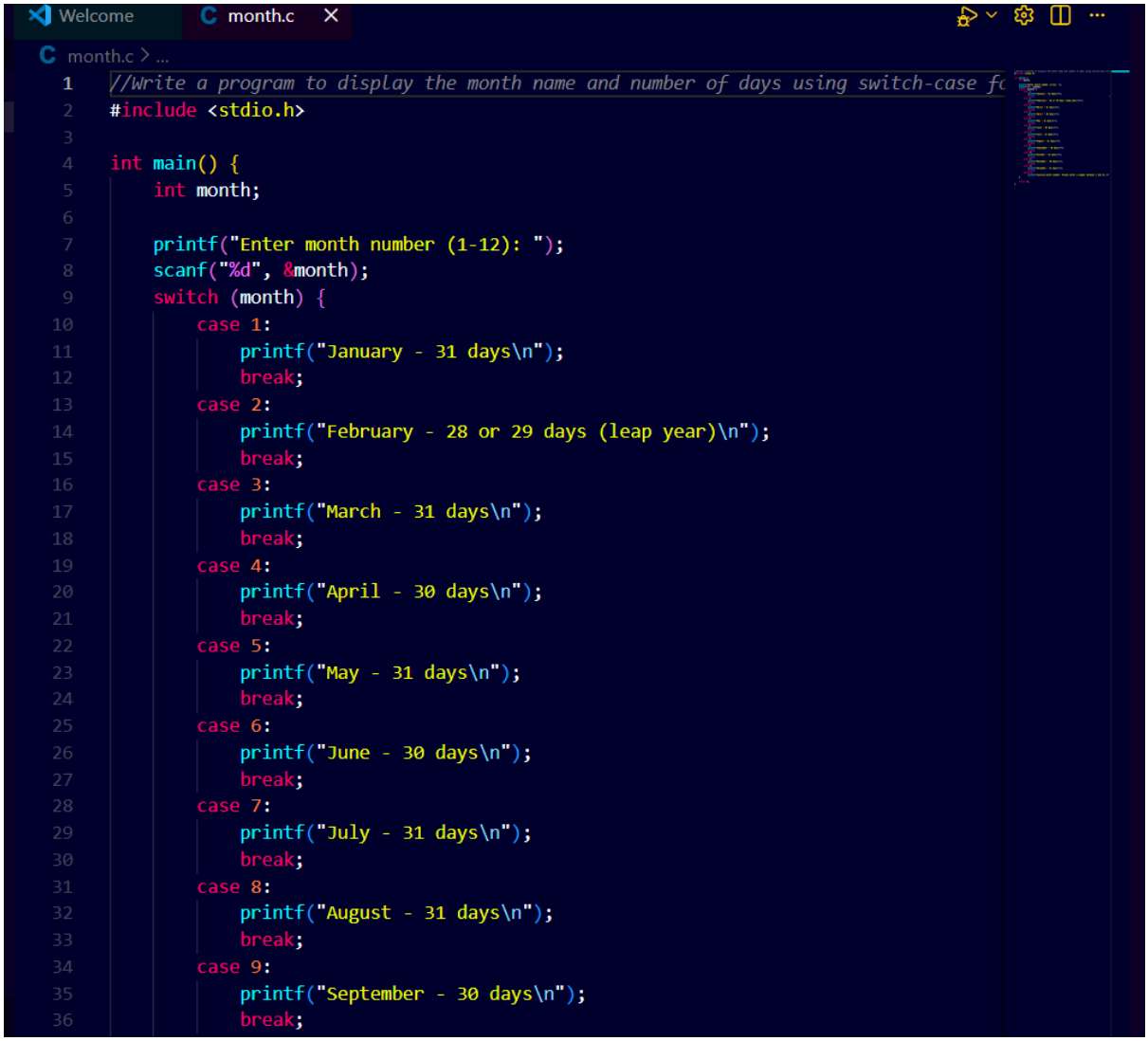


```c
int main() {
    int dayNumber;

    printf("--- Day of the Week Finder ---\n");
    printf("Enter a number (1-7): ");


    scanf("%d", &dayNumber);


    switch (dayNumber) {
        case 1:
            printf("Monday\n");
            break;
        case 2:
            printf("Tuesday\n");
            break;
        case 3:
            printf("Wednesday\n");
            break;
        case 4:
            printf("Thursday\n");
            break;
        case 5:
            printf("Friday\n");
            break;
        case 6:
            printf("Saturday\n");
            break;
        case 7:
            printf("Sunday\n");
            break;
        default:

            printf("Invalid input! Please enter a number between 1 and 7.\n");
            break;
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 10> GCC DAY.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 10> ./A.EXE
--- Day of the Week Finder ---
Enter a number (1-7): 7
Sunday
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 10>
```

DAY-11

Q21 -Write a program to display the month name and number of days using switch-case for a given month number.


Welcome
month.c
month.c > ...
//Write a program to display the month name and number of days using switch-case fc
#include <stdio.h>
int main() {
int month;
printf("Enter month number (1-12): ");
scanf("%d", &month);
switch (month) {
case 1:
printf("January - 31 days\n");
break;
case 2:
printf("February - 28 or 29 days (leap year)\n");
break;
case 3:
printf("March - 31 days\n");
break;
case 4:
printf("April - 30 days\n");
break;
case 5:
printf("May - 31 days\n");
break;
case 6:
printf("June - 30 days\n");
break;
case 7:
printf("July - 31 days\n");
break;
case 8:
printf("August - 31 days\n");
break;
case 9:
printf("September - 30 days\n");
break;

```
            case 10:
                printf("October - 31 days\n");
                break;
            case 11:
                printf("November - 30 days\n");
                break;
            case 12:
                printf("December - 31 days\n");
                break;
            default:
                printf("Invalid month number! Please enter a number between 1 and 12.\r
        }

        return 0;
    }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS        powershell

PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 11> GCC MONTH.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 11> ./A.EXE
Enter month number (1-12): 4
April - 30 days
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 11>
```

Q22 -Write a program to find profit or loss percentage given cost price and selling price.

profit.c > ...

```c
//Write a program to find profit or loss percentage given cost price and selling pr
#include <stdio.h>

int main() {
    float costPrice, sellingPrice, profitLoss, percentage;
    printf("Enter Cost Price: ");
    scanf("%f", &costPrice);

    printf("Enter Selling Price: ");
    scanf("%f", &sellingPrice);


    if (sellingPrice > costPrice) {
        profitLoss = sellingPrice - costPrice;
        percentage = (profitLoss / costPrice) * 100;
        printf("Profit = %.2f\n", profitLoss);
        printf("Profit Percentage = %.2f%%\n", percentage);
    } else if (costPrice > sellingPrice) {
        profitLoss = costPrice - sellingPrice;
        percentage = (profitLoss / costPrice) * 100;
        printf("Loss = %.2f\n", profitLoss);
        printf("Loss Percentage = %.2f%%\n", percentage);
    } else {
        printf("No Profit, No Loss.\n");
    }

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 11> GCC PROFIT.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 11> ./A.EXE
Enter Cost Price: 600
Enter Selling Price: 700
Profit = 100.00
Profit Percentage = 16.67%
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 11>
```

DAY-12

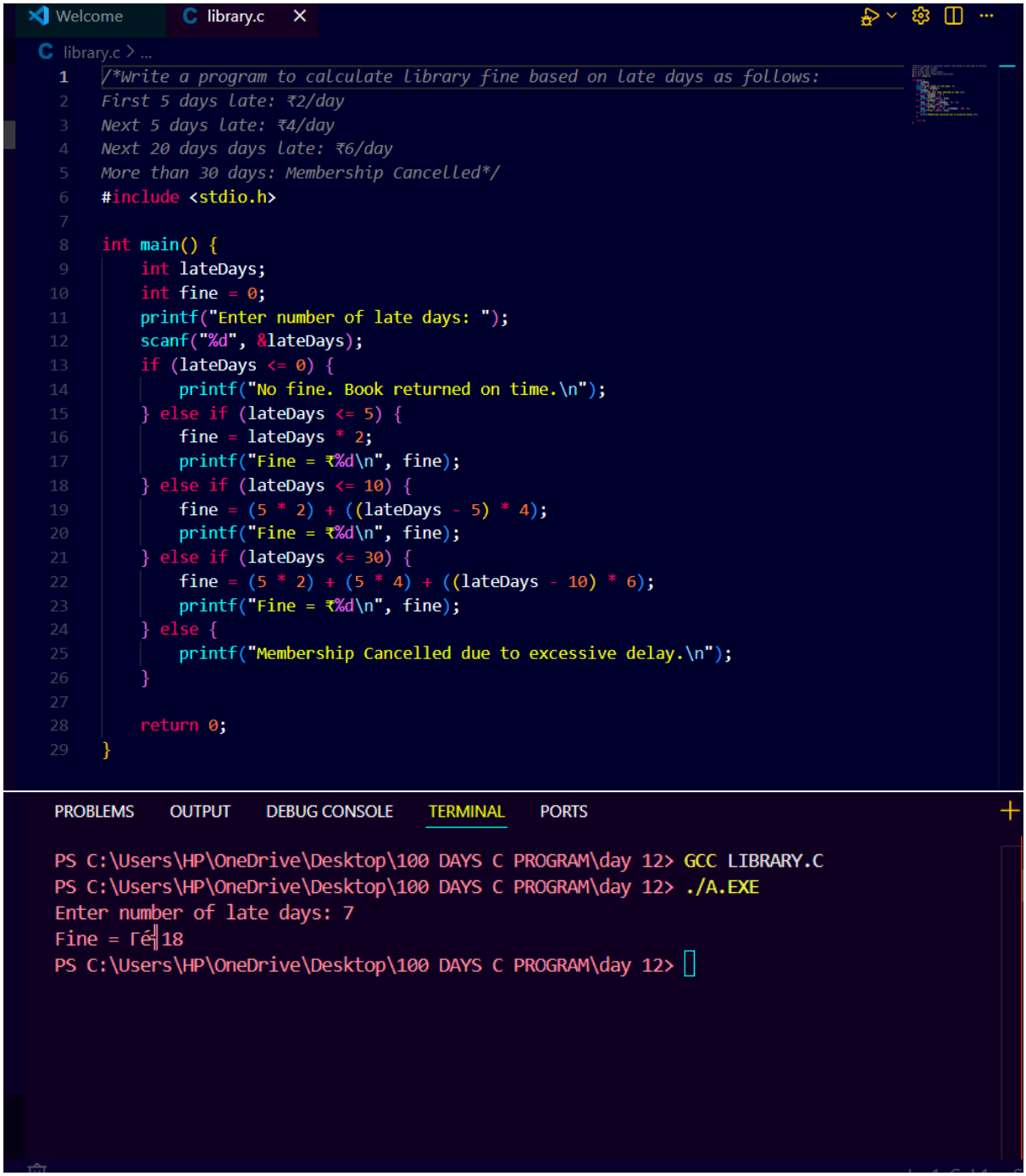
Q23 -Write a program to calculate library fine based on late days as follows:

First 5 days late: ₹2/day

Next 5 days late: ₹4/day

Next 20 days days late: ₹6/day

More than 30 days: Membership Cancelled.



```c
/*Write a program to calculate library fine based on late days as follows:
First 5 days late: ₹2/day
Next 5 days late: ₹4/day
Next 20 days days late: ₹6/day
More than 30 days: Membership Cancelled*/
#include <stdio.h>

int main() {
    int lateDays;
    int fine = 0;
    printf("Enter number of late days: ");
    scanf("%d", &lateDays);
    if (lateDays <= 0) {
        printf("No fine. Book returned on time.\n");
    } else if (lateDays <= 5) {
        fine = lateDays * 2;
        printf("Fine = ₹%d\n", fine);
    } else if (lateDays <= 10) {
        fine = (5 * 2) + ((lateDays - 5) * 4);
        printf("Fine = ₹%d\n", fine);
    } else if (lateDays <= 30) {
        fine = (5 * 2) + (5 * 4) + ((lateDays - 10) * 6);
        printf("Fine = ₹%d\n", fine);
    } else {
        printf("Membership Cancelled due to excessive delay.\n");
    }

    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 12> GCC LIBRARY.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 12> ./A.EXE
Enter number of late days: 7
Fine = Γé¹18
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 12>
```

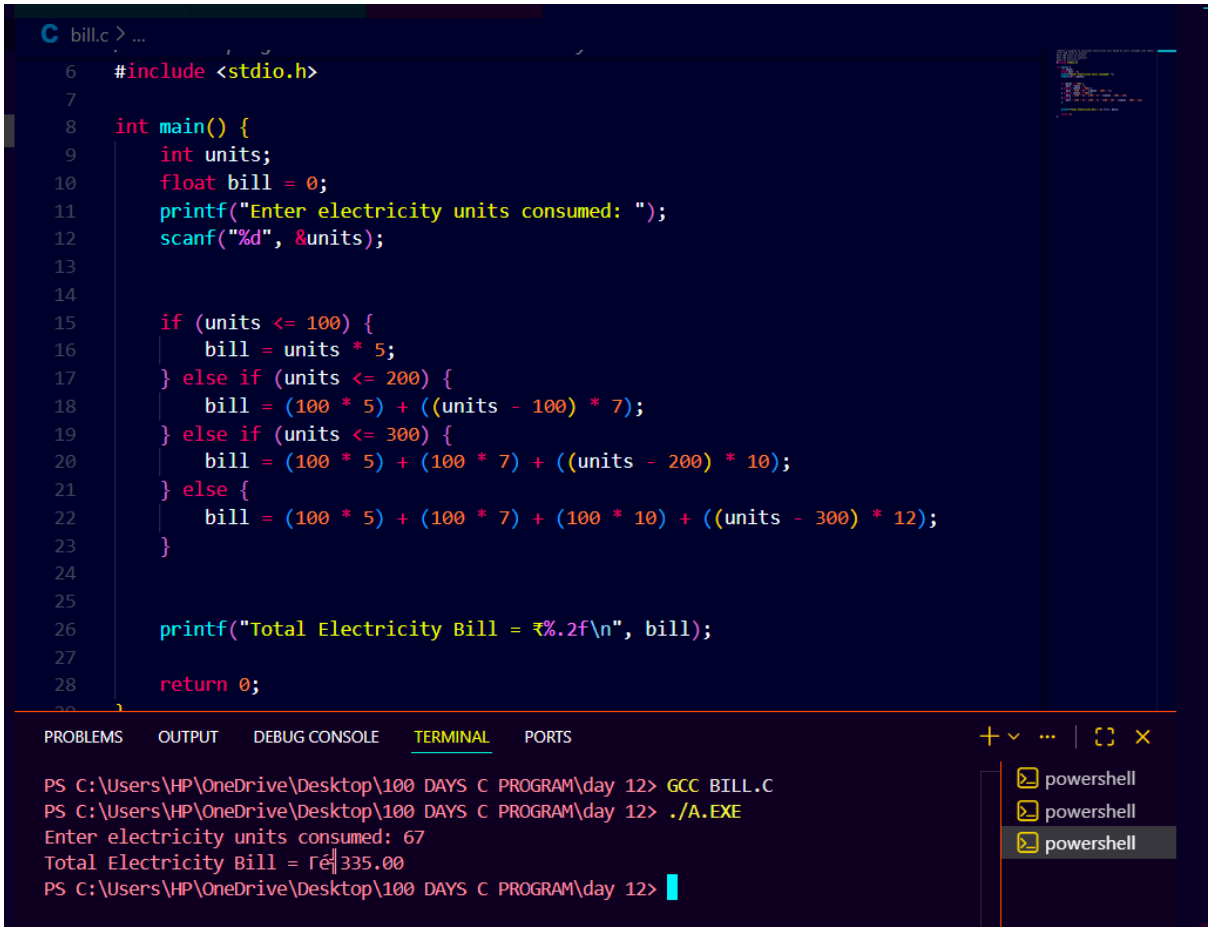
Q24 -Write a program to calculate electricity bill based on units consumed with these rates:

First 100 units at ₹5/unit

Next 100 units at ₹7/unit

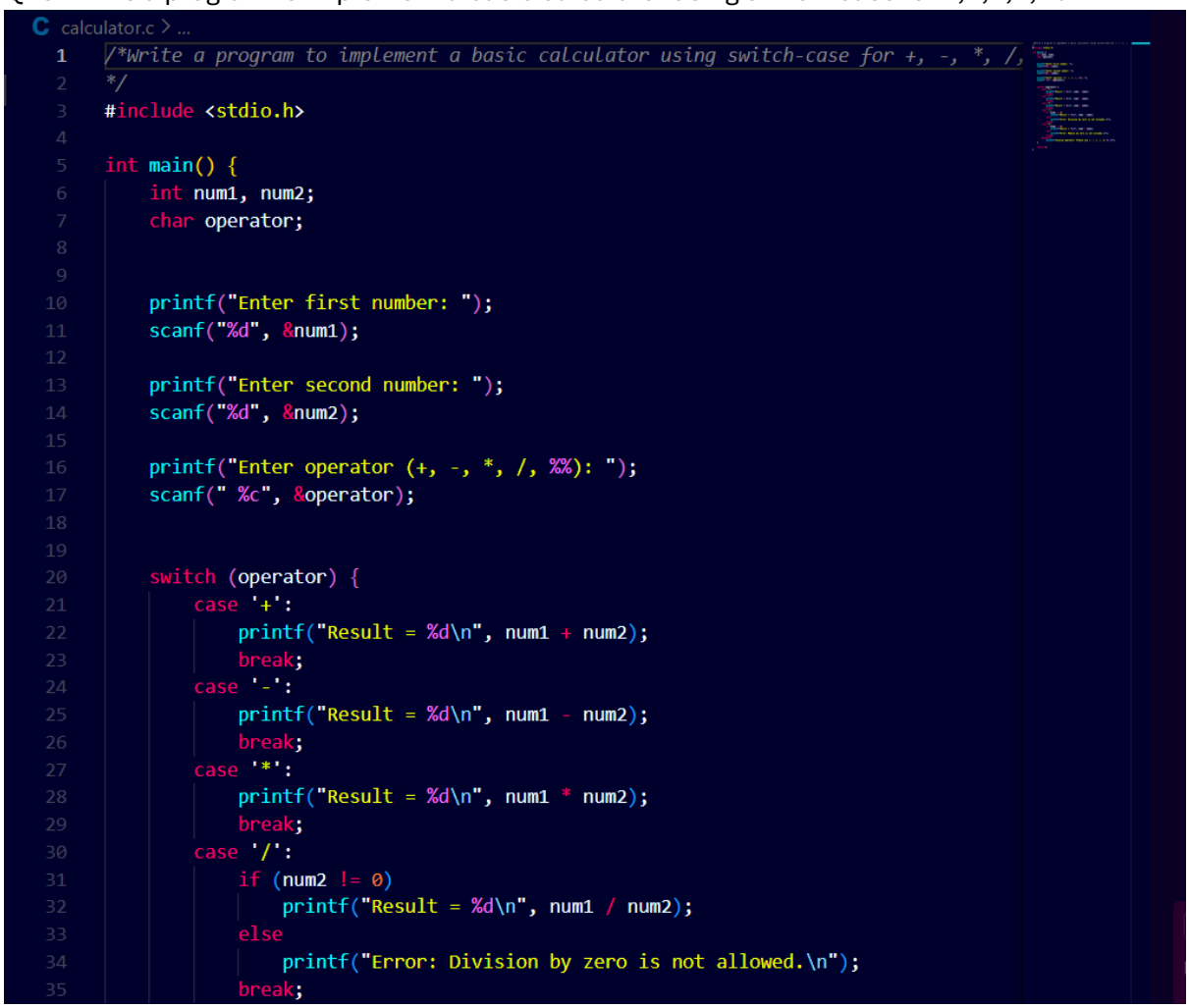
Next 100 units at ₹10/unit

Above at ₹12/unit



DAY-13

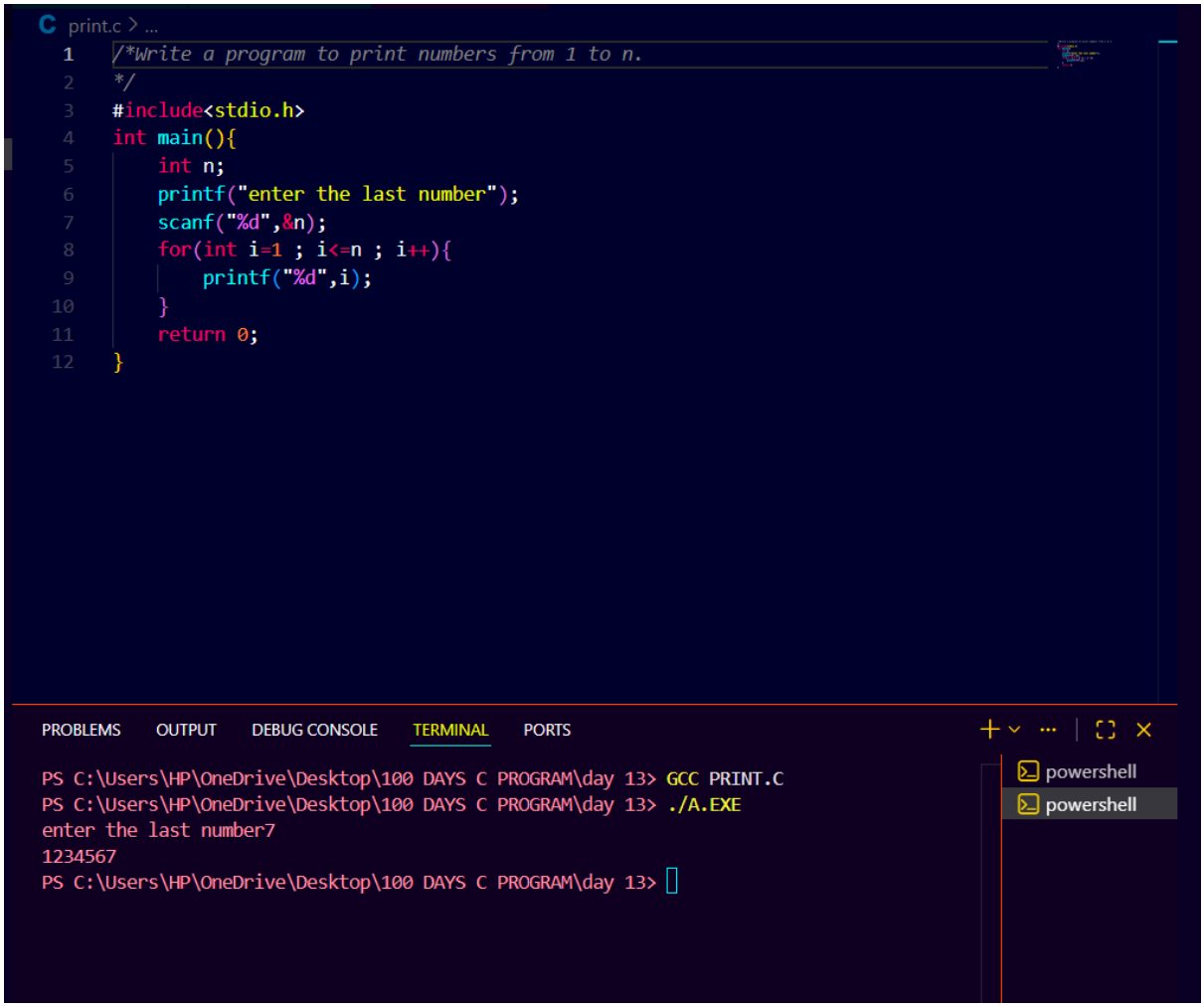
Q25 -Write a program to implement a basic calculator using switch-case for +, -, *, /, %.



calculator.c > ...

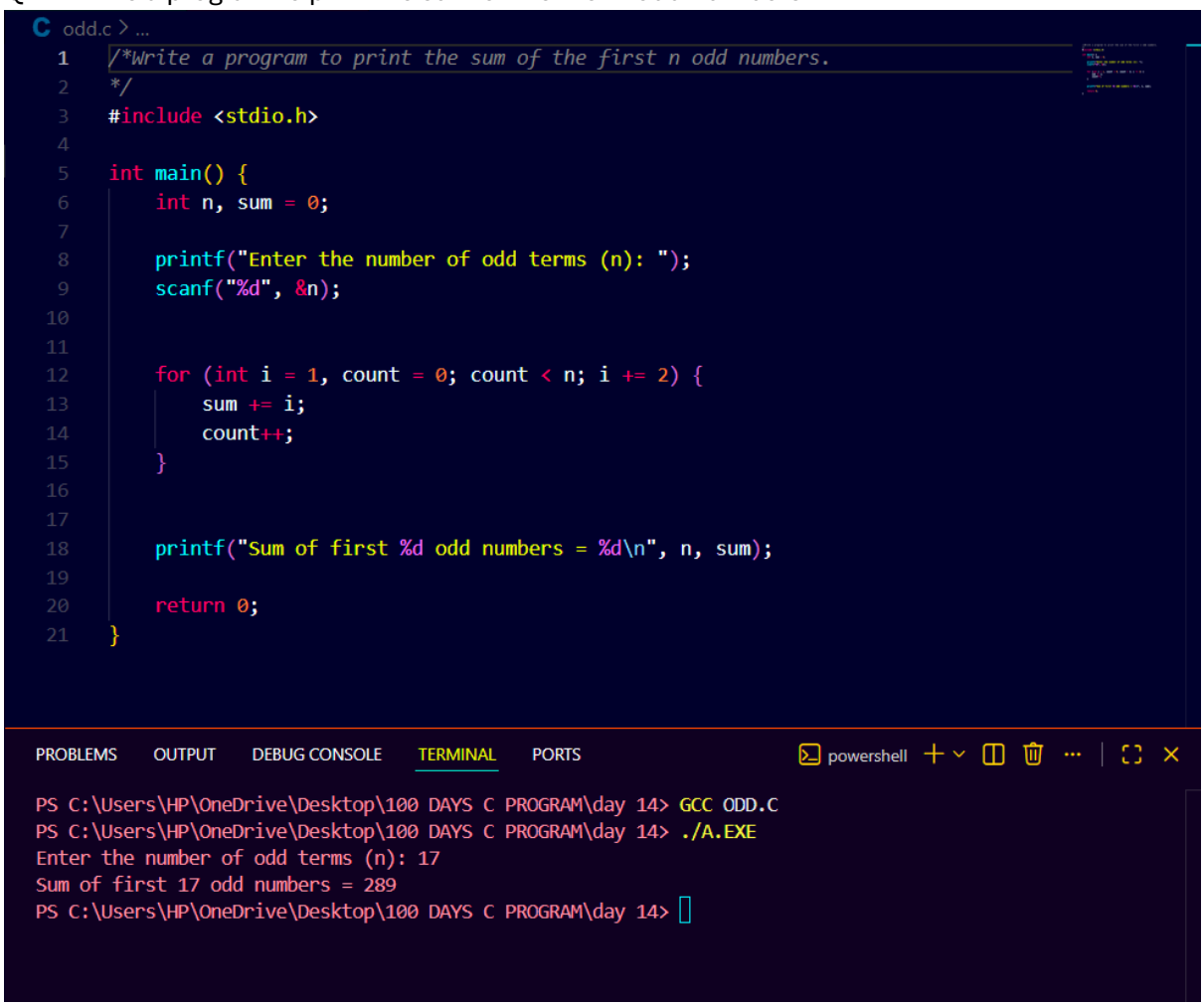
```c
/*Write a program to implement a basic calculator using switch-case for +, -, *, /,
*/
#include <stdio.h>

int main() {
    int num1, num2;
    char operator;


    printf("Enter first number: ");
    scanf("%d", &num1);

    printf("Enter second number: ");
    scanf("%d", &num2);

    printf("Enter operator (+, -, *, /, %%): ");
    scanf(" %c", &operator);


    switch (operator) {
        case '+':
            printf("Result = %d\n", num1 + num2);
            break;
        case '-':
            printf("Result = %d\n", num1 - num2);
            break;
        case '*':
            printf("Result = %d\n", num1 * num2);
            break;
        case '/':
            if (num2 != 0)
                printf("Result = %d\n", num1 / num2);
            else
                printf("Error: Division by zero is not allowed.\n");
            break;
```

Q26-Write a program to print numbers from 1 to n.

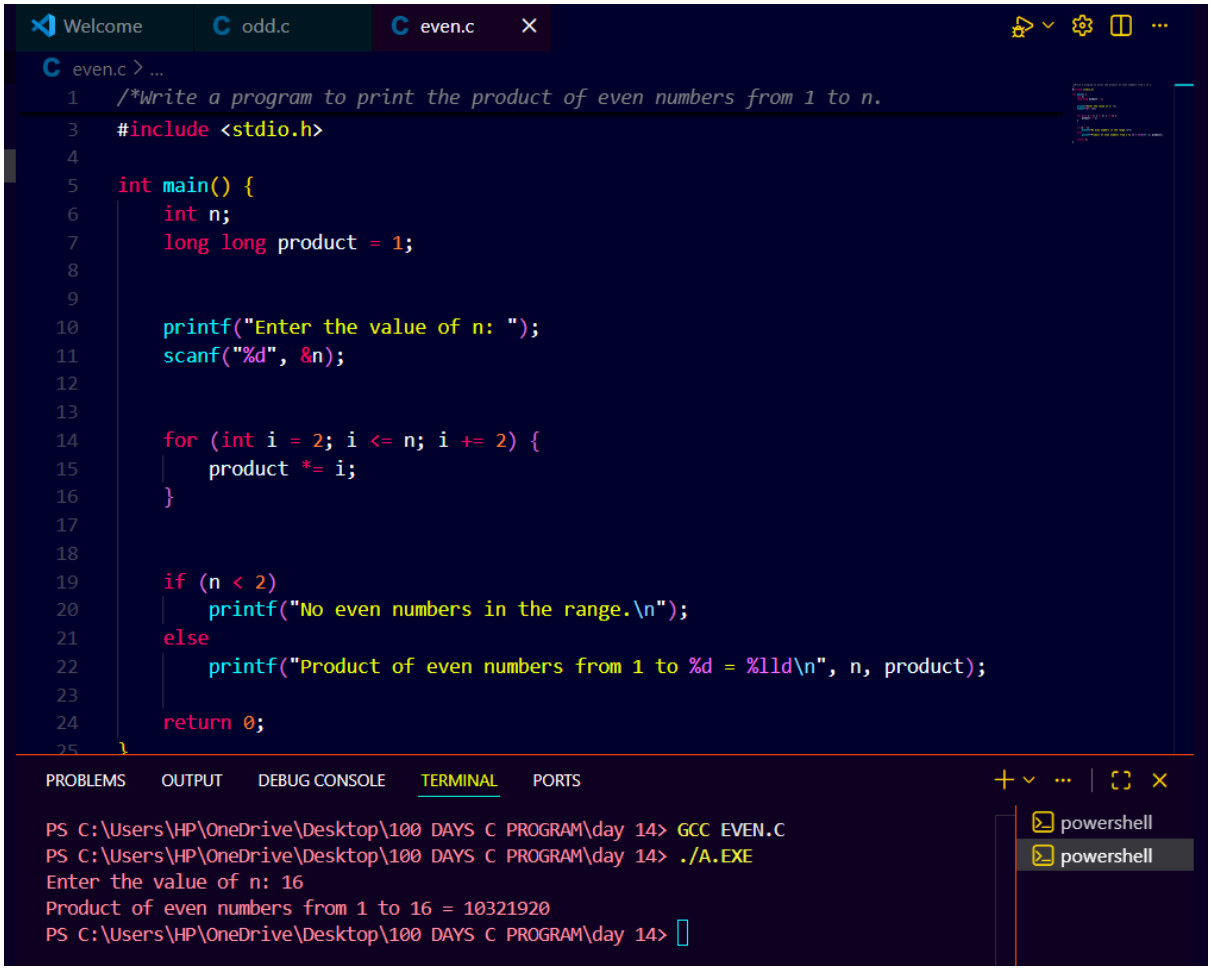


DAY-14

Q27 -Write a program to print the sum of the first n odd numbers.

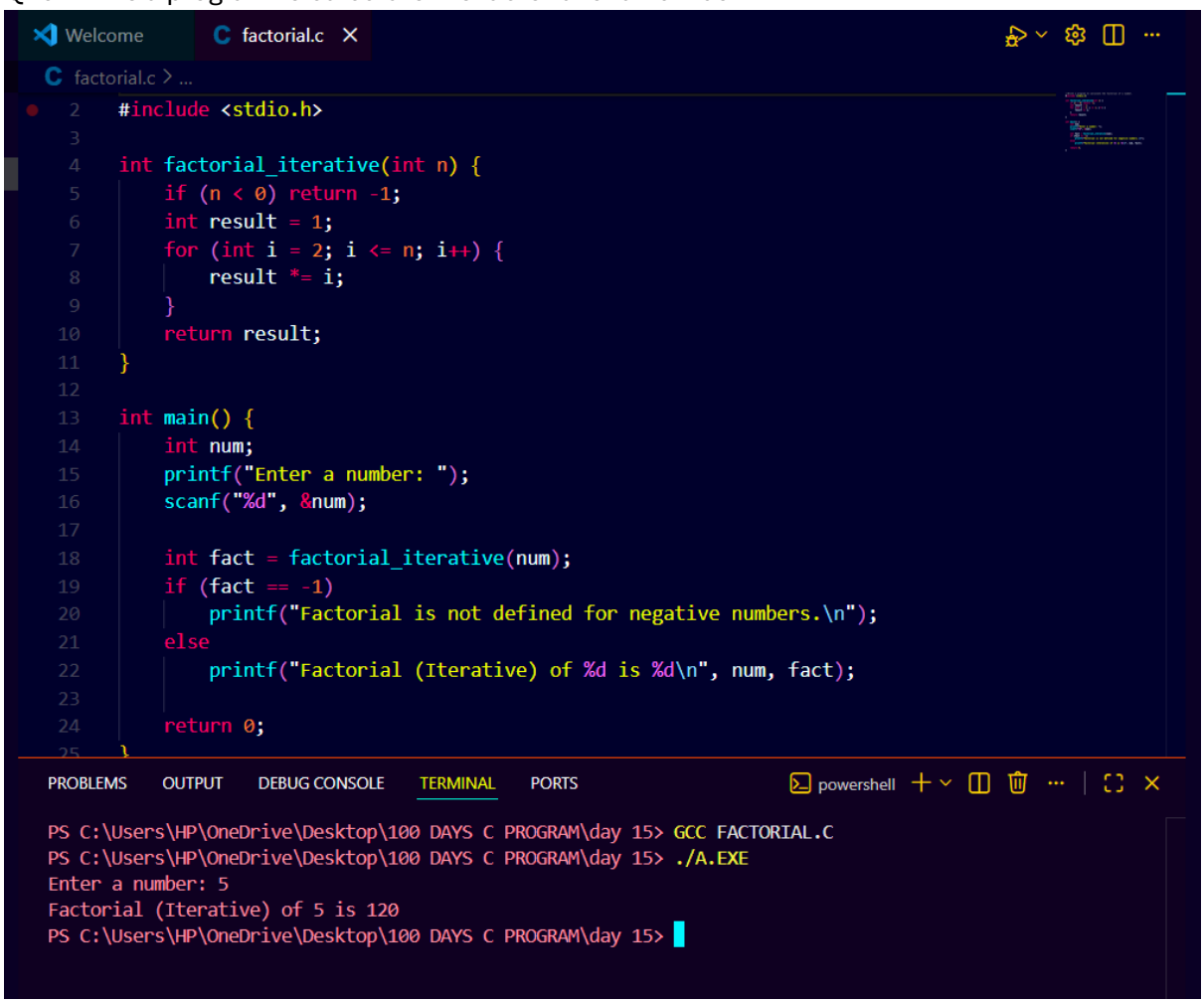


Q28 -Write a program to print the product of even numbers from 1 to n.

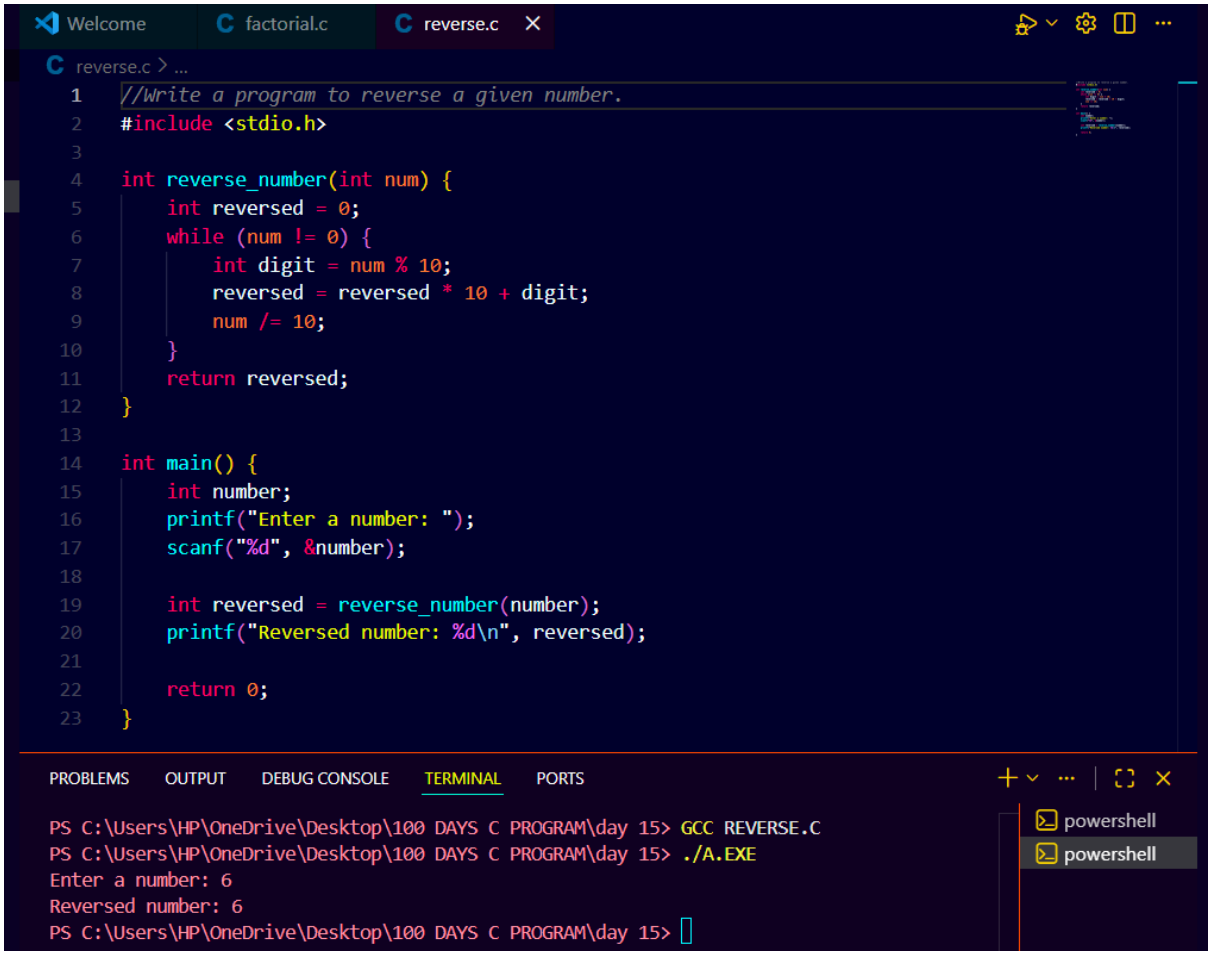


DAY-15

Q29 -Write a program to calculate the factorial of a number.

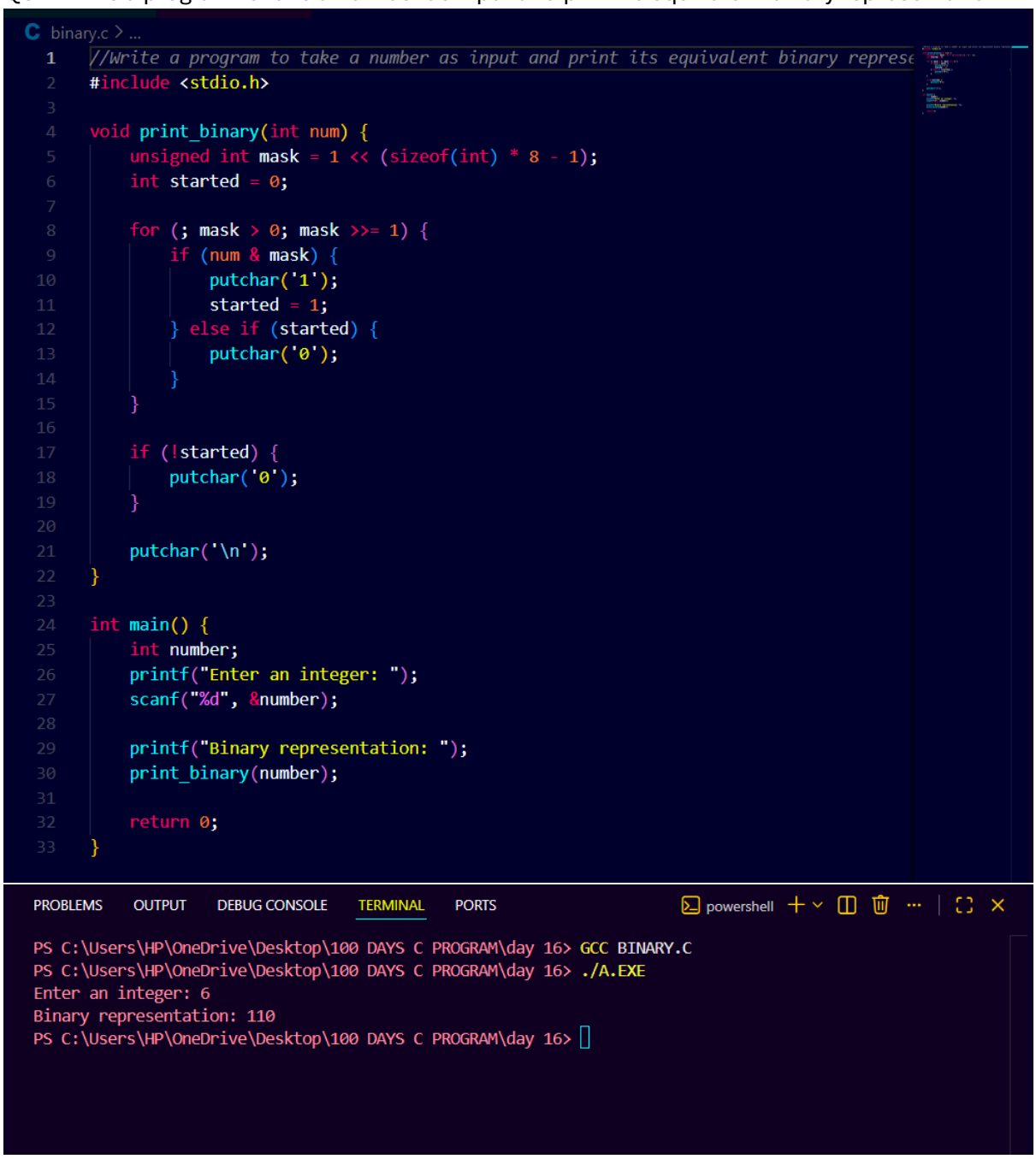


Q30 -Write a program to reverse a given number.



DAY-16

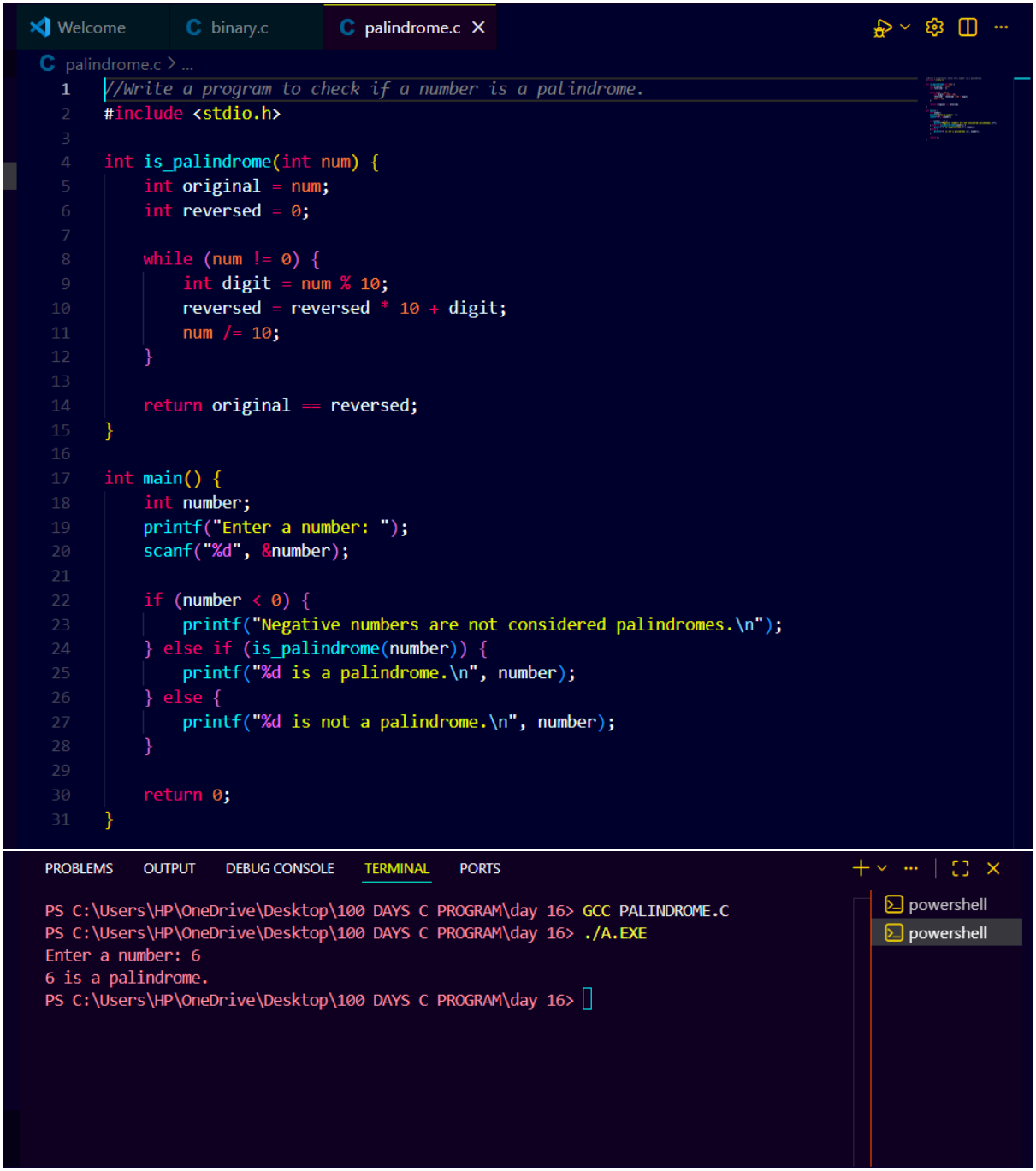
Q31 -Write a program to take a number as input and print its equivalent binary representation.



```c
//Write a program to take a number as input and print its equivalent binary represe
#include <stdio.h>

void print_binary(int num) {
    unsigned int mask = 1 << (sizeof(int) * 8 - 1);
    int started = 0;

    for (; mask > 0; mask >>= 1) {
        if (num & mask) {
            putchar('1');
            started = 1;
        } else if (started) {
            putchar('0');
        }
    }

    if (!started) {
        putchar('0');
    }

    putchar('\n');
}

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

    printf("Binary representation: ");
    print_binary(number);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 16> GCC BINARY.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 16> ./A.EXE
Enter an integer: 6
Binary representation: 110
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 16>
```

Q32 -Write a program to check if a number is a palindrome.

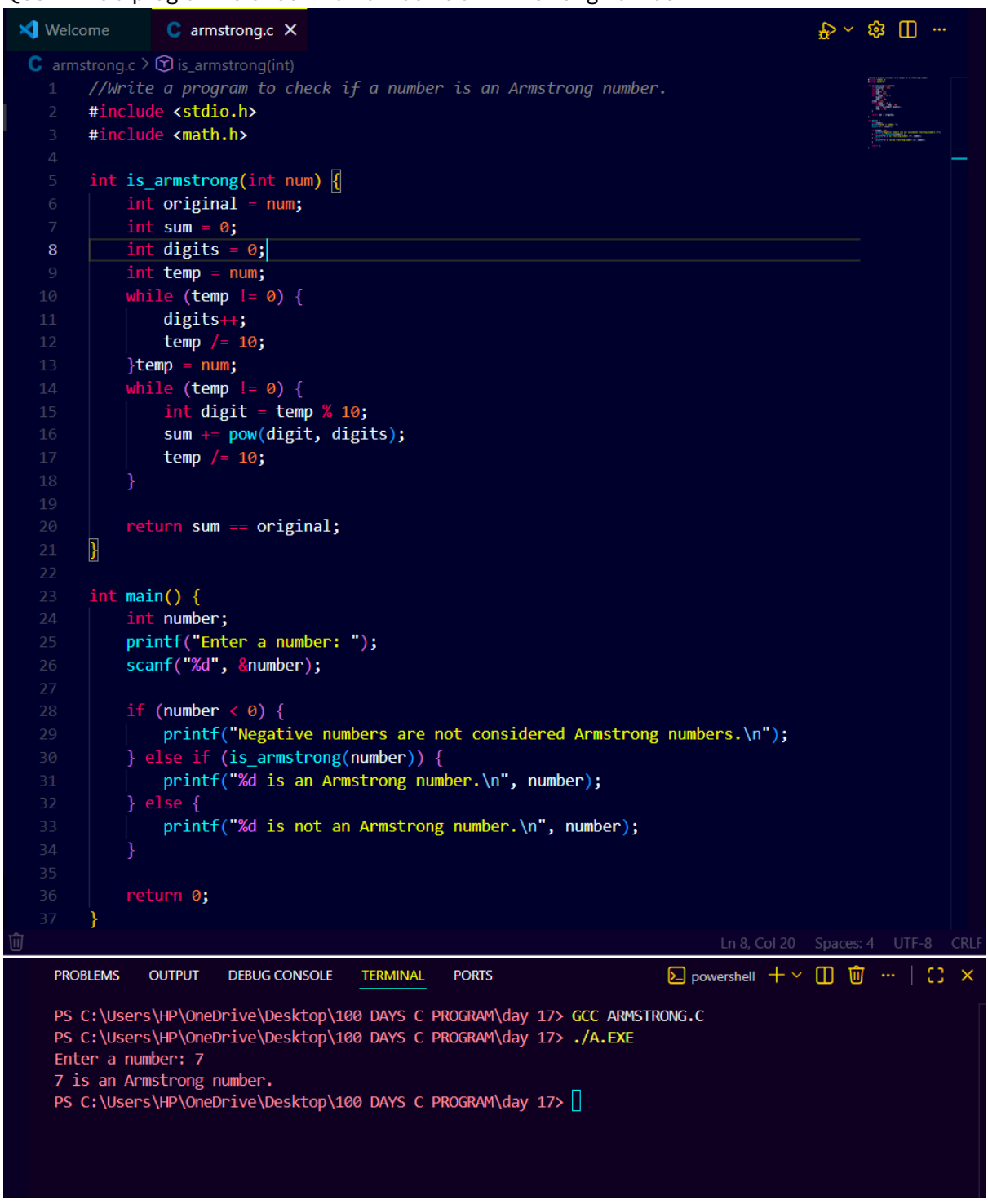


```c
//Write a program to check if a number is a palindrome.
#include <stdio.h>

int is_palindrome(int num) {
    int original = num;
    int reversed = 0;

    while (num != 0) {
        int digit = num % 10;
        reversed = reversed * 10 + digit;
        num /= 10;
    }

    return original == reversed;
}

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

    if (number < 0) {
        printf("Negative numbers are not considered palindromes.\n");
    } else if (is_palindrome(number)) {
        printf("%d is a palindrome.\n", number);
    } else {
        printf("%d is not a palindrome.\n", number);
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 16> GCC PALINDROME.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 16> ./A.EXE
Enter a number: 6
6 is a palindrome.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 16>
```

DAY-17

Q33-Write a program to check if a number is an Armstrong number.



```c
//Write a program to check if a number is an Armstrong number.
#include <stdio.h>
#include <math.h>

int is_armstrong(int num) {
    int original = num;
    int sum = 0;
    int digits = 0;
    int temp = num;
    while (temp != 0) {
        digits++;
        temp /= 10;
    }temp = num;
    while (temp != 0) {
        int digit = temp % 10;
        sum += pow(digit, digits);
        temp /= 10;
    }

    return sum == original;
}

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

    if (number < 0) {
        printf("Negative numbers are not considered Armstrong numbers.\n");
    } else if (is_armstrong(number)) {
        printf("%d is an Armstrong number.\n", number);
    } else {
        printf("%d is not an Armstrong number.\n", number);
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 17> GCC ARMSTRONG.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 17> ./A.EXE
Enter a number: 7
7 is an Armstrong number.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 17>
```

Q34 -Write a program to check if a number is prime.

Welcome armstrong.c prime.c

prime.c > ...

```c
//Write a program to check if a number is prime.
#include <stdio.h>

int is_prime(int num) {
    if (num <= 1) return 0;
    if (num == 2) return 1; // 2 is prime

    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0)
            return 0; // Found a divisor
    }
    return 1; // No divisors found
}

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

    if (is_prime(number))
        printf("%d is a prime number.\n", number);
    else
        printf("%d is not a prime number.\n", number);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 17> GCC PRIME.C
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 17> ./A.EXE
Enter a number: 5
5 is a prime number.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 17>
```

DAY-18

🔁 Q35 (Loops without Arrays/Strings)

Write a program to print all factors of a given number.

factores.c > ...

```c
//Write a program to print all factors of a given number.
#include <stdio.h>

void print_factors(int num) {
    if (num <= 0) {
        printf("Please enter a positive integer.\n");
        return;
    }

    printf("Factors of %d are: ", num);
    for (int i = 1; i <= num; i++) {
        if (num % i == 0) {
            printf("%d ", i);
        }
    }
    printf("\n");
}

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

    print_factors(number);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 18> GCC factores.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 18> ./a.exe
Enter a number: 10
Factors of 10 are: 1 2 5 10
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 18>
```

🔁 Q36 (Loops without Arrays/Strings)

Write a program to find the HCF (GCD) of two numbers.

```c
//Write a program to find the HCF (GCD) of two numbers.
#include <stdio.h>

int find_hcf(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}

int main() {
    int num1, num2;
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    if (num1 == 0 && num2 == 0) {
        printf("HCF is undefined for both numbers being zero.\n");
    } else {
        int hcf = find_hcf(num1, num2);
        printf("HCF of %d and %d is %d\n", num1, num2, hcf);
    }
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 18> GCC HCF.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 18> ./a.exe
Enter two numbers: 10 2
HCF of 10 and 2 is 2
```

DAY-19

🔁 Q37 (Loops without Arrays/Strings)

Write a program to find the LCM of two numbers.

```c
//Write a program to find the LCM of two numbers.
#include <stdio.h>

// Function to find HCF using Euclidean algorithm
int find_hcf(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}

// Function to find LCM
int find_lcm(int a, int b) {
    if (a == 0 || b == 0) return 0; // LCM of 0 is undefined
    int hcf = find_hcf(a, b);
    return (a * b) / hcf;
}

int main() {
    int num1, num2;
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    int lcm = find_lcm(num1, num2);
    if (lcm == 0)
        printf("LCM is undefined for zero input.\n");
    else
        printf("LCM of %d and %d is %d\n", num1, num2, lcm);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 19> GCC LCM.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 19> ./a.exe
Enter two numbers: 80 4
LCM of 80 and 4 is 80
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 19>
```

🔁 Q38 (Loops without Arrays/Strings)

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

sum_digit.c > ...

```c
//Write a program to find the sum of digits of a number.
#include <stdio.h>

int sum_of_digits(int num) {
    int sum = 0;
    num = (num < 0) ? -num : num; // Handle negative numbers

    while (num != 0) {
        sum += num % 10; // Add last digit
        num /= 10;       // Remove last digit
    }

    return sum;
}

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

    int result = sum_of_digits(number);
    printf("Sum of digits: %d\n", result);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 19> GCC sum_digit.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 19> ./a.exe
Enter a number: 145
Sum of digits: 10
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 19>
```

DAY-20

🔁 Q39 (Loops without Arrays/Strings)

Write a program to find the product of odd digits of a number.

```c
//Write a program to find the 1's complement of a binary number and print it.
#include <stdio.h>
#include <string.h>

void ones_complement(char binary[]) {
    for (int i = 0; binary[i] != '\0'; i++) {
        if (binary[i] == '0')
            binary[i] = '1';
        else if (binary[i] == '1')
            binary[i] = '0';
        else {
            printf("Invalid binary digit '%c' found.\n", binary[i]);
            return;
        }
    }
    printf("1's Complement: %s\n", binary);
}

int main() {
    char binary[100];

    printf("Enter a binary number: ");
    scanf("%s", binary);

    ones_complement(binary);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 20> GCC binary_number.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 20> ./a.exe
Enter a binary number: 1010
1's Complement: 0101
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 20>
```

🔁 Q40 (Loops without Arrays/Strings)

Write a program to find the 1’s complement of a binary number and print it.

```c
//Write a program to find the product of odd digits of a number.
#include <stdio.h>

int product_of_odd_digits(int num) {
    int product = 1;
    int has_odd = 0;

    num = (num < 0) ? -num : num; // Handle negative numbers

    while (num != 0) {
        int digit = num % 10;
        if (digit % 2 != 0) {
            product *= digit;
            has_odd = 1;
        }
        num /= 10;
    }

    return has_odd ? product : 0; // Return 0 if no odd digits found
}

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

    int result = product_of_odd_digits(number);
    if (result == 0)
        printf("No odd digits found in the number.\n");
    else
        printf("Product of odd digits: %d\n", result);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 20> GCC product_odd_digit.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 20> ./a.exe
Enter a number: 379
Product of odd digits: 189
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 20>
```

DAY-21

🔁 Q41 (Loops without Arrays/Strings)

Write a program to swap the first and last digit of a number.

```c
//Write a program to swap the first and last digit of a number.
#include <stdio.h>
#include <math.h>

int swap_first_last(int num) {
    if (num < 10) return num; // Single-digit numbers remain unchanged

    int digits = (int)log10(num); // Total digits - 1
    int first = num / (int)pow(10, digits); // First digit
    int last = num % 10;                    // Last digit

    // Remove first and last digits
    int middle = (num % (int)pow(10, digits)) / 10;

    // Reconstruct number with swapped digits
    int swapped = last * (int)pow(10, digits) + middle * 10 + first;
    return swapped;
}

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

    if (number < 0) {
        printf("Negative numbers are not supported.\n");
    } else {
        int result = swap_first_last(number);
        printf("Number after swapping first and last digit: %d\n", result);
    }

    return 0;
}
```

🔁 Q42 (Loops without Arrays/Strings)

Write a program to check if a number is a perfect number.

```c
//Write a program to check if a number is a perfect number.
#include <stdio.h>

int is_perfect(int num) {
    if (num <= 0) return 0;

    int sum = 0;
    for (int i = 1; i <= num / 2; i++) {
        if (num % i == 0)
            sum += i;
    }

    return sum == num;
}

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

    if (is_perfect(number))
        printf("%d is a perfect number.\n", number);
    else
        printf("%d is not a perfect number.\n", number);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 21> GCC perfact_number.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 21> ./a.exe
Enter a number: 123
123 is not a perfect number.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 21>
```

DAY-22

🔁 Q43 (Loops without Arrays/Strings)

Write a program to check if a number is a strong number.

```c
//Write a program to check if a number is a strong number.
#include <stdio.h>

// Function to calculate factorial of a digit
int factorial(int n) {
    int fact = 1;
    for (int i = 2; i <= n; i++)
        fact *= i;
    return fact;
}

// Function to check if a number is strong
int is_strong(int num) {
    int original = num;
    int sum = 0;

    while (num != 0) {
        int digit = num % 10;
        sum += factorial(digit);
        num /= 10;
    }

    return sum == original;
}

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

    if (number < 0) {
        printf("Negative numbers are not considered strong numbers.\n");
    } else if (is_strong(number)) {
        printf("%d is a strong number.\n", number);
    } else {
        printf("%d is not a strong number.\n", number);
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 22> ./a.exe
Enter a number: 121
121 is not a strong number.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 22>
```

🔁 Q44 (Loops without Arrays/Strings)

Write a program to find the sum of the series: 1 + 3/4 + 5/6 + 7/8 + ... up to n terms.

```c
//Write a program to find the sum of the series: 1 + 3/4 + 5/6 + 7/8 + … up
#include <stdio.h>

int main() {
    int n;
    float sum = 0.0;
    int numerator = 1;
    int denominator = 2;

    printf("Enter number of terms (n): ");
    scanf("%d", &n);

    for (int i = 1; i <= n; i++) {
        sum += (float)numerator / denominator;
        numerator += 2;
        denominator += 2;
    }

    printf("Sum of the series up to %d terms is: %.4f\n", n, sum);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 22> GCC sum.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 22> ./a.exe
Enter number of terms (n): 10
Sum of the series up to 10 terms is: 8.5355
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 22>
```

DAY-23

🔁 Q45 (Loops without Arrays/Strings)

Write a program to find the sum of the series: 2/3 + 4/7 + 6/11 + 8/15 + ... up to n terms.

```c
//Write a program to find the sum of the series: 2/3 + 4/7 + 6/11 + 8/15 + ... up t
#include <stdio.h>

int main() {
    int n;
    float sum = 0.0;
    int numerator = 2;
    int denominator = 3;

    printf("Enter number of terms (n): ");
    scanf("%d", &n);

    for (int i = 1; i <= n; i++) {
        sum += (float)numerator / denominator;
        numerator += 2;
        denominator += 4;
    }

    printf("Sum of the series up to %d terms is: %.4f\n", n, sum);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 23> GCC sum.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 23> ./a.exe
Enter number of terms (n): 5
Sum of the series up to 5 terms is: 2.8432
```

🔄 Q46 (Nested Loops without Arrays/Strings)

Write a program to print the following pattern:

*****

*****

*****

*****

```c
//Write a program to print the following pattern:
//*****
//*****
//*****
//*****
//*****
#include <stdio.h>

int main() {
    for (int i = 1; i <= 5; i++) {          // Loop for 5 rows
        for (int j = 1; j <= 5; j++) {      // Loop for 5 columns
            printf("*");
        }
        printf("\n");                       // Move to next line after each row
    }
    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 23> GCC pattern.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 23> ./a.exe
*****
*****
*****
*****
*****
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 23>
```

DAY-24

🔁 Q47 (Nested Loops without Arrays/Strings)

Write a program to print the following pattern:

*

**

***

****

```c
/*Write a program to print the following pattern:
*
**
***
****
***** */
#include <stdio.h>

int main() {
    for (int i = 1; i <= 5; i++) {          // Loop for 5 rows
        for (int j = 1; j <= i; j++) {      // Print i stars in each row
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 24> GCC pattern.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 24> ./a.exe
*
**
***
****
*****
```

🔁 Q48 (Nested Loops without Arrays/Strings)

Write a program to print the following pattern:

1

12

123

1234

12345

```c
/* Write a program to print the following pattern:
1
12
123
1234
12345
*/
#include <stdio.h>

int main() {
    for (int i = 1; i <= 5; i++) {          // Loop for 5 rows
        for (int j = 1; j <= i; j++) {      // Print numbers from 1 to i
            printf("%d", j);
        }
        printf("\n");                       // Move to next line
    }
    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 24> GCC p.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 24> ./a.exe
1
12
123
1234
12345
```

DAY-25

🔄 Q49 (Nested Loops without Arrays/Strings)

Write a program to print the following pattern:

5

45

345

2345

12345

```c
/*Write a program to print the following pattern:
5
45
345
2345
12345 */
#include <stdio.h>

int main() {
    for (int i = 5; i >= 1; i--) {          // Start from 5 down to 1
        for (int j = i; j <= 5; j++) {      // Print from i up to 5
            printf("%d", j);
        }
        printf("\n");                       // Move to next line
    }
    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 25> GCC 25.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 25> ./a.exe
5
45
345
2345
12345
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 25>
```

🔁 Q50 (Nested Loops without Arrays/Strings)

Write a program to print the following pattern:

*****

 ****

  ***

   **

    *

```c
/* Write a program to print the following pattern:
*****
 ****
  ***
   **
    *
    */
   #include <stdio.h>

int main() {
    int rows = 5;

    for (int i = 0; i < rows; i++) {
        // Print leading spaces
        for (int space = 0; space < i; space++) {
            printf(" ");
        }

        // Print stars
        for (int star = 0; star < rows - i; star++) {
            printf("*");
        }

        printf("\n"); // Move to next line
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 25> GCC 25p.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 25> ./a.exe
*****
 ****
  ***
   **
    *
```

DAY-26

🔁 Q51 (Nested Loops without Arrays/Strings)

Write a program to print the following pattern:

5

45

345

2345

12345

```c
/*Write a program to print the following pattern:
    5
   45
  345
 2345
12345 */
#include <stdio.h>

int main() {
    int rows = 5;

    for (int i = rows; i >= 1; i--) {
        // Print leading spaces
        for (int space = 1; space < i; space++) {
            printf(" ");
        }

        // Print numbers from i to 5
        for (int num = i; num <= rows; num++) {
            printf("%d", num);
        }

        printf("\n"); // Move to next line
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 26> GCC 26.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 26> ./a.exe
    5
   45
  345
 2345
12345
```

🔁 Q52 (Nested Loops without Arrays/Strings)

Write a program to print the following pattern:

*

*

*

*

*

*

*

*

*

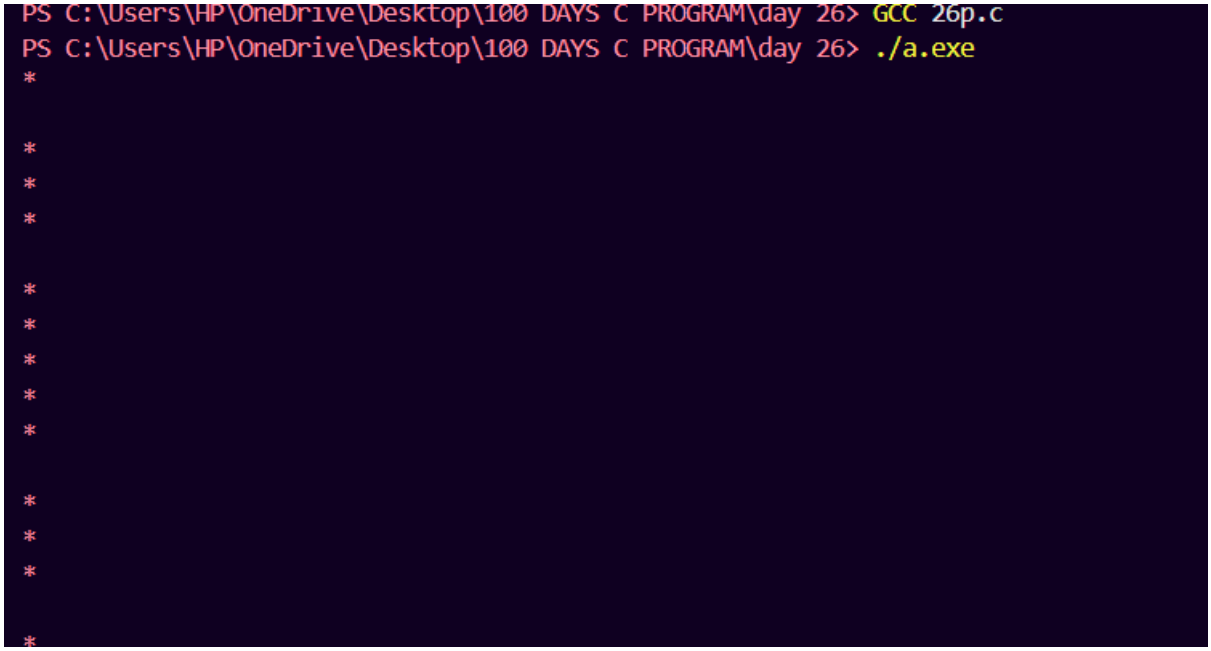
*

*

*

*

```c
#include <stdio.h>

int main() {
    // Array representing number of stars in each group
    int stars[] = {1, 3, 5, 3, 1};
    int groups = sizeof(stars) / sizeof(stars[0]);

    for (int i = 0; i < groups; i++) {
        for (int j = 0; j < stars[i]; j++) {
            printf("*\n");
        }
        if (i != groups - 1) {
            printf("\n"); // Print a blank line between groups
        }
    }

    return 0;
}
```



DAY-27

🔄 Q53 (Nested Loops without Arrays/Strings)

Write a program to print the following pattern:

*

***

*****

****

****

****

*****

***

*

```c
/*Write a program to print the following pattern:
*
***
*****
*******
*********
*******
*****
***
*
*/
#include <stdio.h>

int main() {
    int rows = 5;

    // Upper half
    for (int i = 1; i <= rows; i++) {
        int stars = 2 * i - 1;
        for (int j = 1; j <= stars; j++) {
            printf("*");
        }
        printf("\n");
    }

    // Lower half
    for (int i = rows - 1; i >= 1; i--) {
        int stars = 2 * i - 1;
        for (int j = 1; j <= stars; j++) {
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 27> GCC 27.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 27> ./a.exe
*
***
*****
*******
*********
*******
*****
***
*
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 27>
```

Q54 (Nested Loops without Arrays/Strings)

```c
/*Write a program to print the following pattern:

    *
   ***
  *****
 *******
  *****
   ***
    *
    * */
 #include <stdio.h>

int main() {
    int rows = 4;

    // Upper pyramid
    for (int i = 1; i <= rows; i++) {
        for (int space = 1; space <= rows - i; space++)
            printf(" ");
        for (int star = 1; star <= 2 * i - 1; star++)
            printf("*");
        printf("\n");
    }

    // Lower inverted pyramid
    for (int i = rows - 1; i >= 1; i--) {
        for (int space = 1; space <= rows - i; space++)
            printf(" ");
        for (int star = 1; star <= 2 * i - 1; star++)
            printf("*");
        printf("\n");
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 27> GCC 27p.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 27> ./a.exe
   *
  ***
 *****
*******
 *****
  ***
   *
```

Write a program to print the following pattern:

```
  *
 ***
*****
*******
 *****
  ***
   *
```

DAY-28

🔁 Q55 (Nested Loops without Arrays/Strings)

Write a program to print all the prime numbers from 1 to n.

```c
/*
Write a program to print all the prime numbers from 1 to n.
*/
#include <stdio.h>

int is_prime(int num) {
    if (num <= 1) return 0;
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0)
            return 0;
    }
    return 1;
}

int main() {
    int n;
    printf("Enter the value of n: ");
    scanf("%d", &n);

    printf("Prime numbers from 1 to %d are:\n", n);
    for (int i = 1; i <= n; i++) {
        if (is_prime(i))
            printf("%d ", i);
    }
    printf("\n");

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 28> gcc prime.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 28> ./a.exe
Enter the value of n: 7
Prime numbers from 1 to 7 are:
2 3 5 7
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 28>
```

🔢 Q56 (Arrays (1D))

Read and print elements of a one-dimensional array.

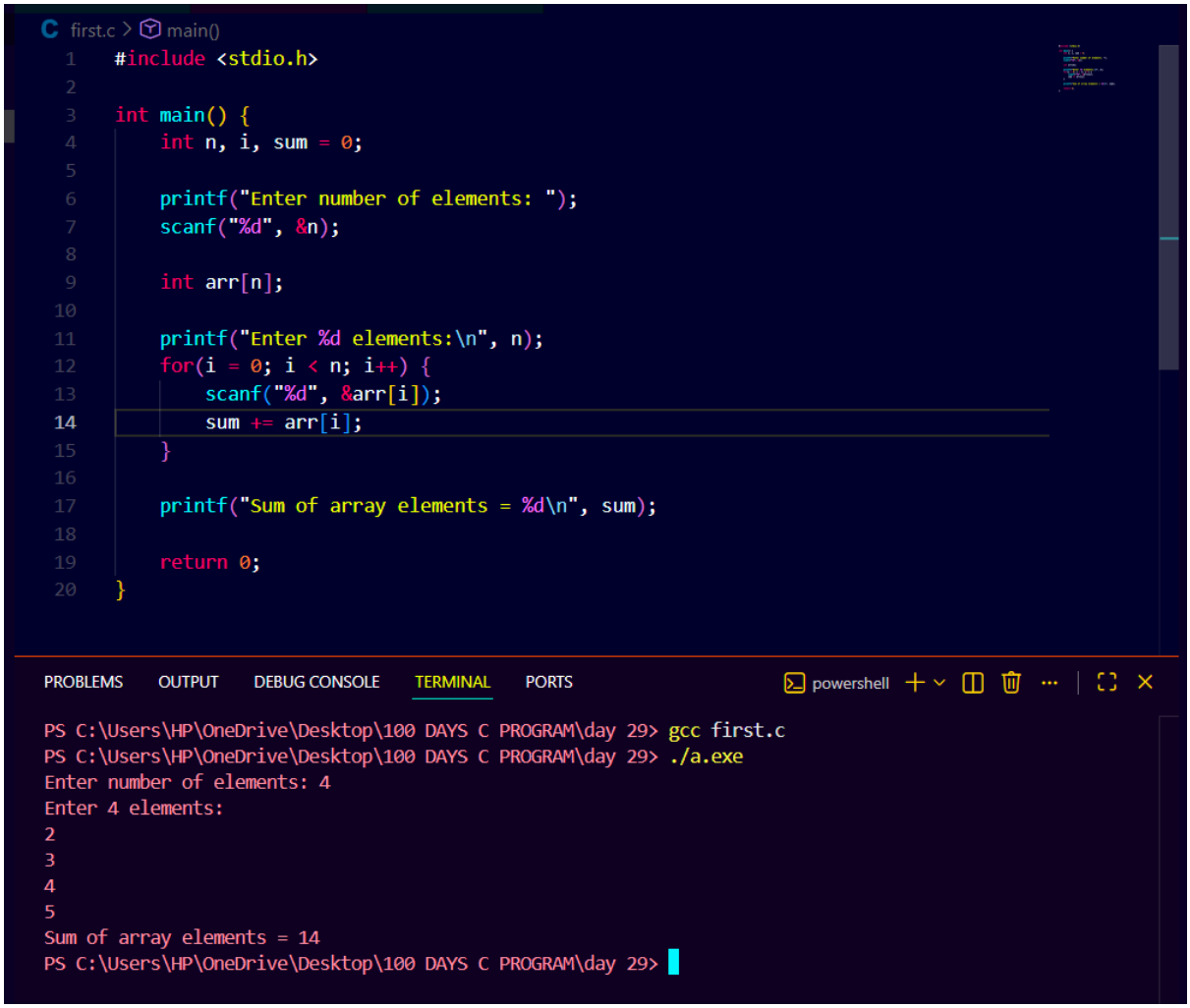
```c
/*Read and print elements of a one-dimensional array.
*/
#include <stdio.h>

int main() {
    int n;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n]; // Declare array of size n

    // Read array elements
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Print array elements
    printf("The elements of the array are:\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 28> GCC array.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 28> ./a.exe
Enter the number of elements: 6
Enter 6 elements:
7
5
8
0
3
2
The elements of the array are:
7 5 8 0 3 2
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 28>
```

DAY-29

🔢 Q57 (Arrays (1D))

Find the sum of array elements.



🔢 Q58 (Arrays (1D))

Find the maximum and minimum element in an array.

second.c > main()

```c
#include <stdio.h>

int main() {
    int n, i;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n];

    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int max = arr[0];
    int min = arr[0];

    for(i = 1; i < n; i++) {
        if(arr[i] > max)
            max = arr[i];
        if(arr[i] < min)
            min = arr[i];
    }

    printf("Maximum element = %d\n", max);
    printf("Minimum element = %d\n", min);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 29> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 29> ./a.exe
Enter number of elements: 3
Enter 3 elements:
4
6
9
Maximum element = 9
Minimum element = 4
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 29>
```

DAY-30

🔢 Q59 (Arrays (1D))

Count even and odd numbers in an array.

first.c > main()

```c
#include <stdio.h>

int main() {
    int n, i;
    int evenCount = 0, oddCount = 0;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n];

    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    for(i = 0; i < n; i++) {
        if(arr[i] % 2 == 0)
            evenCount++;
        else
            oddCount++;
    }

    printf("Count of even numbers = %d\n", evenCount);
    printf("Count of odd numbers = %d\n", oddCount);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 30> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 30> ./a.exe
Enter number of elements: 3
Enter 3 elements:
1
2
3
Count of even numbers = 1
Count of odd numbers = 2
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 30>
```

🔢 Q60 (Arrays (1D))

Count positive, negative, and zero elements in an array.

```c
//Count positive, negative, and zero elements in an array.
#include <stdio.h>

int main() {
    int n, i;
    int posCount = 0, negCount = 0, zeroCount = 0;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n];

    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    for(i = 0; i < n; i++) {
        if(arr[i] > 0)
            posCount++;
        else if(arr[i] < 0)
            negCount++;
        else
            zeroCount++;
    }

    printf("Count of positive numbers = %d\n", posCount);
    printf("Count of negative numbers = %d\n", negCount);
    printf("Count of zeros = %d\n", zeroCount);

    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 30> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 30> ./a.exe
Enter number of elements: 3
Enter 3 elements:
1
2
3
Count of positive numbers = 3
Count of negative numbers = 0
Count of zeros = 0
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 30>
```

DAY-31

🔢 Q61 (Arrays (1D))

Search for an element in an array using linear search.

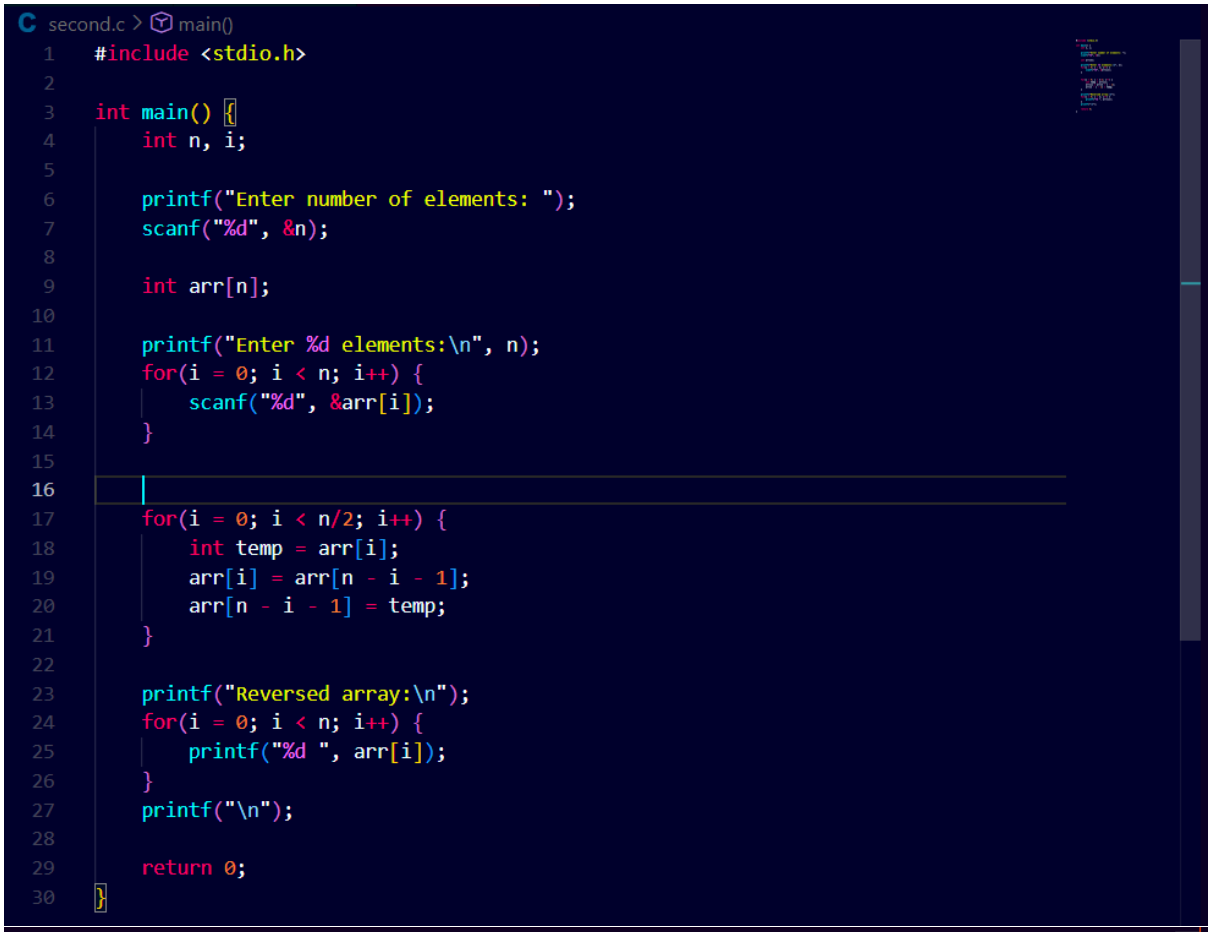
first.c > main()

```c
#include <stdio.h>

int main() {
    int n, i, key, found = 0;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n];

    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Enter element to search: ");
    scanf("%d", &key);

    for(i = 0; i < n; i++) {
        if(arr[i] == key) {
            printf("Element %d found at position %d (index %d)\n", key, i+1, i);
            found = 1;
            break;  // stop after finding first occurrence
        }
    }

    if(!found)
        printf("Element %d not found in the array.\n", key);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 31> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 31> ./a.exe
Enter number of elements: 3
Enter 3 elements:
1
2
3
Enter element to search: 2
Element 2 found at position 2 (index 1)
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 31>
```

🔢 Q62 (Arrays (1D))

Reverse an array without taking extra space.



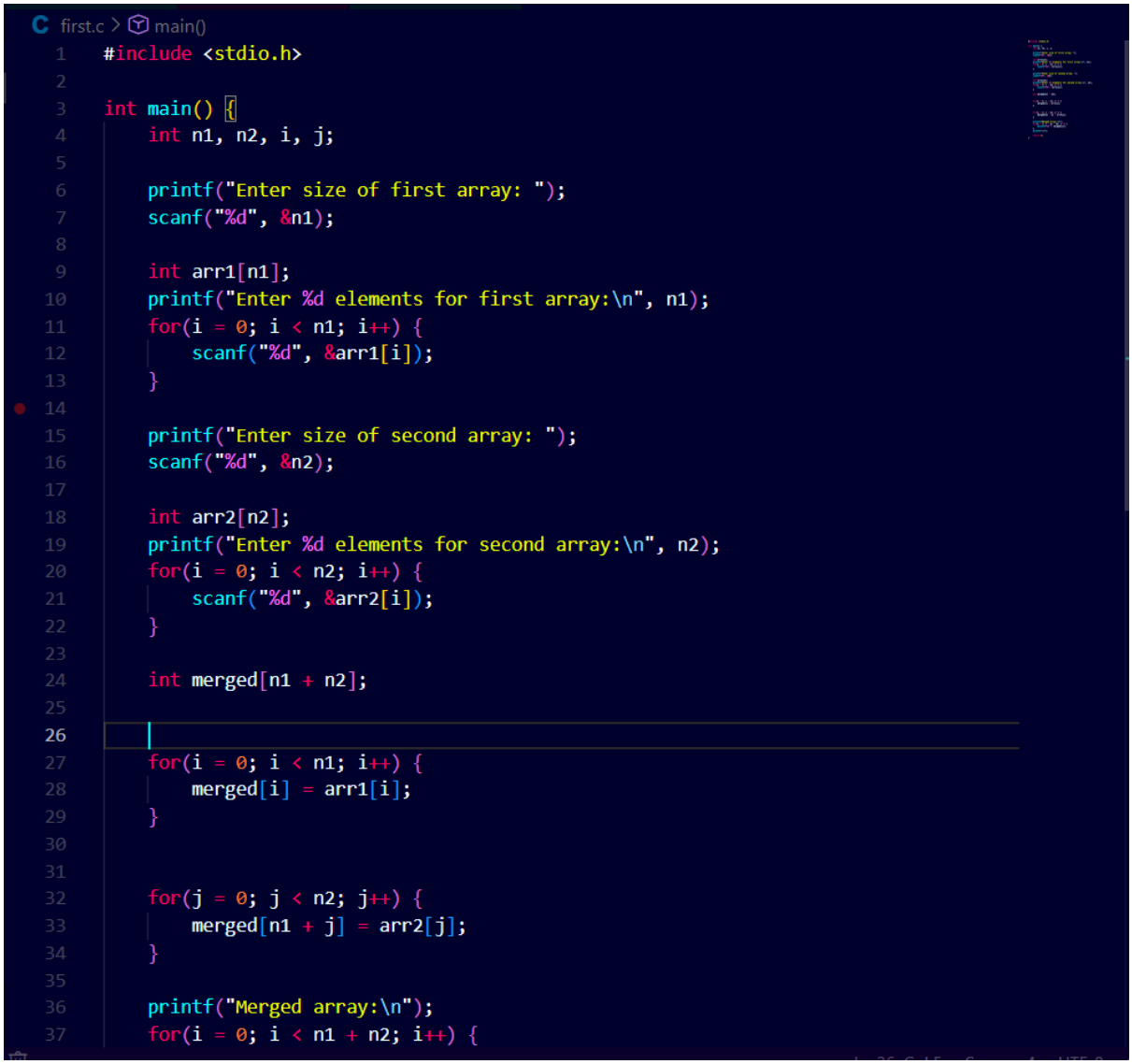
```c
#include <stdio.h>

int main() {
    int n, i;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n];

    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    for(i = 0; i < n/2; i++) {
        int temp = arr[i];
        arr[i] = arr[n - i - 1];
        arr[n - i - 1] = temp;
    }

    printf("Reversed array:\n");
    for(i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

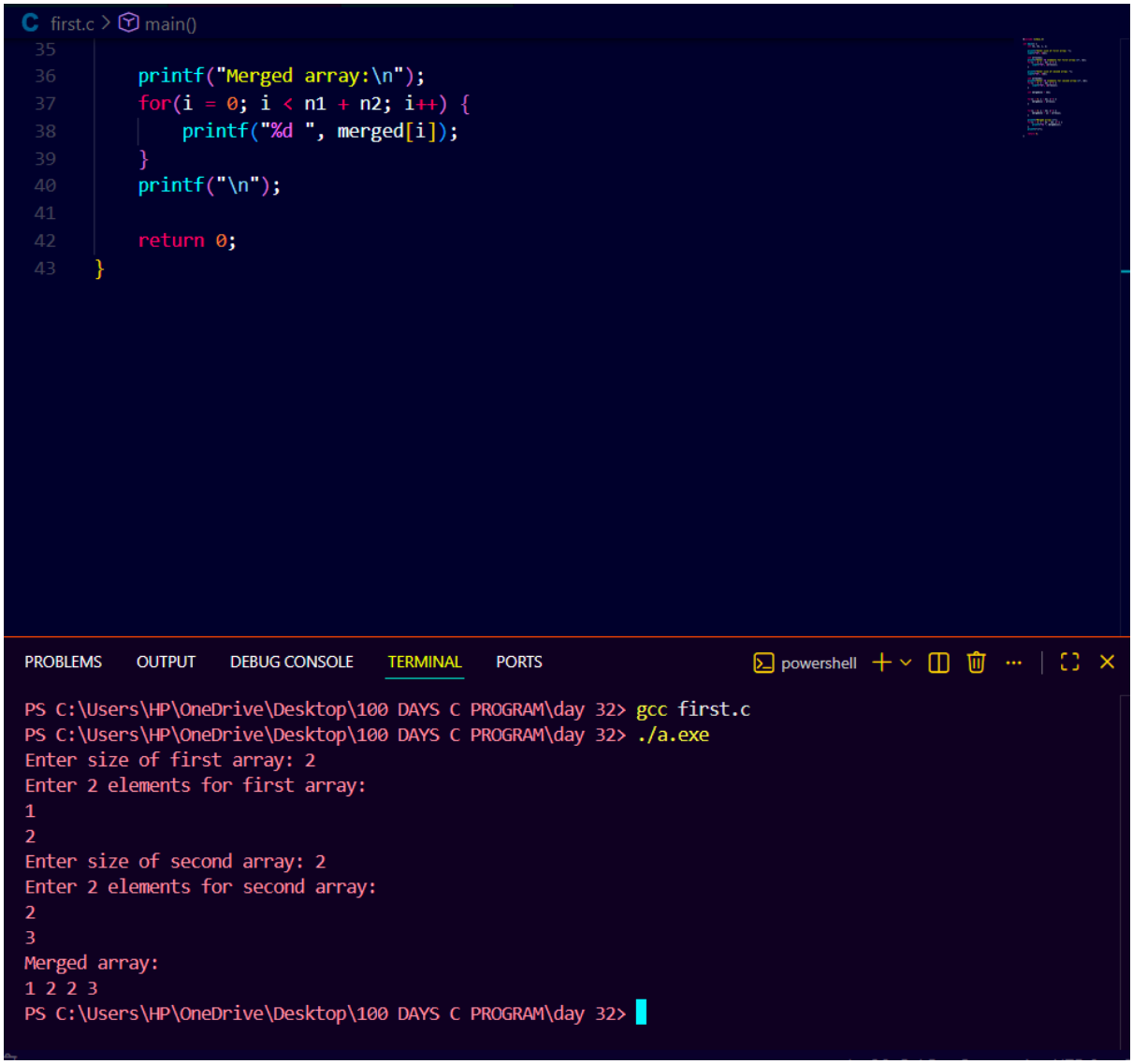
```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 31> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 31> ./a.exe
Enter number of elements: 2
Enter 2 elements:
1
2
Reversed array:
2 1
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 31>
```

DAY-32

🔢 Q63 (Arrays (1D))

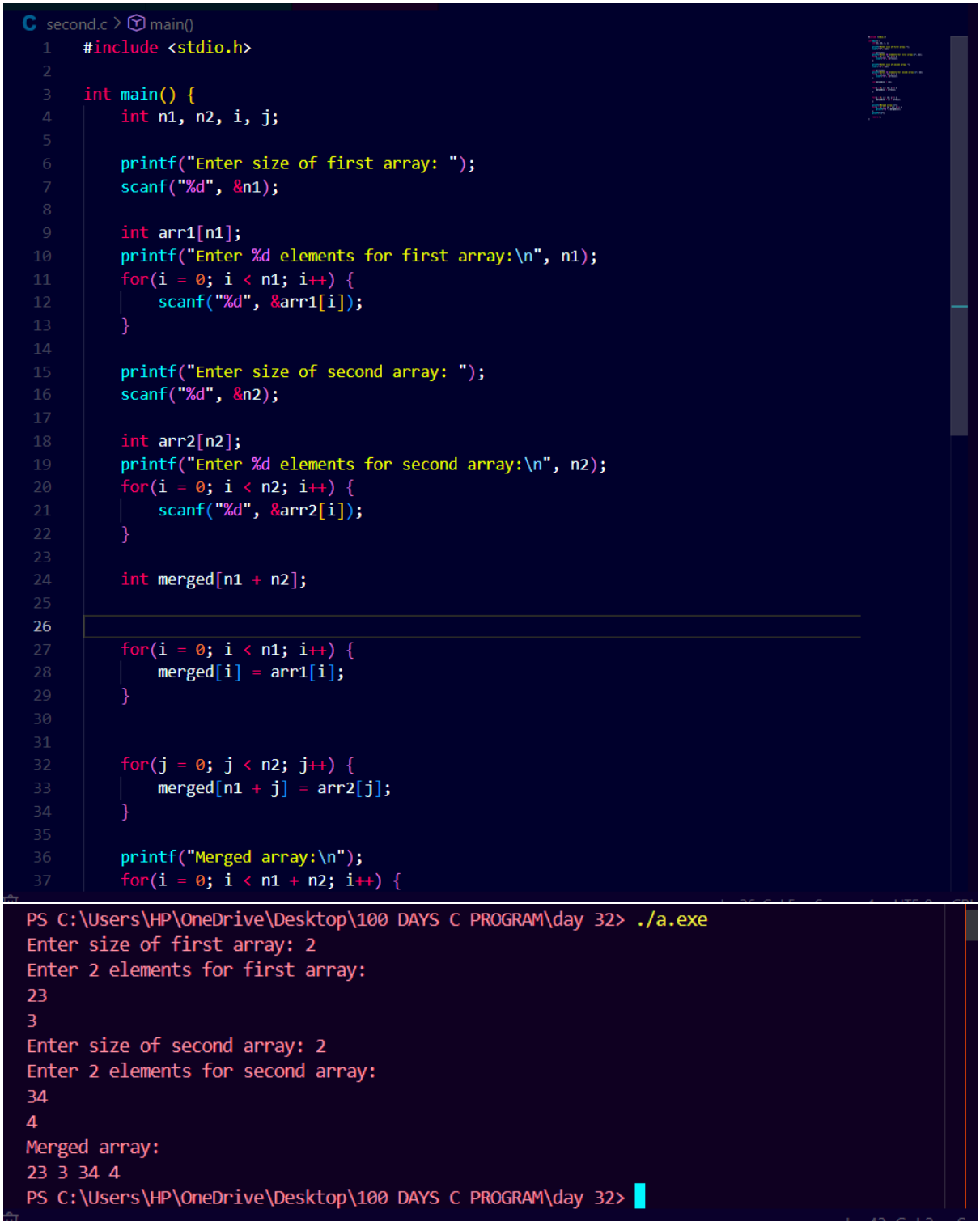
Merge two arrays.





🔢 Q64 (Arrays (1D))

Find the digit that occurs the most times in an integer number.



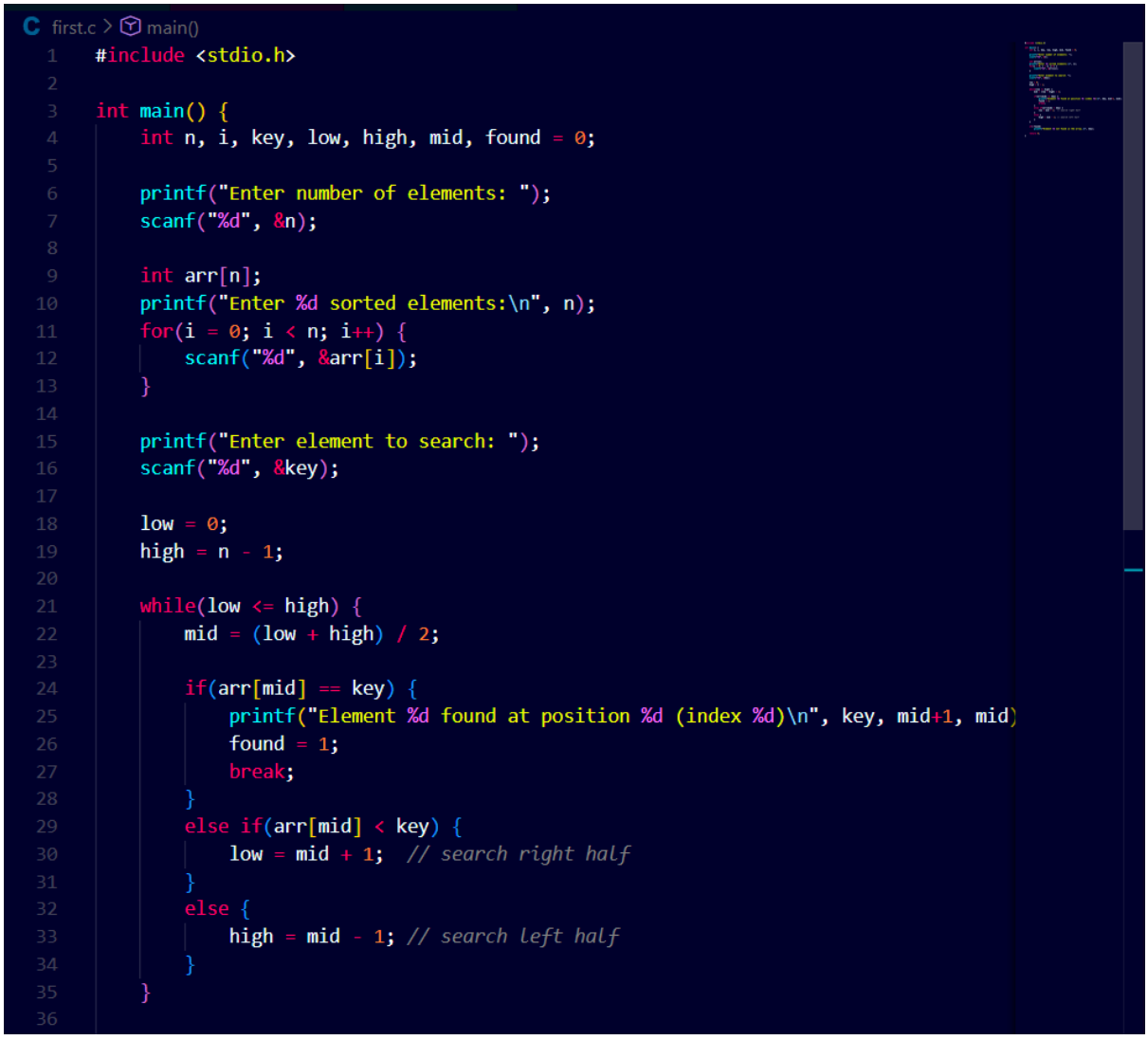
```c
#include <stdio.h>

int main() {
    int n1, n2, i, j;

    printf("Enter size of first array: ");
    scanf("%d", &n1);

    int arr1[n1];
    printf("Enter %d elements for first array:\n", n1);
    for(i = 0; i < n1; i++) {
        scanf("%d", &arr1[i]);
    }

    printf("Enter size of second array: ");
    scanf("%d", &n2);

    int arr2[n2];
    printf("Enter %d elements for second array:\n", n2);
    for(i = 0; i < n2; i++) {
        scanf("%d", &arr2[i]);
    }

    int merged[n1 + n2];


    for(i = 0; i < n1; i++) {
        merged[i] = arr1[i];
    }


    for(j = 0; j < n2; j++) {
        merged[n1 + j] = arr2[j];
    }

    printf("Merged array:\n");
    for(i = 0; i < n1 + n2; i++) {
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 32> ./a.exe
Enter size of first array: 2
Enter 2 elements for first array:
23
3
Enter size of second array: 2
Enter 2 elements for second array:
34
4
Merged array:
23 3 34 4
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 32>
```

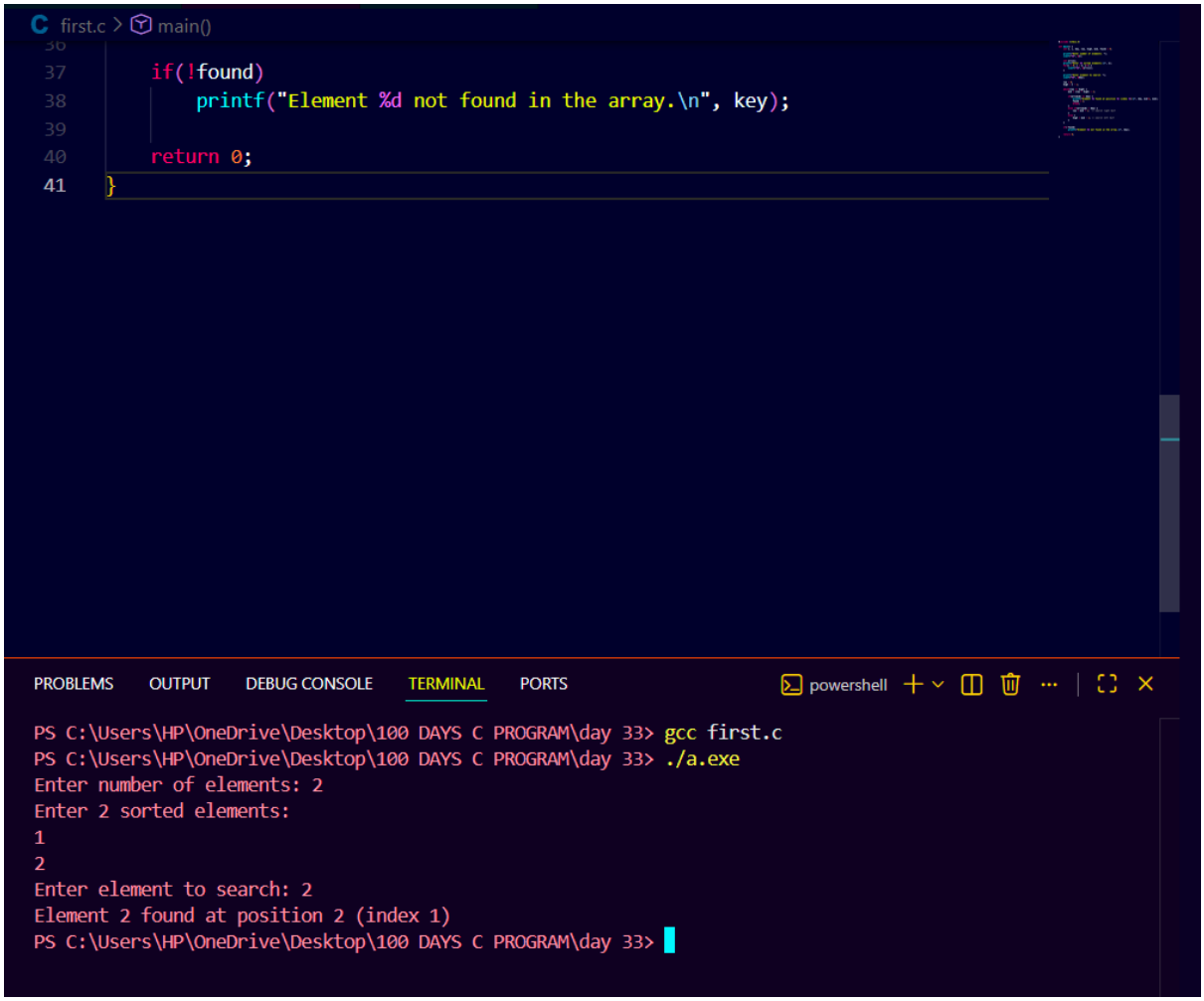
DAY-33

🔢 Q65 (Arrays (1D))

Search in a sorted array using binary search.

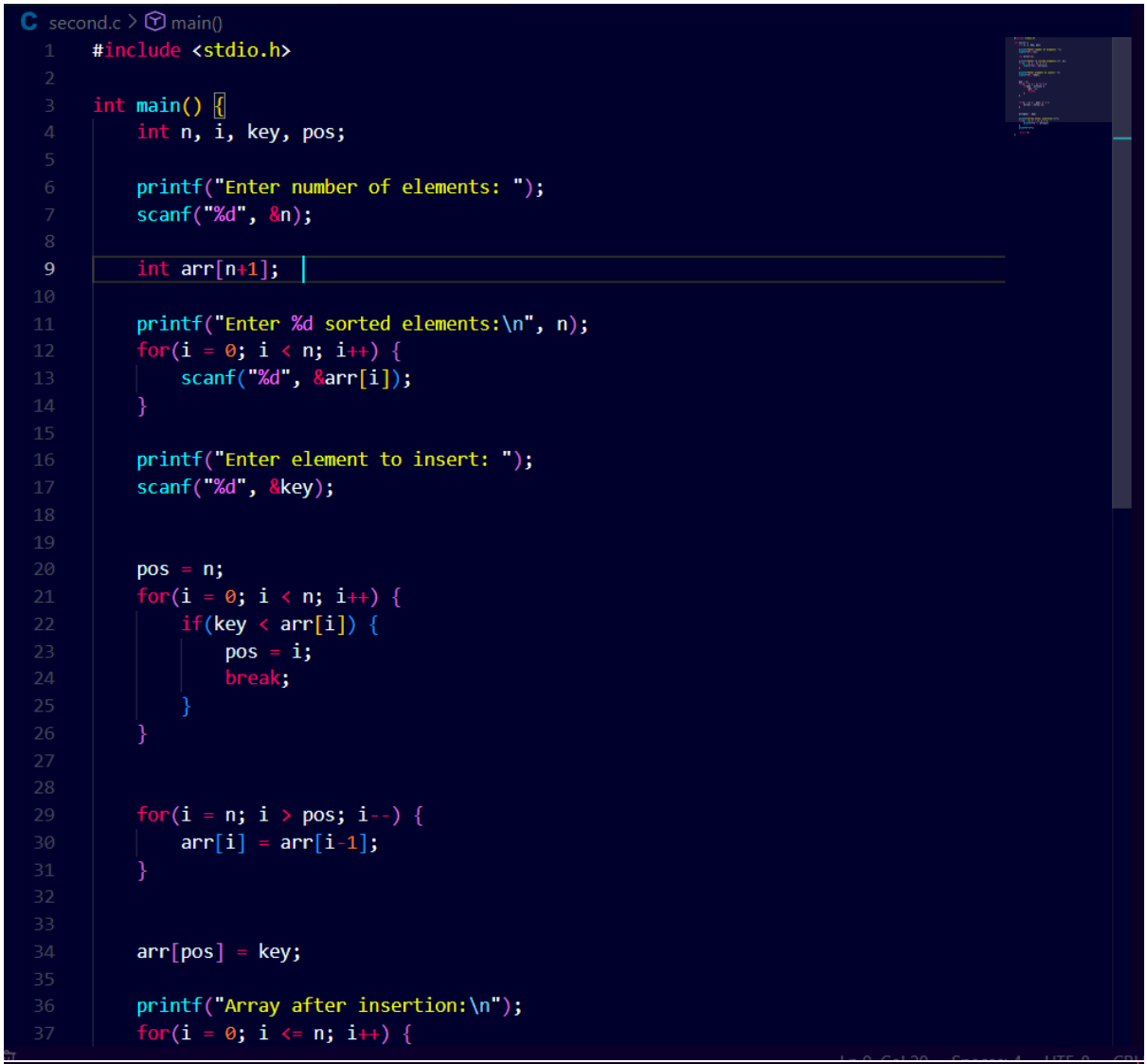


```c
#include <stdio.h>

int main() {
    int n, i, key, low, high, mid, found = 0;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter %d sorted elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Enter element to search: ");
    scanf("%d", &key);

    low = 0;
    high = n - 1;

    while(low <= high) {
        mid = (low + high) / 2;

        if(arr[mid] == key) {
            printf("Element %d found at position %d (index %d)\n", key, mid+1, mid);
            found = 1;
            break;
        }
        else if(arr[mid] < key) {
            low = mid + 1;  // search right half
        }
        else {
            high = mid - 1; // search left half
        }
    }

```



🔢 Q66 (Arrays (1D))

Insert an element in a sorted array at the appropriate position.



```c
#include <stdio.h>

int main() {
    int n, i, key, pos;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n+1];

    printf("Enter %d sorted elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Enter element to insert: ");
    scanf("%d", &key);


    pos = n;
    for(i = 0; i < n; i++) {
        if(key < arr[i]) {
            pos = i;
            break;
        }
    }


    for(i = n; i > pos; i--) {
        arr[i] = arr[i-1];
    }


    arr[pos] = key;

    printf("Array after insertion:\n");
    for(i = 0; i <= n; i++) {
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 33> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 33> ./a.exe
Enter number of elements: 2
Enter 2 sorted elements:
1
2
Enter element to insert: 2
Array after insertion:
1 2 2
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 33>
```

DAY-34

🔢 Q67 (Arrays (1D))

Insert an element in an array at a given position.

first.c > main()

```c
#include <stdio.h>

int main() {
    int n, i, pos, key;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n+1];

    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Enter element to insert: ");
    scanf("%d", &key);

    printf("Enter position (1 to %d): ", n+1);
    scanf("%d", &pos);

    if(pos < 1 || pos > n+1) {
        printf("Invalid position!\n");
        return 0;
    }


    for(i = n; i >= pos; i--) {
        arr[i] = arr[i-1];
    }


    arr[pos-1] = key;

    printf("Array after insertion:\n");
    for(i = 0; i <= n; i++) {
        printf("%d ", arr[i]);
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 34> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 34> ./a.exe
Enter number of elements: 2
Enter 2 elements:
1
2
Enter element to insert: 4
Enter position (1 to 3): 2
Array after insertion:
1 4 2
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 34>
```

🔢 Q68 (Arrays (1D))

Delete an element from an array.

second.c > main()

```c
#include <stdio.h>

int main() {
    int n, i, pos;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n];

    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Enter position to delete (1 to %d): ", n);
    scanf("%d", &pos);

    if(pos < 1 || pos > n) {
        printf("Invalid position!\n");
        return 0;
    }

    for(i = pos - 1; i < n - 1; i++) {
        arr[i] = arr[i + 1];
    }

    printf("Array after deletion:\n");
    for(i = 0; i < n - 1; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 34> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 34> ./a.exe
Enter number of elements: 2
Enter 2 elements:
1
3
Enter position to delete (1 to 2): 1
Array after deletion:
3
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 34>
```

DAY-35

🔢 Q69 (Arrays (1D))

Find the second largest element in an array.

first.c > main()

```c
//Find the second largest element in an array.
#include <stdio.h>

int main() {
    int n, i, pos;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n];

    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Enter position to delete (1 to %d): ", n);
    scanf("%d", &pos);

    if(pos < 1 || pos > n) {
        printf("Invalid position!\n");
        return 0;
    }


    for(i = pos - 1; i < n - 1; i++) {
        arr[i] = arr[i + 1];
    }

    printf("Array after deletion:\n");
    for(i = 0; i < n - 1; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 35> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 35> ./a.exe
Enter number of elements: 2
Enter 2 elements:
1
3
Enter position to delete (1 to 2): 1
Array after deletion:
3
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 35>
```

🔢 Q70 (Arrays (1D))

Rotate an array to the right by k positions.

second.c > main()

```c
#include <stdio.h>

int main() {
    int n, k, i;
 printf("Enter number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Enter k (number of positions to rotate): ");
    scanf("%d", &k);
    k = k % n;
    for(i = 0; i < n/2; i++) {
        int temp = arr[i];
        arr[i] = arr[n-1-i];
        arr[n-1-i] = temp;
    }
    for(i = 0; i < k/2; i++) {
        int temp = arr[i];
        arr[i] = arr[k-1-i];
        arr[k-1-i] = temp;
    }
    for(i = 0; i < (n-k)/2; i++) {
        int temp = arr[k+i];
        arr[k+i] = arr[n-1-i];
        arr[n-1-i] = temp;
    }
    printf("Array after rotation:\n");
    for(i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 35> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 35> ./a.exe
Enter number of elements: 2
Enter 2 elements:
1
3
Enter k (number of positions to rotate): 1
Array after rotation:
3 1
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 35>
```

DAY-36

- [ ] Q71 (2D Arrays)

Read and print a matrix.

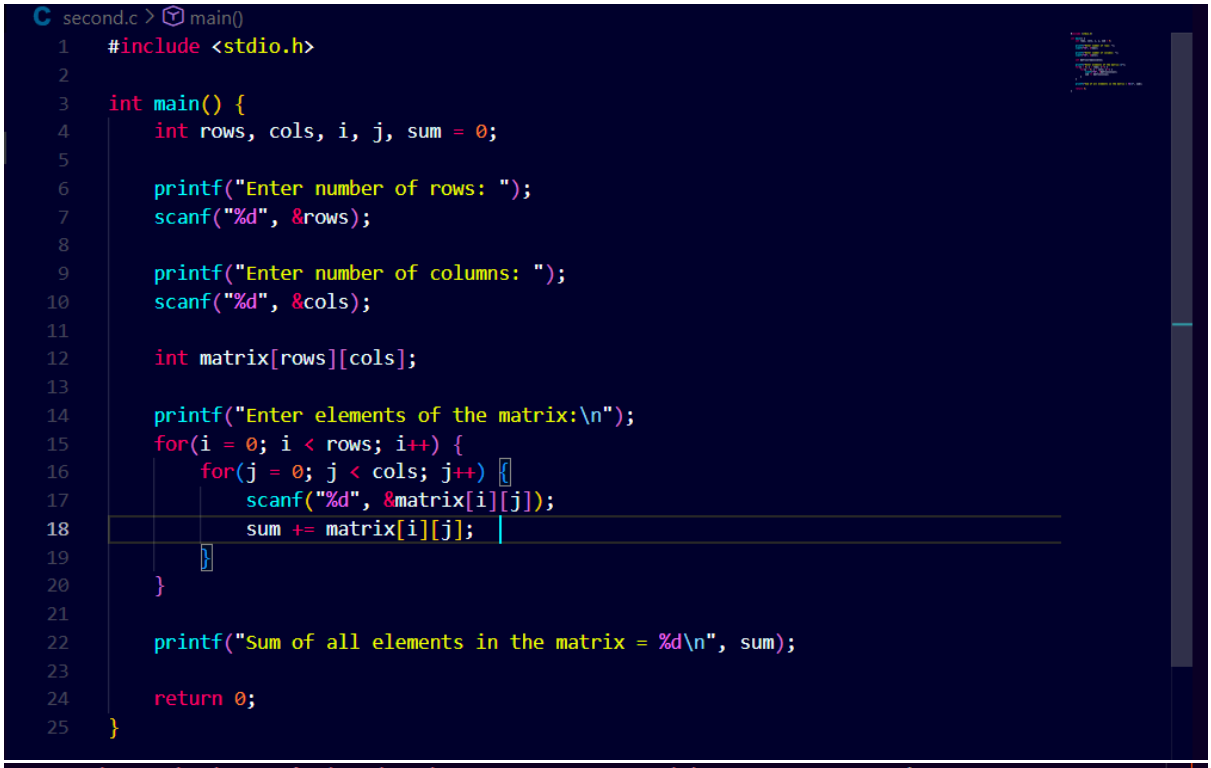
first.c > main()

```c
#include <stdio.h>

int main() {
    int rows, cols, i, j;

    printf("Enter number of rows: ");
    scanf("%d", &rows);

    printf("Enter number of columns: ");
    scanf("%d", &cols);

    int matrix[rows][cols];

    printf("Enter elements of the matrix:\n");
    for(i = 0; i < rows; i++) {
        for(j = 0; j < cols; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }

    printf("The matrix is:\n");
    for(i = 0; i < rows; i++) {
        for(j = 0; j < cols; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

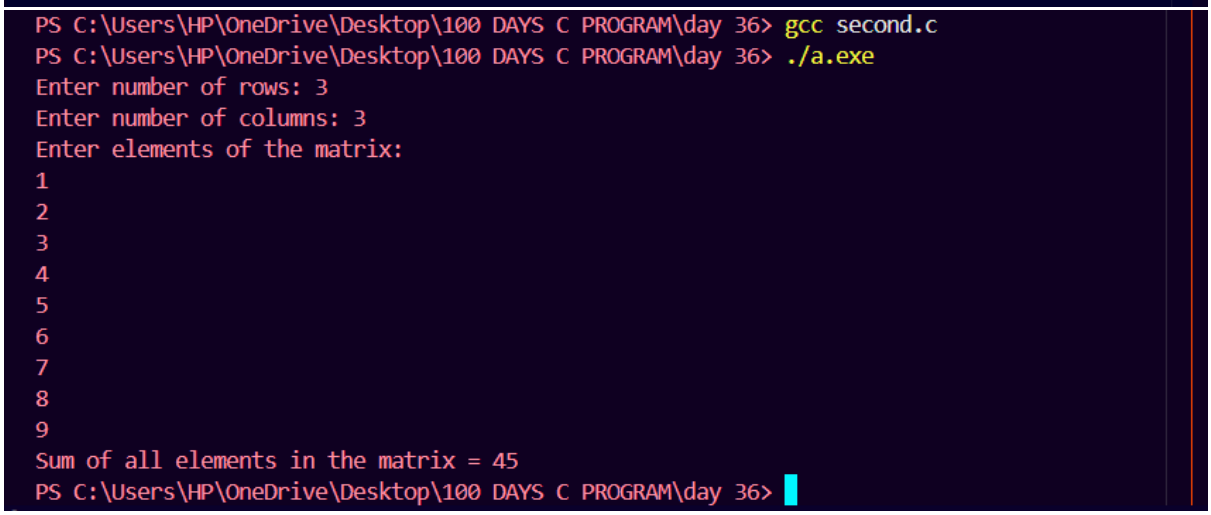
```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 36> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 36> ./a.exe
Enter number of rows: 2
Enter number of columns: 3
Enter elements of the matrix:
1
2
3
4
5
6
The matrix is:
1 2 3
4 5 6
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 36>
```

☐ Q72 (2D Arrays)

Find the sum of all elements in a matrix.



```c
#include <stdio.h>

int main() {
    int rows, cols, i, j, sum = 0;

    printf("Enter number of rows: ");
    scanf("%d", &rows);

    printf("Enter number of columns: ");
    scanf("%d", &cols);

    int matrix[rows][cols];

    printf("Enter elements of the matrix:\n");
    for(i = 0; i < rows; i++) {
        for(j = 0; j < cols; j++) {
            scanf("%d", &matrix[i][j]);
            sum += matrix[i][j];
        }
    }

    printf("Sum of all elements in the matrix = %d\n", sum);

    return 0;
}
```



```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 36> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 36> ./a.exe
Enter number of rows: 3
Enter number of columns: 3
Enter elements of the matrix:
1
2
3
4
5
6
7
8
9
Sum of all elements in the matrix = 45
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 36>
```

DAY-37

- [ ] Q73 (2D Arrays)

Find the sum of each row of a matrix and store it in an array.

first.c > main()

```c
#include <stdio.h>

int main() {
    int rows, cols, i, j;

    printf("Enter number of rows: ");
    scanf("%d", &rows);

    printf("Enter number of columns: ");
    scanf("%d", &cols);

    int matrix[rows][cols];
    int rowSum[rows];

    printf("Enter elements of the matrix:\n");
    for(i = 0; i < rows; i++) {
        for(j = 0; j < cols; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }

    for(i = 0; i < rows; i++) {
        rowSum[i] = 0;
        for(j = 0; j < cols; j++) {
            rowSum[i] += matrix[i][j];
        }
    }


    printf("Sum of each row:\n");
    for(i = 0; i < rows; i++) {
        printf("Row %d sum = %d\n", i+1, rowSum[i]);
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 37> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 37> ./a.exe
Enter number of rows: 2
Enter number of columns: 3
Enter elements of the matrix:
1
2
3
4
5
6
Sum of each row:
Row 1 sum = 6
Row 2 sum = 15
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 37>
```

☐ Q74 (2D Arrays)

Find the transpose of a matrix.

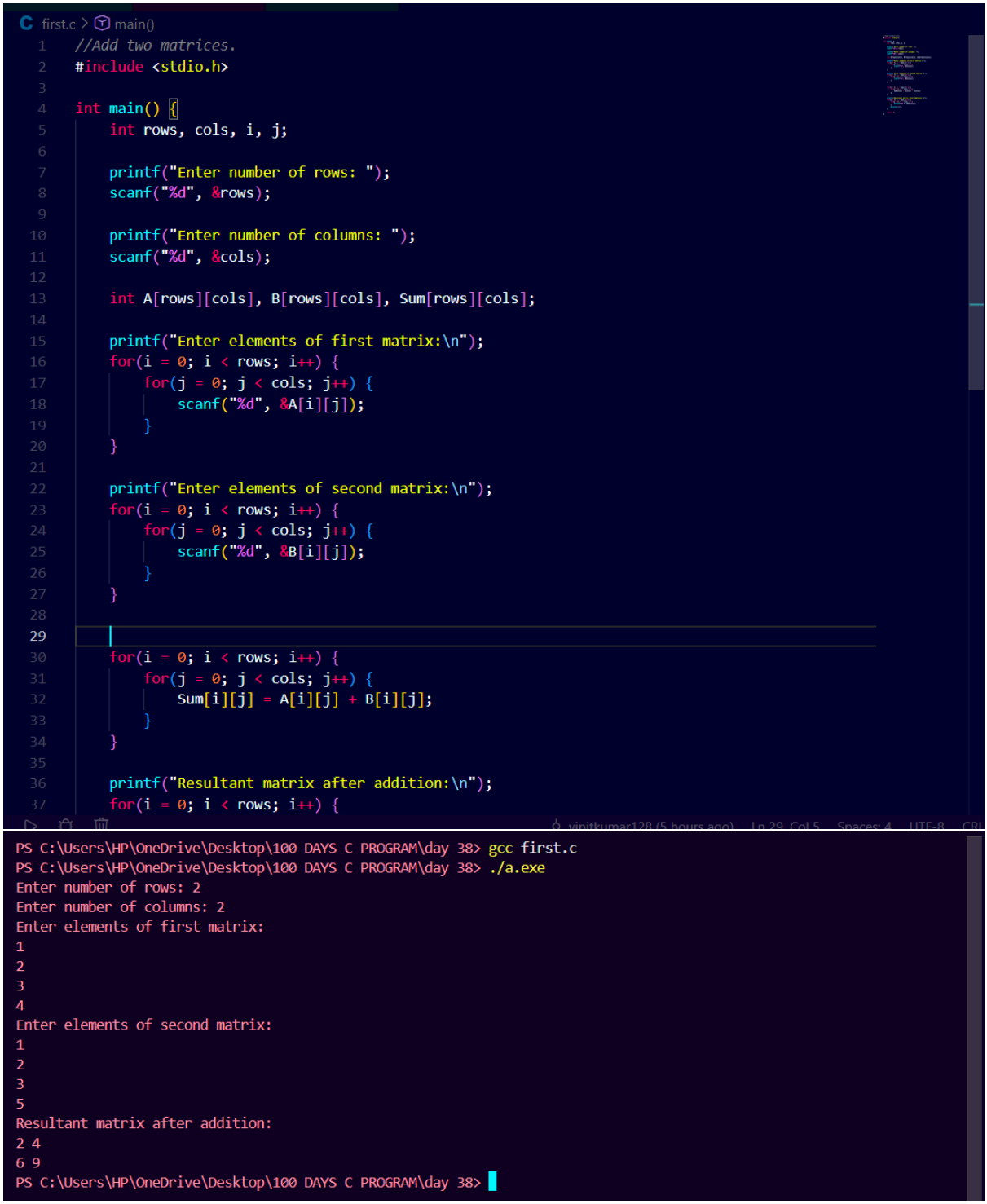
```c
#include <stdio.h>

int main() {
    int rows, cols, i, j;

    printf("Enter number of rows: ");
    scanf("%d", &rows);

    printf("Enter number of columns: ");
    scanf("%d", &cols);

    int matrix[rows][cols], transpose[cols][rows];

    printf("Enter elements of the matrix:\n");
    for(i = 0; i < rows; i++) {
        for(j = 0; j < cols; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }


    for(i = 0; i < rows; i++) {
        for(j = 0; j < cols; j++) {
            transpose[j][i] = matrix[i][j];
        }
    }

    printf("Transpose of the matrix:\n");
    for(i = 0; i < cols; i++) {
        for(j = 0; j < rows; j++) {
            printf("%d ", transpose[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 37> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 37> ./a.exe
Enter number of rows: 2
Enter number of columns: 2
Enter elements of the matrix:
1
2
3
4
Transpose of the matrix:
1 3
2 4
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 37>
```

DAY-38

- [ ] Q75 (2D Arrays)

Add two matrices.



```c
//Add two matrices.
#include <stdio.h>

int main() {
    int rows, cols, i, j;

    printf("Enter number of rows: ");
    scanf("%d", &rows);

    printf("Enter number of columns: ");
    scanf("%d", &cols);

    int A[rows][cols], B[rows][cols], Sum[rows][cols];

    printf("Enter elements of first matrix:\n");
    for(i = 0; i < rows; i++) {
        for(j = 0; j < cols; j++) {
            scanf("%d", &A[i][j]);
        }
    }

    printf("Enter elements of second matrix:\n");
    for(i = 0; i < rows; i++) {
        for(j = 0; j < cols; j++) {
            scanf("%d", &B[i][j]);
        }
    }


    for(i = 0; i < rows; i++) {
        for(j = 0; j < cols; j++) {
            Sum[i][j] = A[i][j] + B[i][j];
        }
    }

    printf("Resultant matrix after addition:\n");
    for(i = 0; i < rows; i++) {
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 38> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 38> ./a.exe
Enter number of rows: 2
Enter number of columns: 2
Enter elements of first matrix:
1
2
3
4
Enter elements of second matrix:
1
2
3
5
Resultant matrix after addition:
2 4
6 9
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 38>
```

☐ Q76 (2D Arrays)

Check if a matrix is symmetric.

```c
#include <stdio.h>

int main() {
    int n, i, j, flag = 1;

    printf("Enter the size of the square matrix: ");
    scanf("%d", &n);

    int matrix[n][n];

    printf("Enter elements of the %dx%d matrix:\n", n, n);
    for(i = 0; i < n; i++) {
        for(j = 0; j < n; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }

    for(i = 0; i < n; i++) {
        for(j = 0; j < n; j++) {
            if(matrix[i][j] != matrix[j][i]) {
                flag = 0;
                break;
            }
        }
        if(flag == 0) break;
    }

    if(flag)
        printf("The matrix is symmetric.\n");
    else
        printf("The matrix is not symmetric.\n");

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 38> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 38> ./a.exe
Enter the size of the square matrix: 3
Enter elements of the 3x3 matrix:
1
2
4
5
6
7
8
9
0
The matrix is not symmetric.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 38>
```

DAY-39

- [ ] Q77 (2D Arrays)

Check if the elements on the diagonal of a matrix are distinct.

first.c

```c
#include <stdio.h>

int main() {
    int n, i, j, k, flag = 1;

    printf("Enter the size of the square matrix: ");
    scanf("%d", &n);

    int matrix[n][n];

    printf("Enter elements of the %dx%d matrix:\n", n, n);
    for(i = 0; i < n; i++) {
        for(j = 0; j < n; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }


    for(i = 0; i < n; i++) {
        for(j = i + 1; j < n; j++) {
            if(matrix[i][i] == matrix[j][j]) {
                flag = 0;
                break;
            }
        }
        if(flag == 0) break;
    }

    if(flag)
        printf("Diagonal elements are distinct.\n");
    else
        printf("Diagonal elements are NOT distinct.\n");

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 39> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 39> ./a.exe
Enter the size of the square matrix: 2
Enter elements of the 2x2 matrix:
1
3
5
7
Diagonal elements are distinct.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 39>
```

☐ Q78 (2D Arrays)

Find the sum of main diagonal elements for a square matrix.

second.c > main()

```c
#include <stdio.h>

int main() {
    int n, i, j, sum = 0;

    printf("Enter the size of the square matrix: ");
    scanf("%d", &n);

    int matrix[n][n];

    printf("Enter elements of the %dx%d matrix:\n", n, n);
    for(i = 0; i < n; i++) {
        for(j = 0; j < n; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }


    for(i = 0; i < n; i++) {
        sum += matrix[i][i];
    }

    printf("Sum of main diagonal elements = %d\n", sum);

    return 0;
}
```

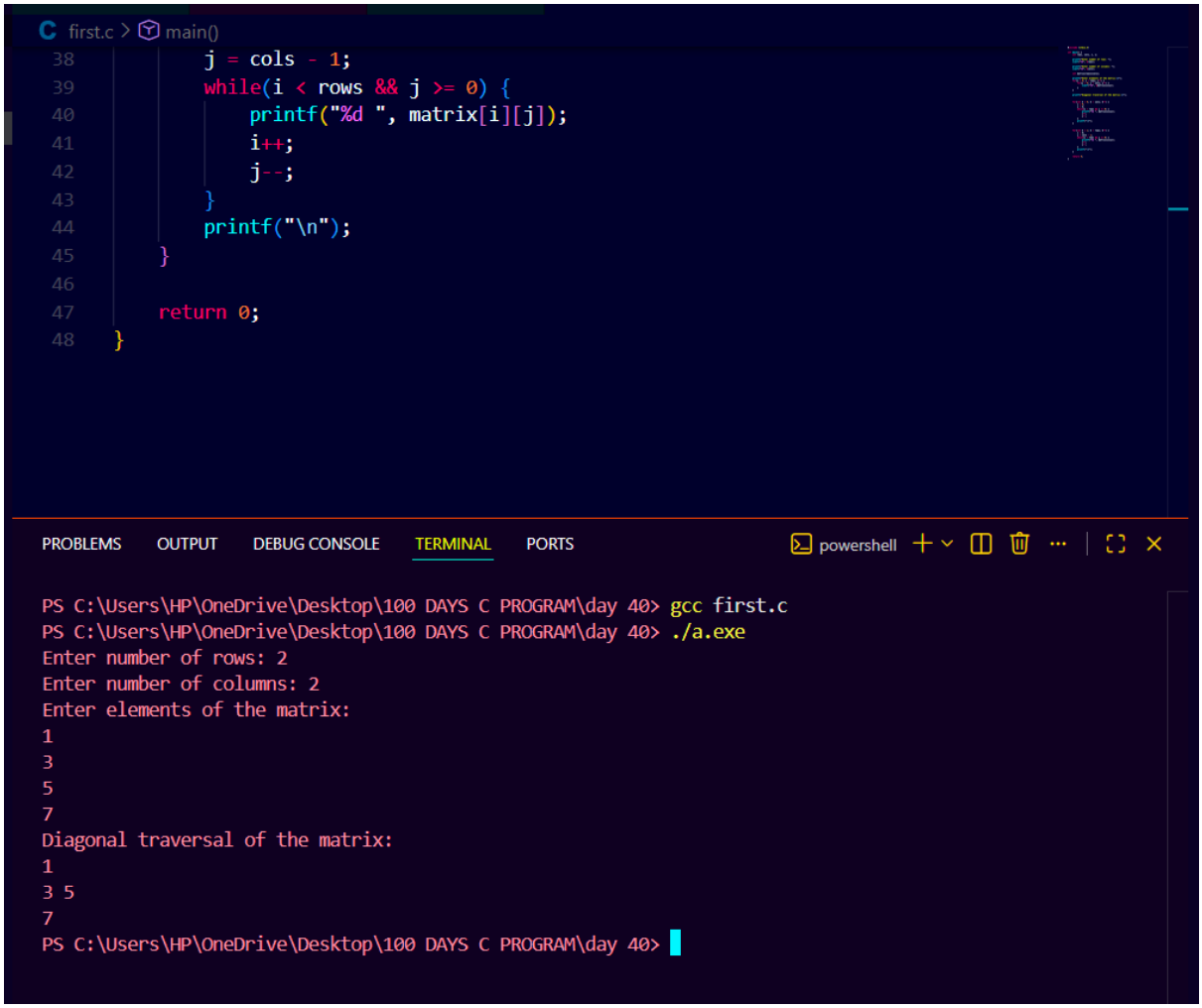
```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 39> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 39> ./a.exe
Enter the size of the square matrix: 2
Enter elements of the 2x2 matrix:
1
2
3
4
Sum of main diagonal elements = 5
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 39>
```

DAY-40

- [ ] Q79 (2D Arrays)

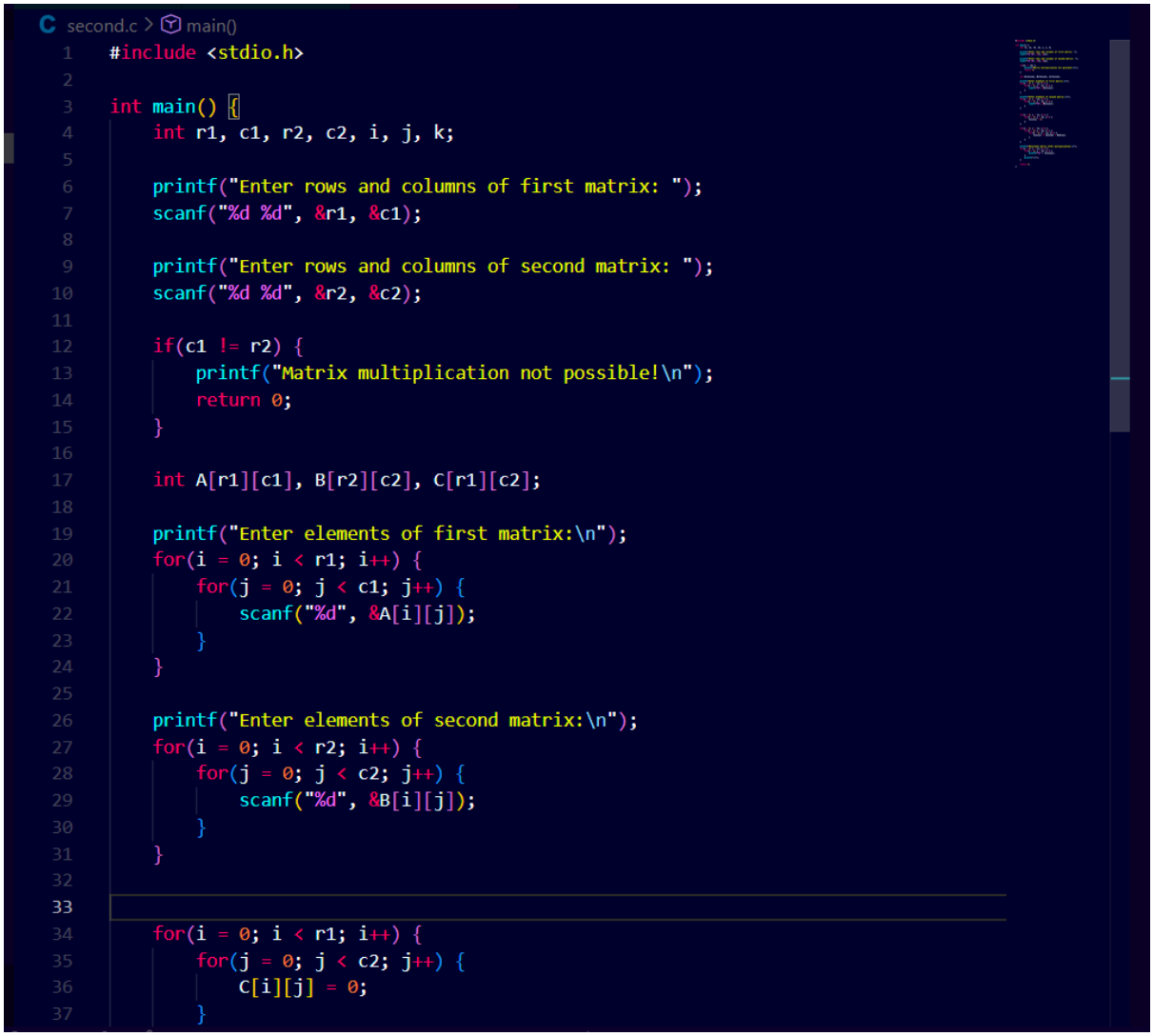
Perform diagonal traversal of a matrix.

```c
#include <stdio.h>

int main() {
    int rows, cols, i, j;

    printf("Enter number of rows: ");
    scanf("%d", &rows);

    printf("Enter number of columns: ");
    scanf("%d", &cols);

    int matrix[rows][cols];

    printf("Enter elements of the matrix:\n");
    for(i = 0; i < rows; i++) {
        for(j = 0; j < cols; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }

    printf("Diagonal traversal of the matrix:\n");


    for(int k = 0; k < cols; k++) {
        i = 0;
        j = k;
        while(i < rows && j >= 0) {
            printf("%d ", matrix[i][j]);
            i++;
            j--;
        }
        printf("\n");
    }


    for(int k = 1; k < rows; k++) {
        i = k;
```

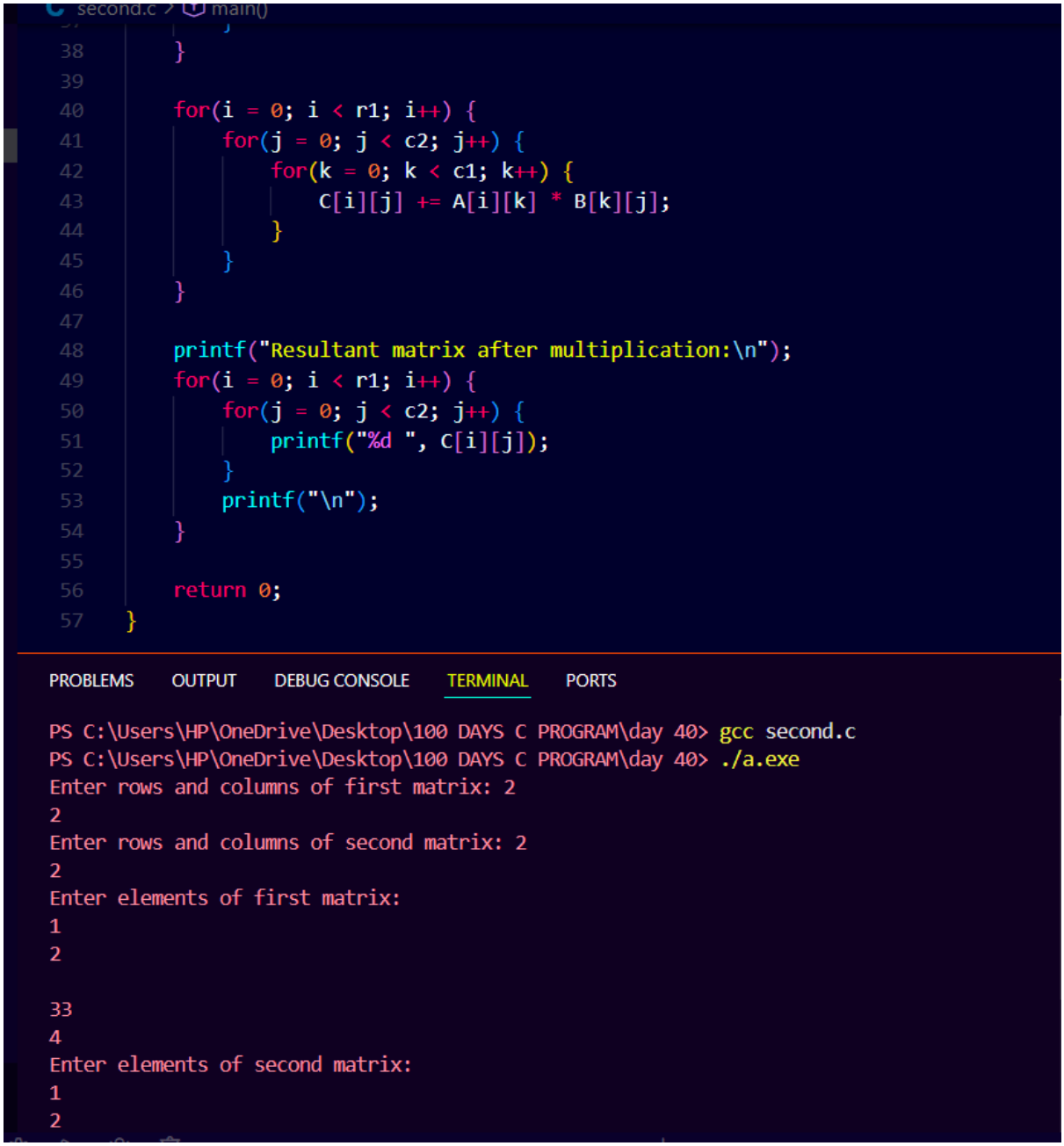


☐ Q80 (2D Arrays)

Multiply two matrices.



```c
#include <stdio.h>

int main() {
    int r1, c1, r2, c2, i, j, k;

    printf("Enter rows and columns of first matrix: ");
    scanf("%d %d", &r1, &c1);

    printf("Enter rows and columns of second matrix: ");
    scanf("%d %d", &r2, &c2);

    if(c1 != r2) {
        printf("Matrix multiplication not possible!\n");
        return 0;
    }

    int A[r1][c1], B[r2][c2], C[r1][c2];

    printf("Enter elements of first matrix:\n");
    for(i = 0; i < r1; i++) {
        for(j = 0; j < c1; j++) {
            scanf("%d", &A[i][j]);
        }
    }

    printf("Enter elements of second matrix:\n");
    for(i = 0; i < r2; i++) {
        for(j = 0; j < c2; j++) {
            scanf("%d", &B[i][j]);
        }
    }


    for(i = 0; i < r1; i++) {
        for(j = 0; j < c2; j++) {
            C[i][j] = 0;
        }
```



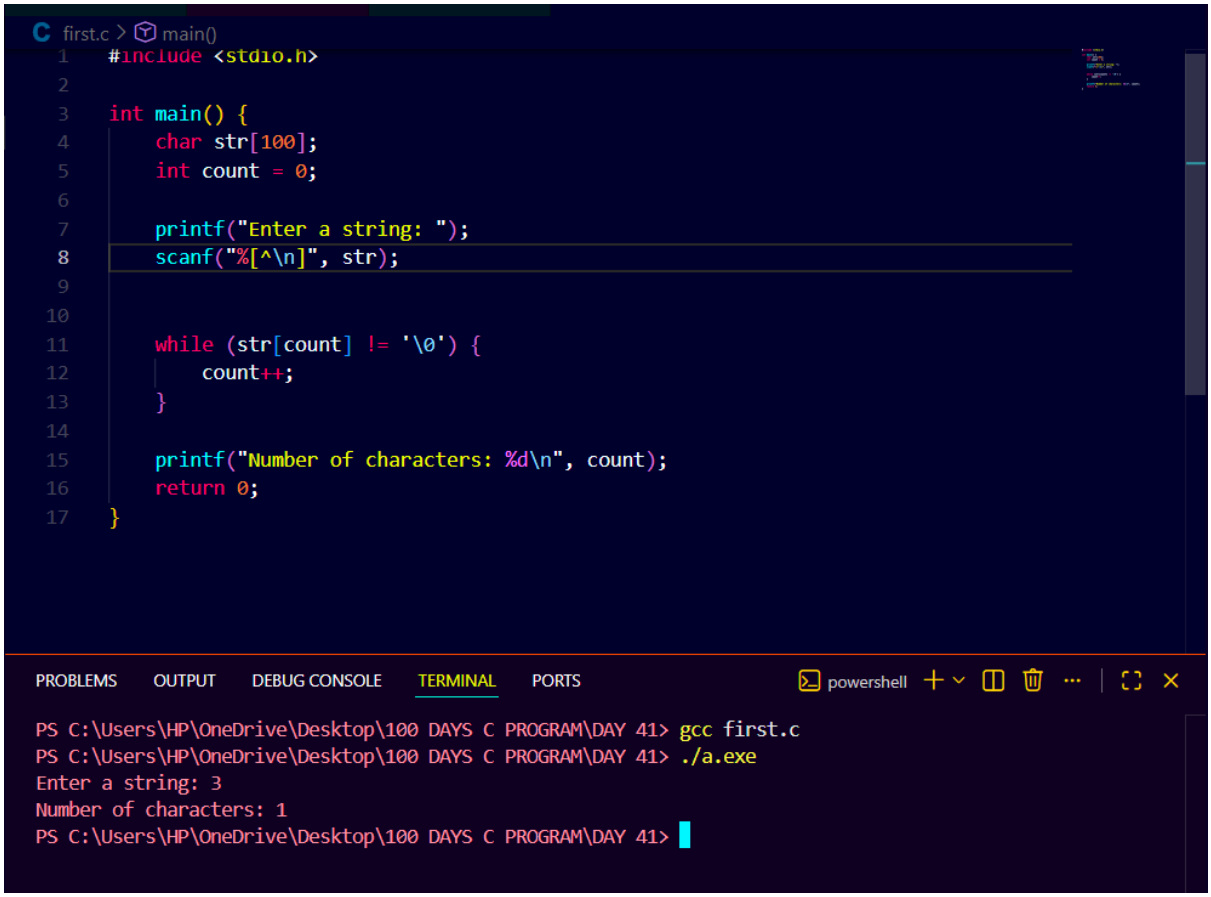
```
second.c > main()
    }

    for(i = 0; i < r1; i++) {
        for(j = 0; j < c2; j++) {
            for(k = 0; k < c1; k++) {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }

    printf("Resultant matrix after multiplication:\n");
    for(i = 0; i < r1; i++) {
        for(j = 0; j < c2; j++) {
            printf("%d ", C[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 40> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 40> ./a.exe
Enter rows and columns of first matrix: 2
2
Enter rows and columns of second matrix: 2
2
Enter elements of first matrix:
1
2

33
4
Enter elements of second matrix:
1
2
```

DAY-41

🔤 Q81 (Strings)

Count characters in a string without using built-in length functions.

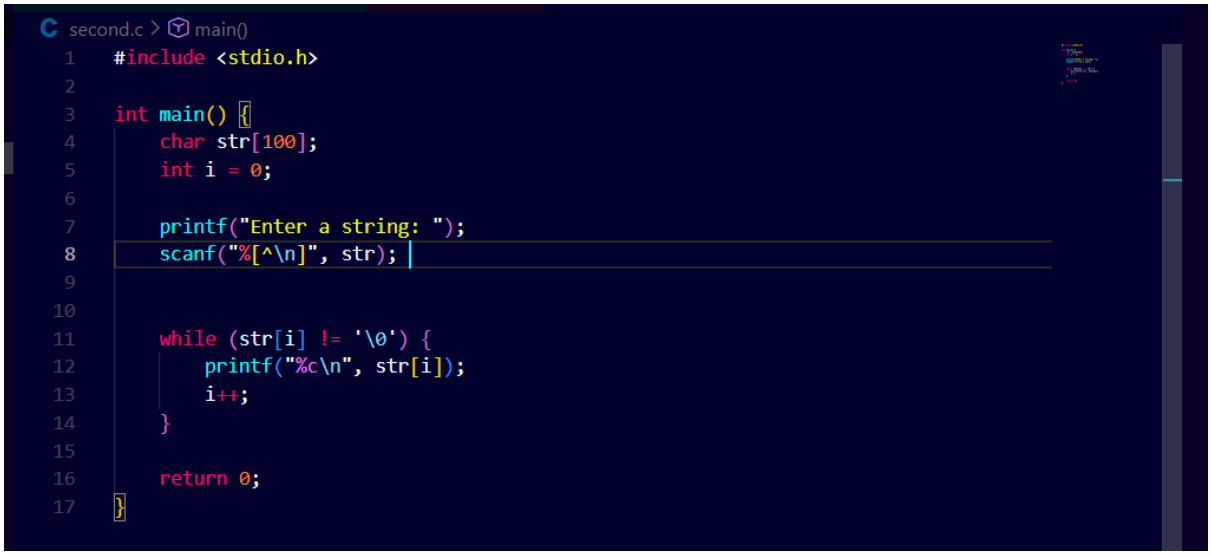


```c
#include <stdio.h>

int main() {
    char str[100];
    int count = 0;

    printf("Enter a string: ");
    scanf("%[^\n]", str);


    while (str[count] != '\0') {
        count++;
    }

    printf("Number of characters: %d\n", count);
    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\DAY 41> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\DAY 41> ./a.exe
Enter a string: 3
Number of characters: 1
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\DAY 41>
```

Q82 (Strings)

Print each character of a string on a new line.



```c
#include <stdio.h>

int main() {
    char str[100];
    int i = 0;

    printf("Enter a string: ");
    scanf("%[^\n]", str);


    while (str[i] != '\0') {
        printf("%c\n", str[i]);
        i++;
    }

    return 0;
}
```

DAY-42

Q83 (Strings)

Count vowels and consonants in a string.

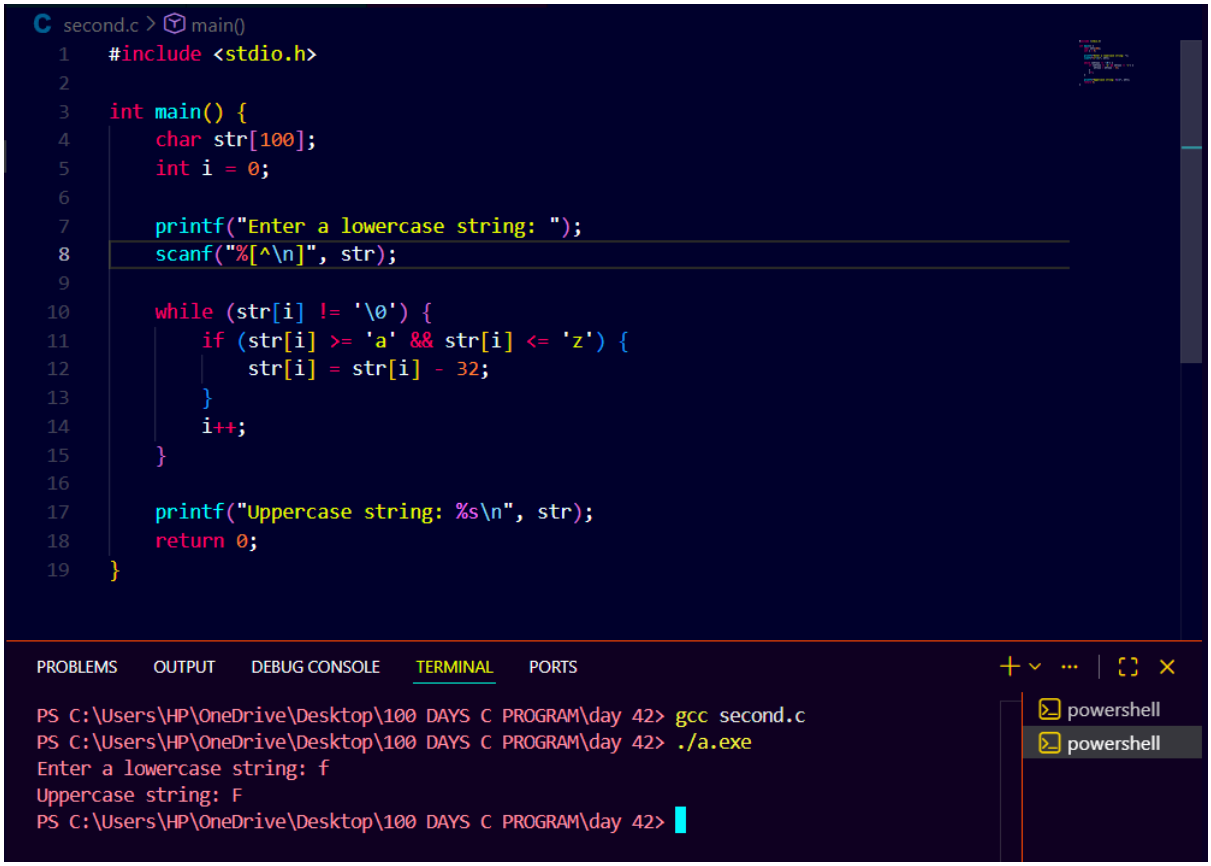
first.c > main()

```c
#include <stdio.h>

int main() {
    char str[100];
    int i = 0, vowels = 0, consonants = 0;

    printf("Enter a string: ");
    scanf("%[^\n]", str);

    while (str[i] != '\0') {
        char ch = str[i];


        if (ch >= 'A' && ch <= 'Z') {
            ch = ch + 32;
        }


        if (ch >= 'a' && ch <= 'z') {
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
                vowels++;
            else
                consonants++;
        }
        i++;
    }

    printf("Vowels: %d\n", vowels);
    printf("Consonants: %d\n", consonants);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 42> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 42> ./a.exe
Enter a string: 23s
Vowels: 0
Consonants: 1
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 42>
```

abc Q84 (Strings)

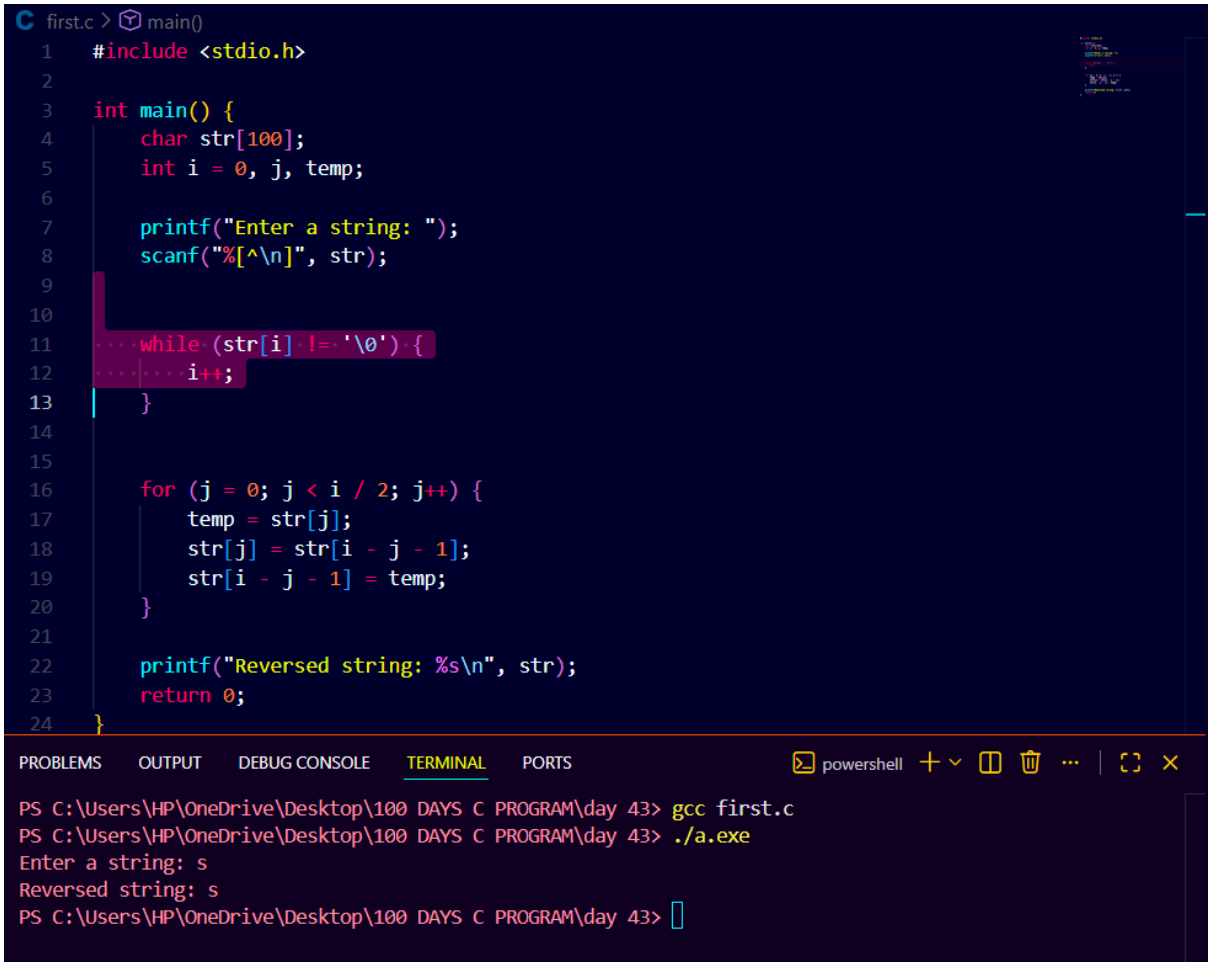
Convert a lowercase string to uppercase without using built-in functions.



DAY-43

Q85 (Strings)

Reverse a string.



Q86 (Strings)

Check if a string is a palindrome.

```c
#include <stdio.h>

int main() {
    char str[100];
    int i = 0, j, length = 0, flag = 1;

    printf("Enter a string: ");
    scanf("%[^\n]", str);


    while (str[length] != '\0') {
        length++;
    }


    for (i = 0, j = length - 1; i < j; i++, j--) {
        if (str[i] != str[j]) {
            flag = 0;
            break;
        }
    }

    if (flag == 1)
        printf("The string is a palindrome.\n");
    else
        printf("The string is NOT a palindrome.\n");

    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 43> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 43> ./a.exe
Enter a string: t
The string is a palindrome.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 43>
```

DAY-44

Q87 (Strings)

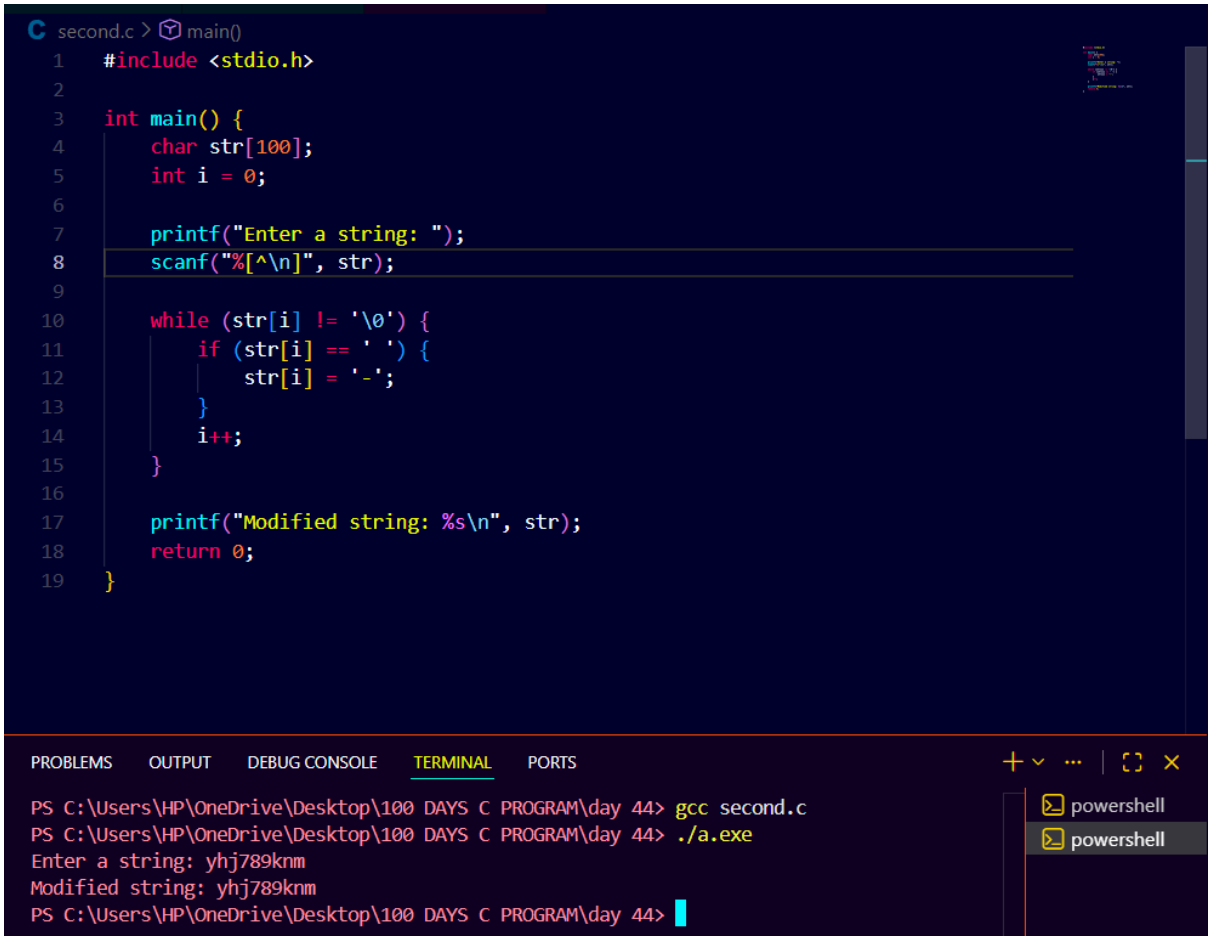
Count spaces, digits, and special characters in a string.

```c
#include <stdio.h>

int main() {
    char str[100];
    int i = 0, spaces = 0, digits = 0, specials = 0;

    printf("Enter a string: ");
    scanf("%[^\n]", str);

    while (str[i] != '\0') {
        if (str[i] == ' ') {
            spaces++;
        }
        else if (str[i] >= '0' && str[i] <= '9') {
            digits++;
        }
        else if ((str[i] >= 'A' && str[i] <= 'Z') || (str[i] >= 'a' && str[i] <= 'z
        }
        else {
            specials++;
        }
        i++;
    }

    printf("Spaces: %d\n", spaces);
    printf("Digits: %d\n", digits);
    printf("Special characters: %d\n", specials);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 44> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 44> ./a.exe
Enter a string: g
Spaces: 0
Digits: 0
Special characters: 0
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 44>
```

🔤 Q88 (Strings)

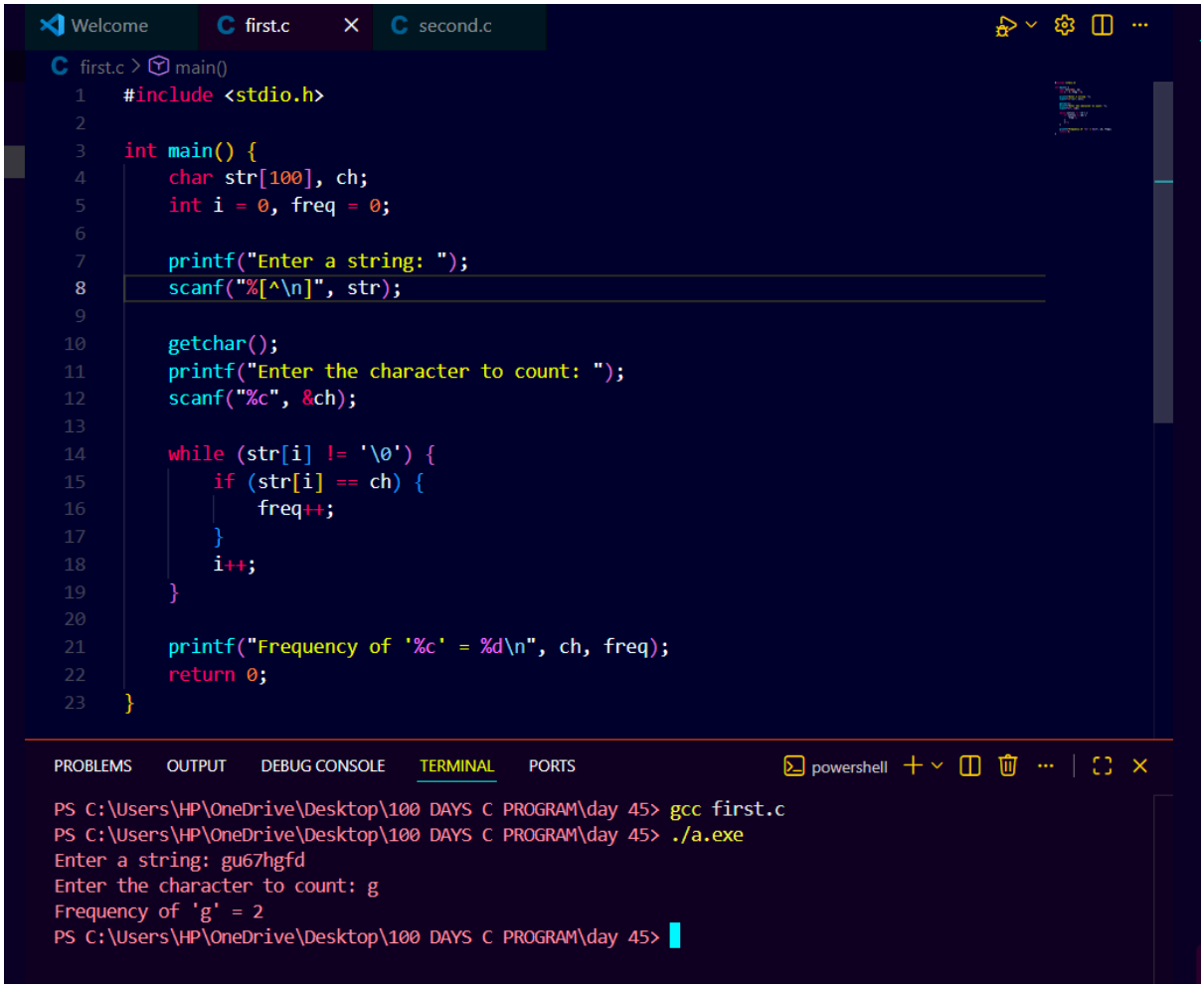
Replace spaces with hyphens in a string.



DAY-45

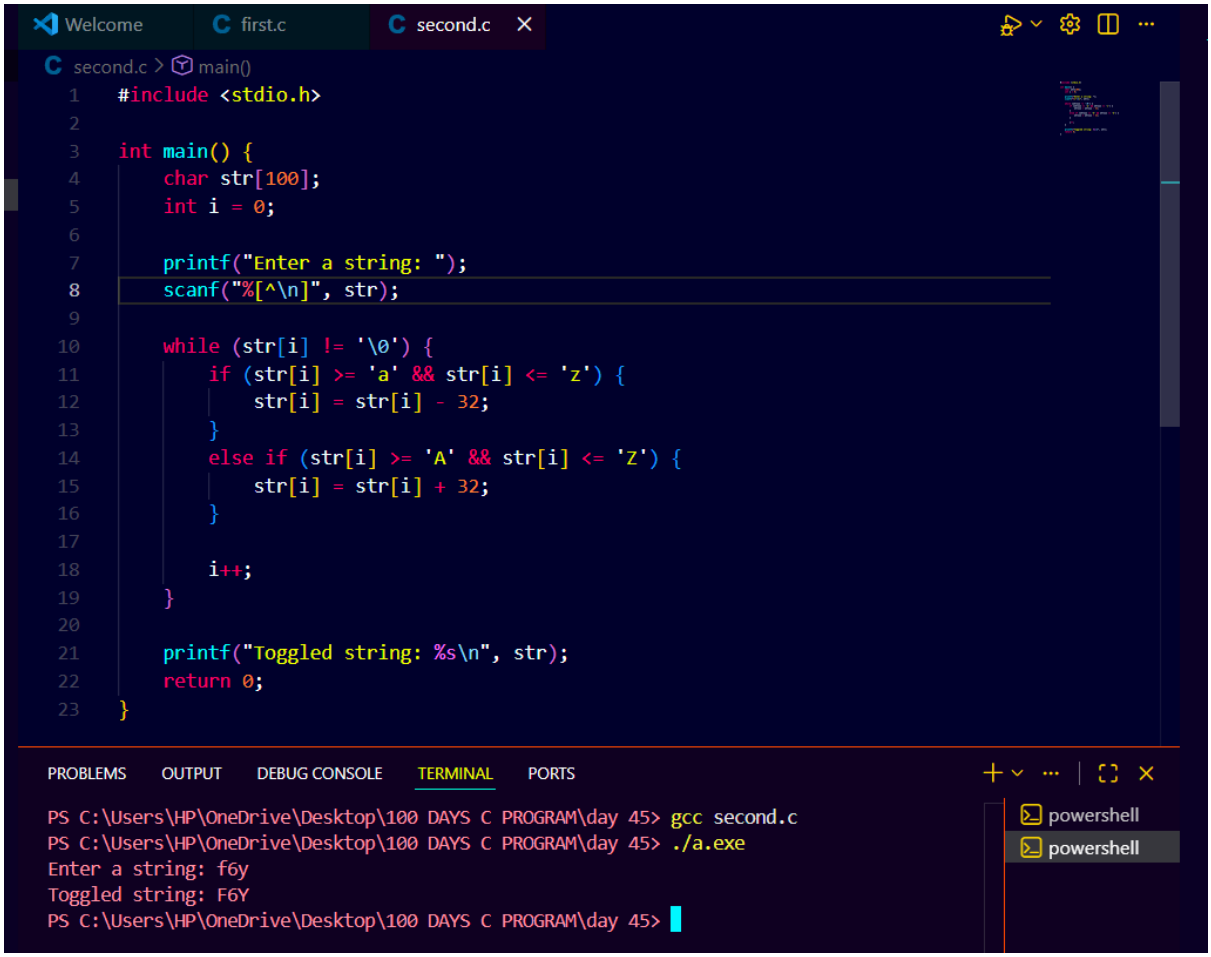
Q89 (Strings)

Count frequency of a given character in a string.



Q90 (Strings)

Toggle case of each character in a string.



DAY-46

Q91 (Strings)

Remove all vowels from a string.

first.c > main()

```c
#include <stdio.h>

int main() {
    char str[100], result[100];
    int i = 0, j = 0;

    printf("Enter a string: ");
    scanf("%[^\n]", str);

    while (str[i] != '\0') {
        char ch = str[i];

        if (ch >= 'A' && ch <= 'Z') {
            ch = ch + 32;
        }


        if (!(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')) {
            result[j] = str[i];
            j++;
        }
        i++;
    }
    result[j] = '\0';

    printf("String without vowels: %s\n", result);
    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 46> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 46> ./a.exe
Enter a string: er3
String without vowels: r3
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 46>
```

abc Q92 (Strings)

Find the first repeating lowercase alphabet in a string.

second.c > main()

```c
#include <stdio.h>

int main() {
    char str[100];
    int seen[26] = {0};
    int i = 0, found = 0;

    printf("Enter a string: ");
    scanf("%[^\n]", str);

    while (str[i] != '\0') {
        if (str[i] >= 'a' && str[i] <= 'z') {
            int index = str[i] - 'a';
            if (seen[index] == 1) {
                printf("First repeating lowercase alphabet: %c\n", str[i]);
                found = 1;
                break;
            } else {
                seen[index] = 1;
            }
        }
        i++;
    }

    if (!found) {
        printf("No repeating lowercase alphabet found.\n");
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 46> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 46> ./a.exe
Enter a string: df4
No repeating lowercase alphabet found.
```

DAY-47

abc Q93 (Strings)

Check if two strings are anagrams of each other.

first.c > main()

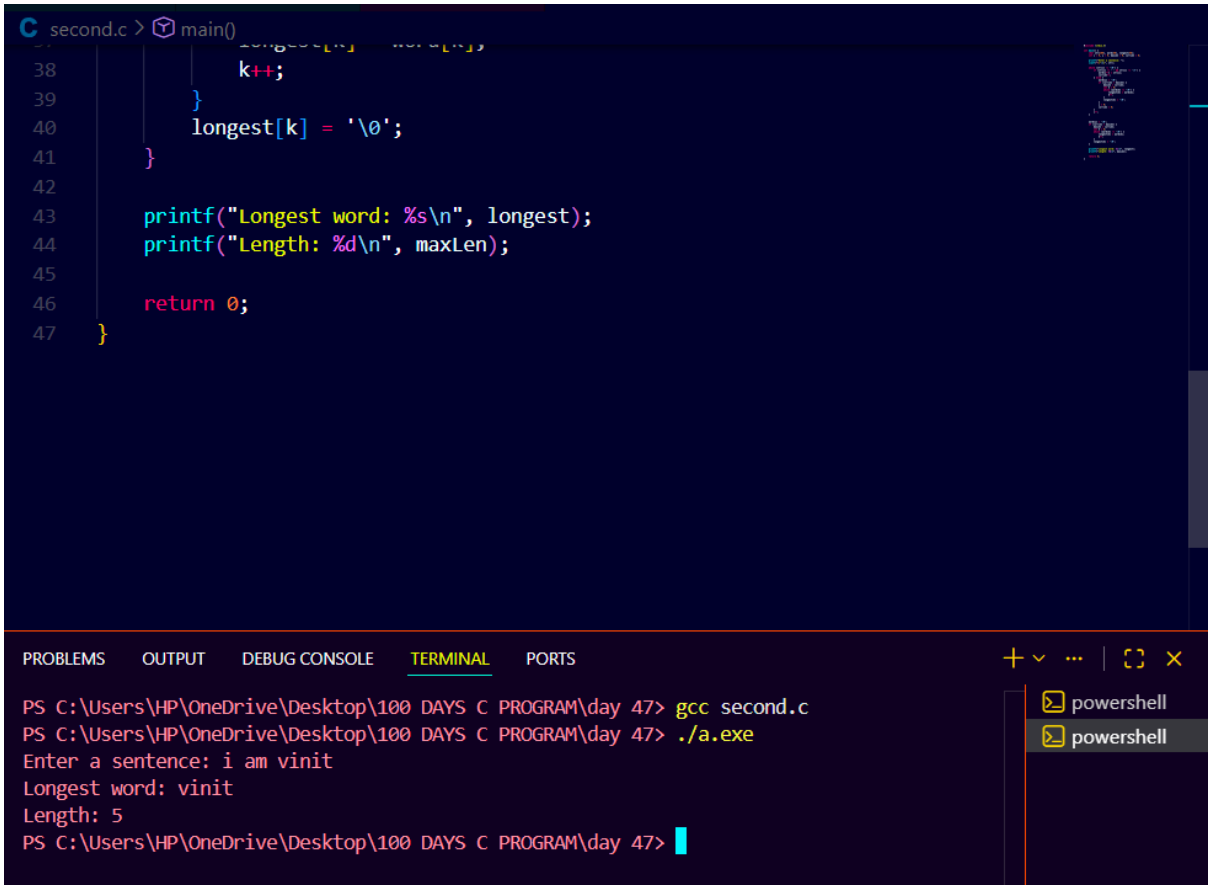
```c
#include <stdio.h>

int main() {
    char str1[100], str2[100];
    int freq1[26] = {0}, freq2[26] = {0};
    int i = 0, flag = 1;

    printf("Enter first string: ");
    scanf("%[^\n]", str1);
    getchar();
    printf("Enter second string: ");
    scanf("%[^\n]", str2);


    while (str1[i] != '\0') {
        if (str1[i] >= 'a' && str1[i] <= 'z')
            freq1[str1[i] - 'a']++;
        else if (str1[i] >= 'A' && str1[i] <= 'Z')
            freq1[str1[i] - 'A']++;
        i++;
    }

    i = 0;

    while (str2[i] != '\0') {
        if (str2[i] >= 'a' && str2[i] <= 'z')
            freq2[str2[i] - 'a']++;
        else if (str2[i] >= 'A' && str2[i] <= 'Z')
            freq2[str2[i] - 'A']++;
        i++;
    }


    for (i = 0; i < 26; i++) {
        if (freq1[i] != freq2[i]) {
            flag = 0;
            break;
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 47> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 47> ./a.exe
Enter first string: fr5
Enter second string: sd4
The strings are NOT anagrams.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 47>
```

🔤 Q94 (Strings)

Find the longest word in a sentence.

```c
#include <stdio.h>

int main() {
    char str[200], word[50], longest[50];
    int i = 0, j = 0, maxLen = 0, currLen = 0;

    printf("Enter a sentence: ");
    scanf("%[^\n]", str);

    while (str[i] != '\0') {
        if (str[i] != ' ' && str[i] != '\t') {
            word[j++] = str[i];
            currLen++;
        } else {
            word[j] = '\0';
            if (currLen > maxLen) {
                maxLen = currLen;
                int k = 0;
                while (word[k] != '\0') {
                    longest[k] = word[k];
                    k++;
                }
                longest[k] = '\0';
            }
            j = 0;
            currLen = 0;
        }
        i++;
    }


    word[j] = '\0';
    if (currLen > maxLen) {
        maxLen = currLen;
        int k = 0;
        while (word[k] != '\0') {
            longest[k] = word[k];
```



DAY-48

abc Q95 (Strings)

Check if one string is a rotation of another.

first.c > isRotation(char *, char *)

```c
#include <stdio.h>
#include <string.h>

int isRotation(char *str1, char *str2) {
    int len1 = 0, len2 = 0;

    // Find lengths manually
    while (str1[len1] != '\0') len1++;
    while (str2[len2] != '\0') len2++;

    if (len1 != len2) return 0;


    char temp[200];
    int i, j;

    for (i = 0; i < len1; i++) temp[i] = str1[i];
    for (j = 0; j < len1; j++) temp[i + j] = str1[j];
    temp[2 * len1] = '\0';


    for (i = 0; i <= 2 * len1 - len2; i++) {
        int k = 0;
        while (k < len2 && temp[i + k] == str2[k]) {
            k++;
        }
        if (k == len2) return 1;
    }

    return 0;
}

int main() {
    char str1[100], str2[100];
    printf("Enter first string: ");
    scanf("%s", str1);
    printf("Enter second string: ");
    scanf("%s", str2);

    if (isRotation(str1, str2))
        printf("The strings are rotations of each other.\n");
    else
        printf("The strings are NOT rotations of each other.\n");

    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS        powershell

PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 48> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 48> ./a.exe
Enter first string: hbgdjuh7
Enter second string: uhu6
The strings are NOT rotations of each other.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 48>
```

abc Q96 (Strings)

Reverse each word in a sentence without changing the word order.

second.c > main()

```c
#include <stdio.h>

int main() {
    char str[200];
    int i = 0, start = 0, end;

    printf("Enter a sentence: ");
    scanf("%[^\n]", str);

    while (str[i] != '\0') {

        if (str[i] != ' ' && str[i] != '\t') {
            start = i;

            while (str[i] != ' ' && str[i] != '\t' && str[i] != '\0') {
                i++;
            }
            end = i - 1;


            while (start < end) {
                char temp = str[start];
                str[start] = str[end];
                str[end] = temp;
                start++;
                end--;
            }
        }
        else {
            i++;
        }
    }

    printf("Sentence with each word reversed: %s\n", str);
    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 48> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 48> ./a.exe
Enter a sentence: i am vinit
Sentence with each word reversed: i ma tiniv
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 48>
```

DAY-49

abc Q97 (Strings)

Print the initials of a name.

abc Q98 (Strings)

Print initials of a name with the surname displayed in full.

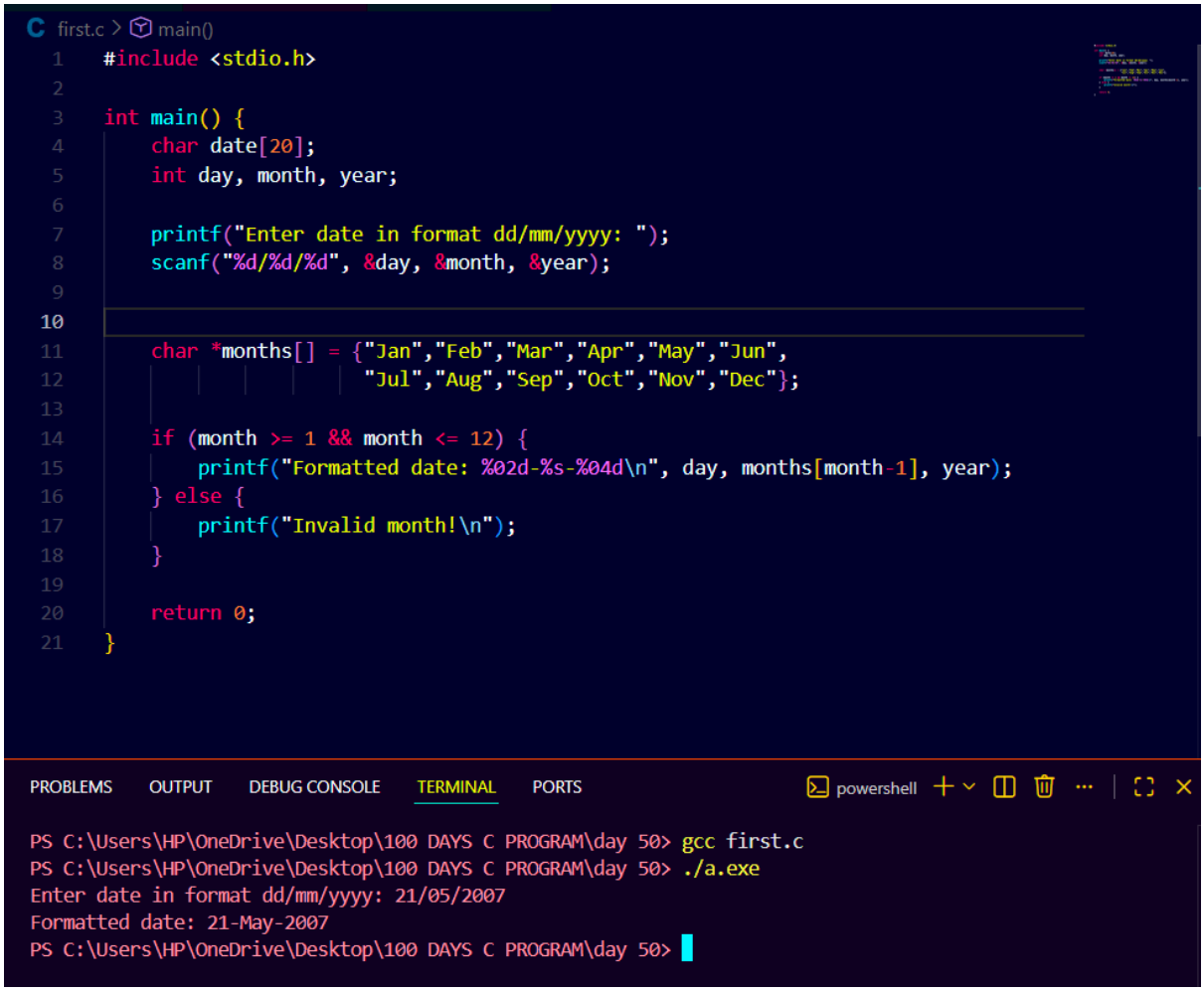
```c
#include <stdio.h>

int main() {
    char str[100];
    int i = 0;

    printf("Enter a full name: ");
    scanf("%[^\n]", str);
    if (str[0] != ' ')
        printf("%c.", str[0]);
    while (str[i] != '\0') {
        if (str[i] == ' ' && str[i+1] != ' ' && str[i+1] != '\0') {
            int j = i + 1;
            while (str[j] != ' ' && str[j] != '\0') {
                j++;
            }
            if (str[j] == '\0') {

                printf(" ");
                j = i + 1;
                while (str[j] != '\0') {
                    printf("%c", str[j]);
                    j++;
                }
                break;
            } else {

                printf("%c.", str[i+1]);
            }
        }
        i++;
    }

    printf("\n");
    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 49> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 49> ./a.exe
Enter a full name: vinit kumar
v. kumar
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 49>
```

DAY-50

Q99 (Strings)

Change the date format from dd/04/yyyy to dd-Apr-yyyy.



```c
#include <stdio.h>

int main() {
    char date[20];
    int day, month, year;

    printf("Enter date in format dd/mm/yyyy: ");
    scanf("%d/%d/%d", &day, &month, &year);


    char *months[] = {"Jan","Feb","Mar","Apr","May","Jun",
                      "Jul","Aug","Sep","Oct","Nov","Dec"};

    if (month >= 1 && month <= 12) {
        printf("Formatted date: %02d-%s-%04d\n", day, months[month-1], year);
    } else {
        printf("Invalid month!\n");
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 50> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 50> ./a.exe
Enter date in format dd/mm/yyyy: 21/05/2007
Formatted date: 21-May-2007
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 50>
```

Q100 (Strings)

Print all sub-strings of a string.

second.c > main()

```c
#include <stdio.h>

int main() {
    char str[100];
    int i, j, k;

    printf("Enter a string: ");
    scanf("%s", str);


    int length = 0;
    while (str[length] != '\0') {
        length++;
    }


    for (i = 0; i < length; i++) {
        for (j = i; j < length; j++) {

            for (k = i; k <= j; k++) {
                printf("%c", str[k]);
            }
            printf("\n");
        }
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 50> gcc second.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 50> ./a.exe
Enter a string: hbhg6
h
hb
hbh
hbhg
hbhg6
b
bh
bhg
bhg6
h
hg
hg6
g
g6
6
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 50>
```

DAY-51

abc Q101 (Logic Enhancers)

Write a Program to take a sorted array(say nums[]) and an integer (say target) as inputs. The elements in the sorted array might be repeated. You need to print the first and last occurrence of the target and print the index of first and last occurrence. Print -1, -1 if the target is not present.

Follow-up(optional): Can you do it in O(log n) Time Complexity?

```c
#include <stdio.h>

int firstOccurrence(int nums[], int n, int target) {
    int low = 0, high = n - 1, result = -1;
    while (low <= high) {
        int mid = (low + high) / 2;
        if (nums[mid] == target) {
            result = mid;
            high = mid - 1;
        } else if (nums[mid] < target) {
            low = mid + 1;
        } else {
            high = mid - 1;
        }
    }
    return result;
}

int lastOccurrence(int nums[], int n, int target) {
    int low = 0, high = n - 1, result = -1;
    while (low <= high) {
        int mid = (low + high) / 2;
        if (nums[mid] == target) {
            result = mid;
            low = mid + 1;
        } else if (nums[mid] < target) {
            low = mid + 1;
        } else {
            high = mid - 1;
        }
    }
    return result;
}
```

```c
int main() {
    int n, target;
    printf("Enter size of array: ");
    scanf("%d", &n);

    int nums[n];
    printf("Enter sorted array elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &nums[i]);
    }

    printf("Enter target element: ");
    scanf("%d", &target);

    int first = firstOccurrence(nums, n, target);
    int last = lastOccurrence(nums, n, target);

    if (first == -1 || last == -1)
        printf("-1, -1\n");
    else
        printf("First occurrence index: %d\nLast occurrence index: %d\n", first, la

    return 0;
}
```

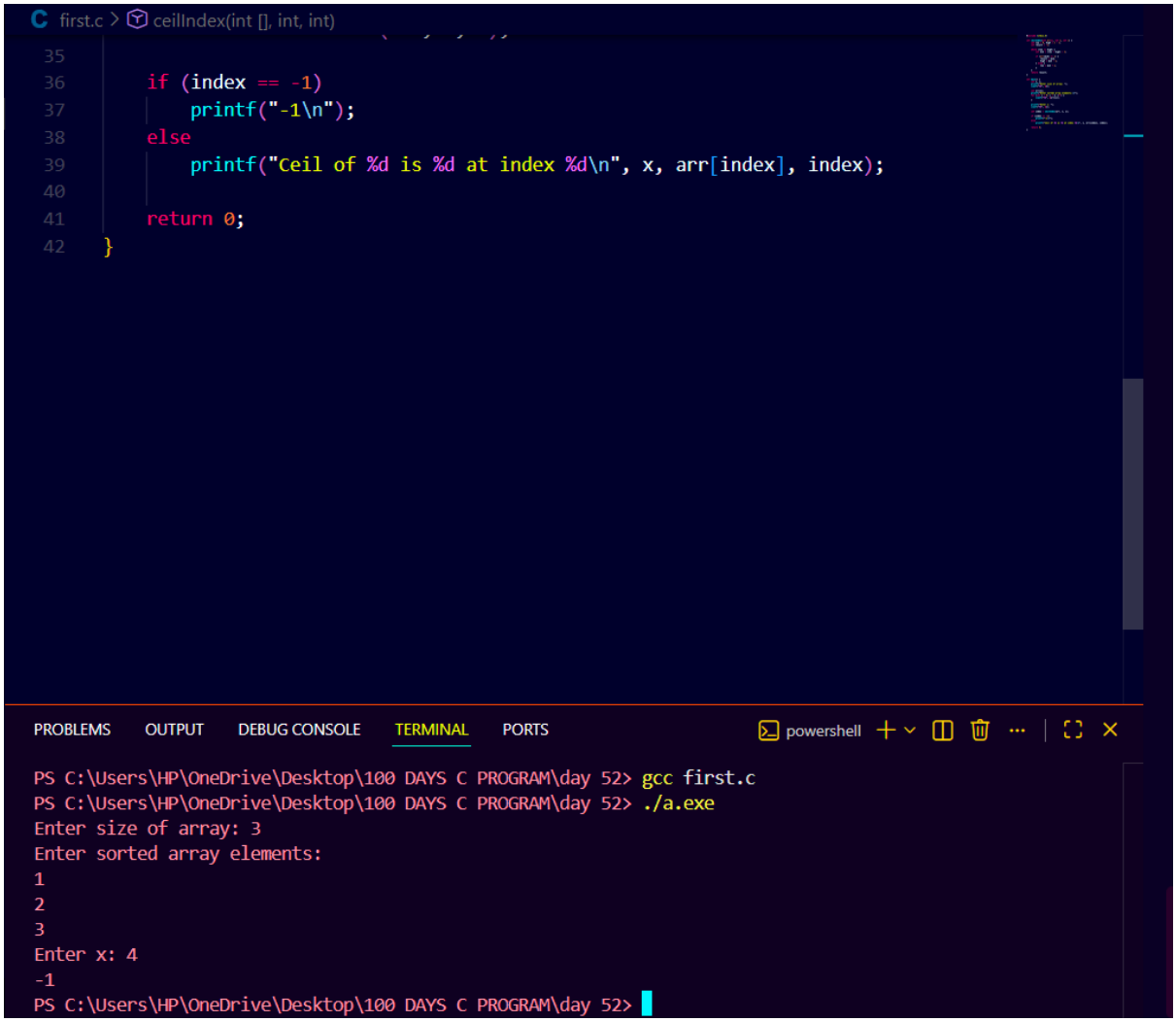
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS   powershell

PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 51> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 51> ./a.exe
Enter size of array: 2
Enter sorted array elements:
1
2
Enter target element: 1
First occurrence index: 0
Last occurrence index: 0
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 51>
```

DAY-52

abc Q102 (Logic Enhancers)

Write a Program to take a sorted array arr[] and an integer x as input, find the index (0-based) of the smallest element in arr[] that is greater than or equal to x and print it. This element is called the ceil of x. If such an element does not exist, print -1. Note: In case of multiple occurrences of ceil of x, return the index of the first occurrence.

Follow-up(optional): Can you do it in O(log n) Time Complexity?



DAY-53

Q103 (Logic Enhancers)

Write a Program to take an array of integers as input, calculate the pivot index of this array. The pivot index is the index where the sum of all the numbers strictly to the left of the index is equal to the sum of all the numbers strictly to the index's right. If the index is on the left edge of the array, then the left sum is 0 because there are no elements to the left. This also applies to the right edge of the array. Print the leftmost pivot index. If no such index exists, print -1.

Follow-up(optional): Try to solve this in O(n) TC

first.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Enter size of array: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter array elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }


    int totalSum = 0;
    for (int i = 0; i < n; i++) {
        totalSum += arr[i];
    }

    int leftSum = 0;
    int pivotIndex = -1;

    for (int i = 0; i < n; i++) {
        int rightSum = totalSum - leftSum - arr[i];
        if (leftSum == rightSum) {
            pivotIndex = i;
            break;
        }
        leftSum += arr[i];
    }

    printf("Pivot index: %d\n", pivotIndex);
    return 0;
}
```

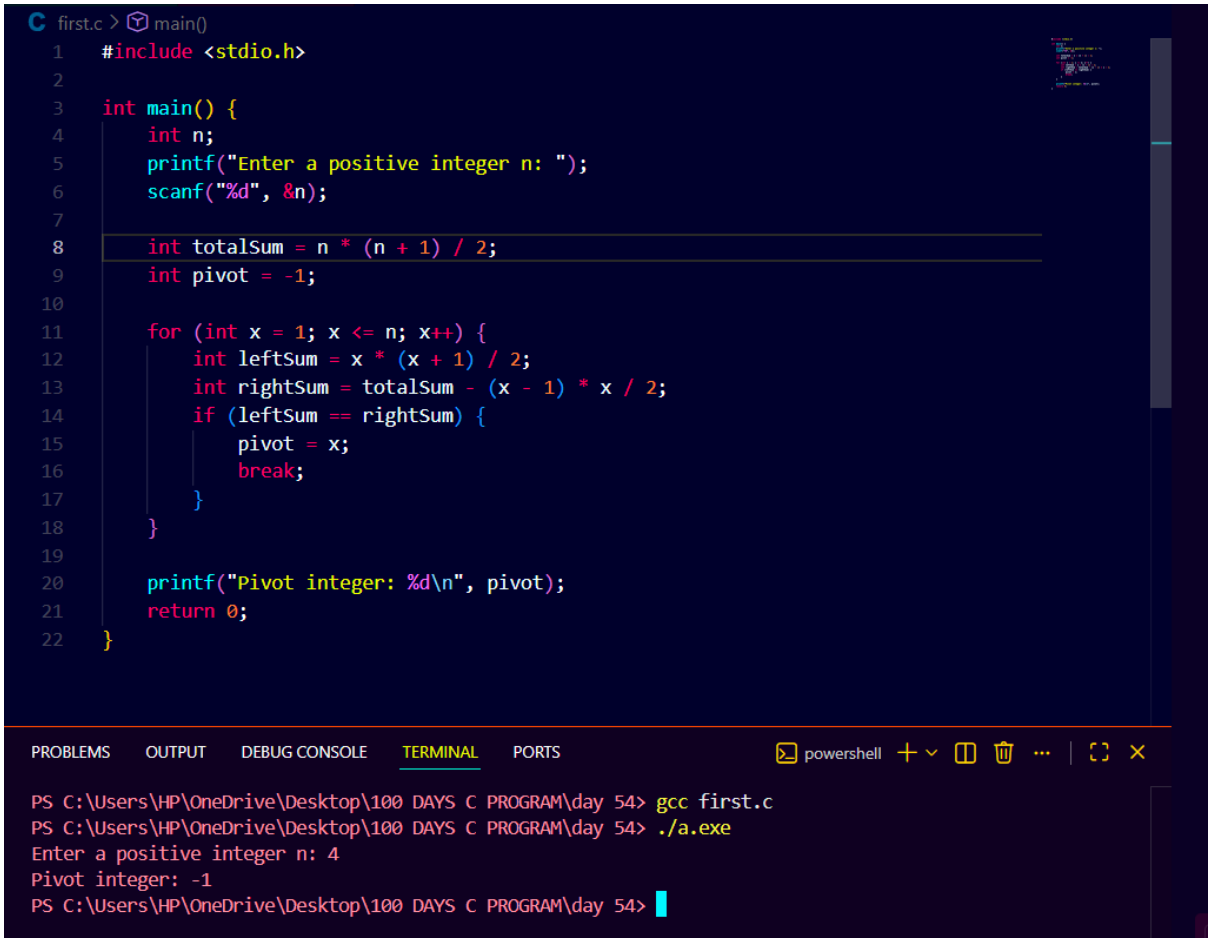
```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 53> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 53> ./a.exe
Enter size of array: 2
Enter array elements:
1
2
Pivot index: -1
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 53>
```

DAY-54

Q104 (Logic Enhancers)

Write a Program to take a positive integer n as input, and find the pivot integer x such that the sum of all elements between 1 and x inclusively equals the sum of all elements between x and n inclusively. Print the pivot integer x. If no such integer exists, print -1. Assume that it is guaranteed that there will be at most one pivot integer for the given input.

Follow-up 1 (optional): Can you do it in O(log n) Time Complexity? Follow-up 2 (optional): Can you do it in O(1) Time Complexity?



DAY-55

abc Q105 (Logic Enhancers)

Write a program to take an integer array nums of size n, and print the majority element. The majority element is the element that appears strictly more than ⌊n / 2⌋ times. Print -1 if no such element exists. Note: Majority Element is not necessarily the element that is present most number of times.

Follow-up (optional): Can you do it in O(n) Time Complexity?

first.c > main()

```c
#include <stdio.h>

int main() {
    int n;
    printf("Enter size of array: ");
    scanf("%d", &n);

    int nums[n];
    printf("Enter array elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &nums[i]);
    }
    int candidate = -1, count = 0;
    for (int i = 0; i < n; i++) {
        if (count == 0) {
            candidate = nums[i];
            count = 1;
        } else if (nums[i] == candidate) {
            count++;
        } else {
            count--;
        }
    }
    int freq = 0;
    for (int i = 0; i < n; i++) {
        if (nums[i] == candidate) {
            freq++;
        }
    }

    if (freq > n / 2)
        printf("Majority element: %d\n", candidate);
    else
        printf("-1\n");

    return 0;
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 55> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 55> ./a.exe
Enter size of array: 2
Enter array elements:
1
2
-1
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 55>
```

DAY-56

abc Q106 (Logic Enhancers)

Write a program to take an array arr[] of integers as input, the task is to find the next greater element for each element of the array in order of their appearance in the array. Next greater element of an element in the array is the nearest element on the right which is greater than the

current element. If there does not exist next greater of current element, then next greater element for current element is -1.

N.B:

- Print the output for each element in a comma separated fashion.

- Do not use Stack, use brute force approach (nested loop) to solve.

Try solving this using brute force (nested loop). No need of attempting the optimized stack-based solution.

first.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Enter size of array: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter array elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Next Greater Elements: ");
    for (int i = 0; i < n; i++) {
        int next = -1;
        for (int j = i + 1; j < n; j++) {
            if (arr[j] > arr[i]) {
                next = arr[j];
                break;
            }
        }
        printf("%d", next);
        if (i != n - 1) printf(",");
    }
    printf("\n");

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 56> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 56> ./a.exe
Enter size of array: 2
Enter array elements:
1
3
Next Greater Elements: 3,-1
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 56>
```

DAY-57

🔤 Q107 (Logic Enhancers)

Write a program to take an array arr[] of integers as input, the task is to find the previous greater element for each element of the array in order of their appearance in the array. Previous greater element of an element in the array is the nearest element on the left which is greater than the

current element. If there does not exist next greater of current element, then previous greater element for current element is -1.

N.B:

- Print the output for each element in a comma separated fashion.

- Do not use Stack, use brute force approach (nested loop) to solve.

Try solving this using brute force (nested loop). No need of attempting the optimized stack-based solution.

first.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Enter size of array: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter array elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Previous Greater Elements: ");
    for (int i = 0; i < n; i++) {
        int prev = -1;

        for (int j = i - 1; j >= 0; j--) {
            if (arr[j] > arr[i]) {
                prev = arr[j];
                break;
            }
        }
        printf("%d", prev);
        if (i != n - 1) printf(",");
    }
    printf("\n");

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 57> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 57> ./a.exe
Enter size of array: 2
Enter array elements:
1

2
Previous Greater Elements: -1,-1
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 57>
```

DAY-58

Q108 (Logic Enhancers)

Write a Program to take an integer array nums. Print an array answer such that answer[i] is equal to the product of all the elements of nums except nums[i]. The product of any prefix or suffix of nums is guaranteed to fit in a 32-bit integer.

Follow-up(optional): Can you write a code that runs in O(n) time and without using the division operation.

first.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Enter size of array: ");
    scanf("%d", &n);

    int nums[n];
    printf("Enter array elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &nums[i]);
    }
    int left[n], right[n], answer[n];
    left[0] = 1;
    for (int i = 1; i < n; i++) {
        left[i] = left[i - 1] * nums[i - 1];

    right[n - 1] = 1;
    for (int i = n - 2; i >= 0; i--) {
        right[i] = right[i + 1] * nums[i + 1];
    }
    for (int i = 0; i < n; i++) {
        answer[i] = left[i] * right[i];
    }

    printf("Answer array: ");
    for (int i = 0; i < n; i++) {
        printf("%d", answer[i]);
        if (i != n - 1) printf(",");
    }
    printf("\n");

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 58> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 58> ./a.exe
Enter size of array: 2
Enter array elements:
1
2
Answer array: 2,1
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 58>
```

DAY-59

abc Q109 (Logic Enhancers)

Write a program to take an integer array arr and an integer k as inputs. Print the maximum sum of all the subarrays of size k.

first.c > main()

```c
#include <stdio.h>

int main() {
    int n, k;
    printf("Enter size of array: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter array elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }printf("Enter k: ");
    scanf("%d", &k);

    if (k > n || k <= 0) {
        printf("Invalid k\n");
        return 0;
    }
    int windowSum = 0;
    for (int i = 0; i < k; i++) {
        windowSum += arr[i];
    }

    int maxSum = windowSum;
    for (int i = k; i < n; i++) {
        windowSum += arr[i] - arr[i - k];
        if (windowSum > maxSum) {
            maxSum = windowSum;
        }
    }

    printf("Maximum sum of subarrays of size %d: %d\n", k, maxSum);
    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 59> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 59> ./a.exe
Enter size of array: 1
Enter array elements:
1
Enter k: 2
Invalid k
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 59>
```

DAY-60

abc Q110 (Logic Enhancers)

Write a program to take an integer array arr and an integer k as inputs. The task is to find the maximum element in each subarray of size k moving from left to right. Print the maximum elements for each window separated by spaces as output.

first.c

```c
#include <stdio.h>

int main() {
    int n, k;
    printf("Enter size of array: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter array elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }printf("Enter k: ");
    scanf("%d", &k);

    if (k > n || k <= 0) {
        printf("Invalid k\n");
        return 0;
    } printf("Maximum elements in each subarray of size %d: ", k);
    for (int i = 0; i <= n - k; i++) {
        int max = arr[i];
        for (int j = i + 1; j < i + k; j++) {
            if (arr[j] > max) {
                max = arr[j];
            }
        }
        printf("%d", max);
        if (i != n - k) printf(" ");
    }
    printf("\n");

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 60> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 60> ./a.exe
Enter size of array: 2
Enter array elements:
1
2
Enter k: 2
Maximum elements in each subarray of size 2: 2
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 60>
```

DAY-61

🔤 Q111 (Logic Enhancers)

Write a program to take an integer array arr and an integer k as inputs. The task is to find the first negative integer in each subarray of size k moving from left to right. If no negative exists in a window, print "0" for that window. Print the results separated by spaces as output.

first.c > ...

```c
#include <stdio.h>
void firstNegativeInWindow(int arr[], int n, int k) {
    int result[n - k + 1];
    int front = 0, rear = -1;
    int queue[n];
    for (int i = 0; i < k; i++) {
        if (arr[i] < 0) {
            queue[++rear] = i;
        }
    }
    for (int i = k; i < n; i++) {

        if (front <= rear) {
            result[i - k] = arr[queue[front]];
        } else {
            result[i - k] = 0;
        }
        while (front <= rear && queue[front] <= i - k) {
            front++;
        }
        if (arr[i] < 0) {
            queue[++rear] = i;
        }
    }
    if (front <= rear) {
        result[n - k] = arr[queue[front]];
    } else {
        result[n - k] = 0;
    }
    for (int i = 0; i < n - k + 1; i++) {
        printf("%d ", result[i]);
    }
    printf("\n");
}

int main() {
    int arr[] = {12, -1, -7, 8, -15, 30, 16, 28};
    int n = sizeof(arr) / sizeof(arr[0]);
    int k = 3;

    firstNegativeInWindow(arr, n, k);

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 61> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 61> ./a.exe
-1 -1 -7 -15 -15 0
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 61>
```

DAY-62

Q112 (Logic Enhancers)

Write a program to take an integer array arr as input. The task is to find the maximum sum of any contiguous subarray using Kadane's algorithm. Print the maximum sum as output. If all elements are negative, print the largest (least negative) element.

first.c

```c
#include <stdio.h>

int maxSubarraySum(int arr[], int n) {
    int max_so_far = arr[0];
    int curr_max = arr[0];

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

        if (curr_max + arr[i] > arr[i])
            curr_max = curr_max + arr[i];
        else
            curr_max = arr[i];
        if (curr_max > max_so_far)
            max_so_far = curr_max;
    }

    return max_so_far;
}

int main() {
    int arr[] = { -2, -3, 4, -1, -2, 1, 5, -3 };
    int n = sizeof(arr) / sizeof(arr[0]);

    int max_sum = maxSubarraySum(arr, n);
    printf("%d\n", max_sum);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 62> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 62> ./a.exe
7
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 62>
```

DAY-63

Q113 (Logic Enhancers)

Write a program to take an integer array arr and an integer k as inputs. The task is to find the kth smallest element in the array. Print the kth smallest element as output.

```c
#include <stdio.h>

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
int partition(int arr[], int low, int high) {
    int pivot = arr[high];
    int i = low - 1;

    for (int j = low; j < high; j++) {
        if (arr[j] <= pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);
    return i + 1;
}
void quickSort(int arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

int main() {
    int arr[] = {12, 3, 5, 7, 19};
    int n = sizeof(arr) / sizeof(arr[0]);
    int k = 2;
    quickSort(arr, 0, n - 1);
    printf("%d\n", arr[k - 1]);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 63> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 63> ./a.exe
5
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 63>
```

DAY-64

🔤 Q114 (Logic Enhancers)

Write a program to take a string s as input. The task is to find the length of the longest substring without repeating characters. Print the length as output.

first.c > longestUniqueSubstring(char *)

```c
#include <stdio.h>
#include <string.h>

int longestUniqueSubstring(char *s) {
    int n = strlen(s);
    int visited[256];
    for (int i = 0; i < 256; i++) visited[i] = -1;

    int max_len = 0;
    int start = 0;

    for (int i = 0; i < n; i++) {
        if (visited[(unsigned char)s[i]] >= start) {
            start = visited[(unsigned char)s[i]] + 1;
        }
        visited[(unsigned char)s[i]] = i;
        int curr_len = i - start + 1;
        if (curr_len > max_len) {
            max_len = curr_len;
        }
    }

    return max_len;
}

int main() {
    char s[100];
    printf("Enter a string: ");
    scanf("%s", s);

    int result = longestUniqueSubstring(s);
    printf("%d\n", result);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 64> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 64> ./a.exe
Enter a string: yf5
3
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 64>
```

DAY-65

🔤 Q115 (Logic Enhancers)

Write a program to take two strings s and t as inputs (assume all characters are lowercase). The task is to determine if s and t are valid anagrams, meaning they contain the same characters with the same frequencies. Print "Anagram" if they are, otherwise "Not Anagram".

first.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char s[100], t[100];
    int freq[26] = {0};

    printf("Enter first string: ");
    scanf("%s", s);
    printf("Enter second string: ");
    scanf("%s", t);


    if (strlen(s) != strlen(t)) {
        printf("Not Anagram\n");
        return 0;
    }


    for (int i = 0; s[i] != '\0'; i++) {
        freq[s[i] - 'a']++;
    }
    for (int i = 0; t[i] != '\0'; i++) {
        freq[t[i] - 'a']--;
    }
    for (int i = 0; i < 26; i++) {
        if (freq[i] != 0) {
            printf("Not Anagram\n");
            return 0;
        }
    } printf("Anagram\n");
    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 65> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 65> ./a.exe
Enter first string: fr5
Enter second string: 56
Not Anagram
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 65>
```

DAY-66

abc Q116 (Logic Enhancers)

Write a program to take an integer array nums which contains only positive integers, and an integer target as inputs. The goal is to find two distinct indices i and j in the array such that nums[i] + nums[j] equals the target. Assume exactly one solution exists and return the indices in any order. Print the two indices separated by a space as output. If no solution exists, print "-1 -1".

first.c > twoSum(int [], int, int)

```c
#include <stdio.h>

void twoSum(int nums[], int n, int target) {
    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            if (nums[i] + nums[j] == target) {
                printf("%d %d\n", i, j);
                return;
            }
        }
    }

    printf("-1 -1\n");
}

int main() {
    int nums[] = {2, 7, 11, 15};
    int n = sizeof(nums) / sizeof(nums[0]);
    int target = 9;

    twoSum(nums, n, target);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 66> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 66> ./a.exe
0 1
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 66>
```

DAY-67

🔤 Q117 (Logic Enhancers)

Write a program to take two sorted arrays of size m and n as input. Merge both the arrays such that the merged array is also sorted. Print the merged array.

DAY-68

Q118 (Logic Enhancers)

Write a program to take an input array of size n. The arra y should contain all the integers between 0 to n except for one. Print that missing number

DAY-69

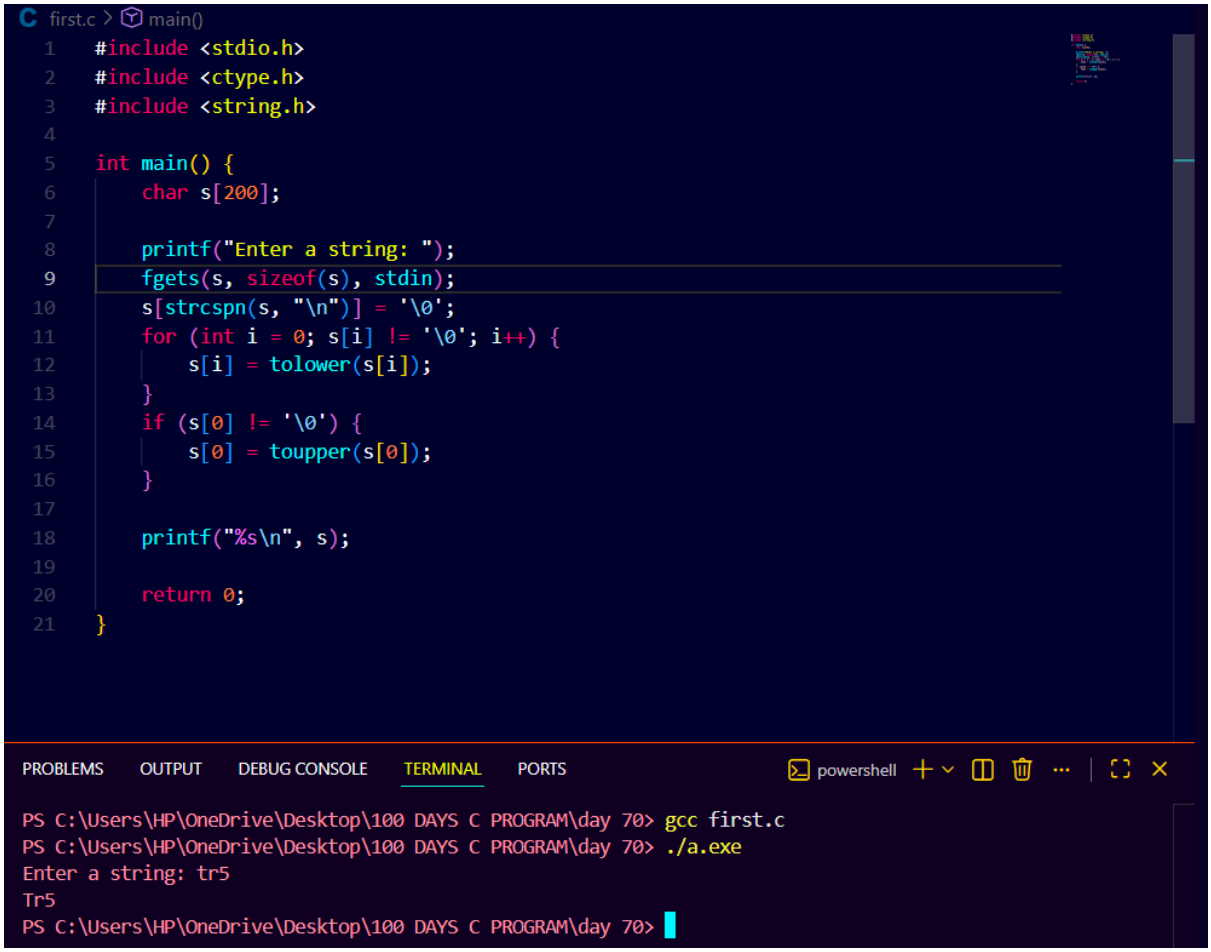
Q119 (Logic Enhancers)

Write a program to take an integer array as input. Only one element will be repeated. Print the repeated element. Try to find the result in one single iteration.

DAY-70

📁 Q120 (File Handling)

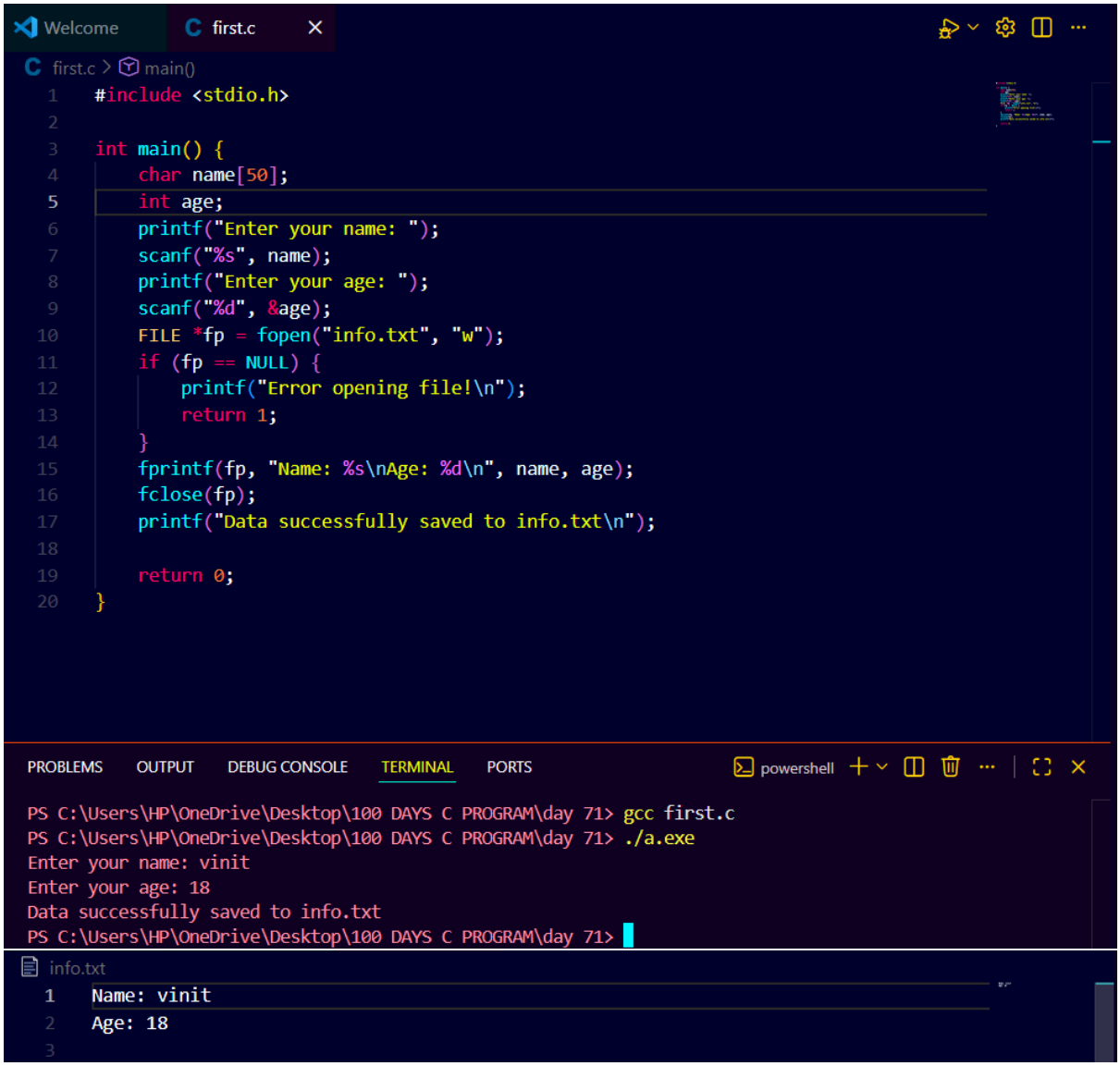
Write a program to take a string input. Change it to sentence case.



DAY-71

📁 Q121 (File Handling)

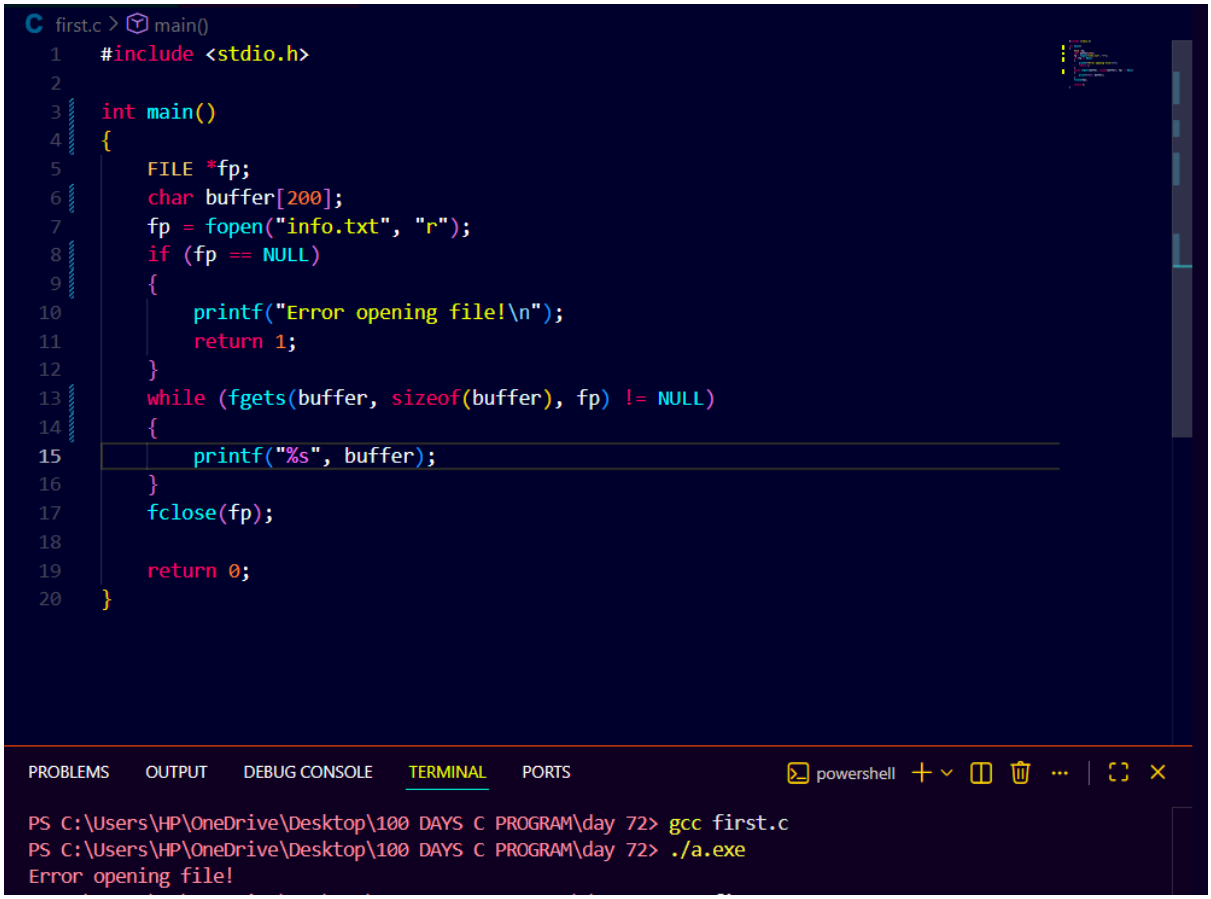
Write a C program that creates a text file named info.txt in write mode. The program should take the user's name and age as input, and write them to the file using fprintf(). After writing, display a message confirming that the data was successfully saved.



DAY-72

📁 Q122 (File Handling)

Write a C program that opens an existing file (e.g., info.txt) and reads its contents using fgets(). The program should print all the lines to the console until EOF (end of file) is reached.



DAY-73

📁 Q123 (File Handling)

Read a text file and count the total number of characters, words, and lines. A word is defined as a sequence of non-space characters separated by spaces or newlines.

first.c

```c
#include <stdio.h>
#include <ctype.h>

int main() {
    FILE *fp;
    char ch;
    int characters = 0, words = 0, lines = 0;
    int inWord = 0;
    fp = fopen("info.txt", "r");
    if (fp == NULL) {
        printf("Error opening file!\n");
        return 1;
    }
    while ((ch = fgetc(fp)) != EOF) {
        characters++;
        if (ch == '\n') {
            lines++;
        }
        if (isspace(ch)) {
            inWord = 0;
        } else {
            if (inWord == 0) {
                words++;
                inWord = 1;
            }
        }
    } fclose(fp);
    if (characters > 0 && lines == 0) {
        lines = 1;
    }

    printf("Characters: %d\n", characters);
    printf("Words: %d\n", words);
    printf("Lines: %d\n", lines);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 73> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 73> ./a.exe
Error opening file!
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 73>
```

DAY-74

📁 Q124 (File Handling)

Take two filenames from the user – a source file and a destination file. Copy all the content from the source file to the destination file using fgetc() and fputc().

first.c

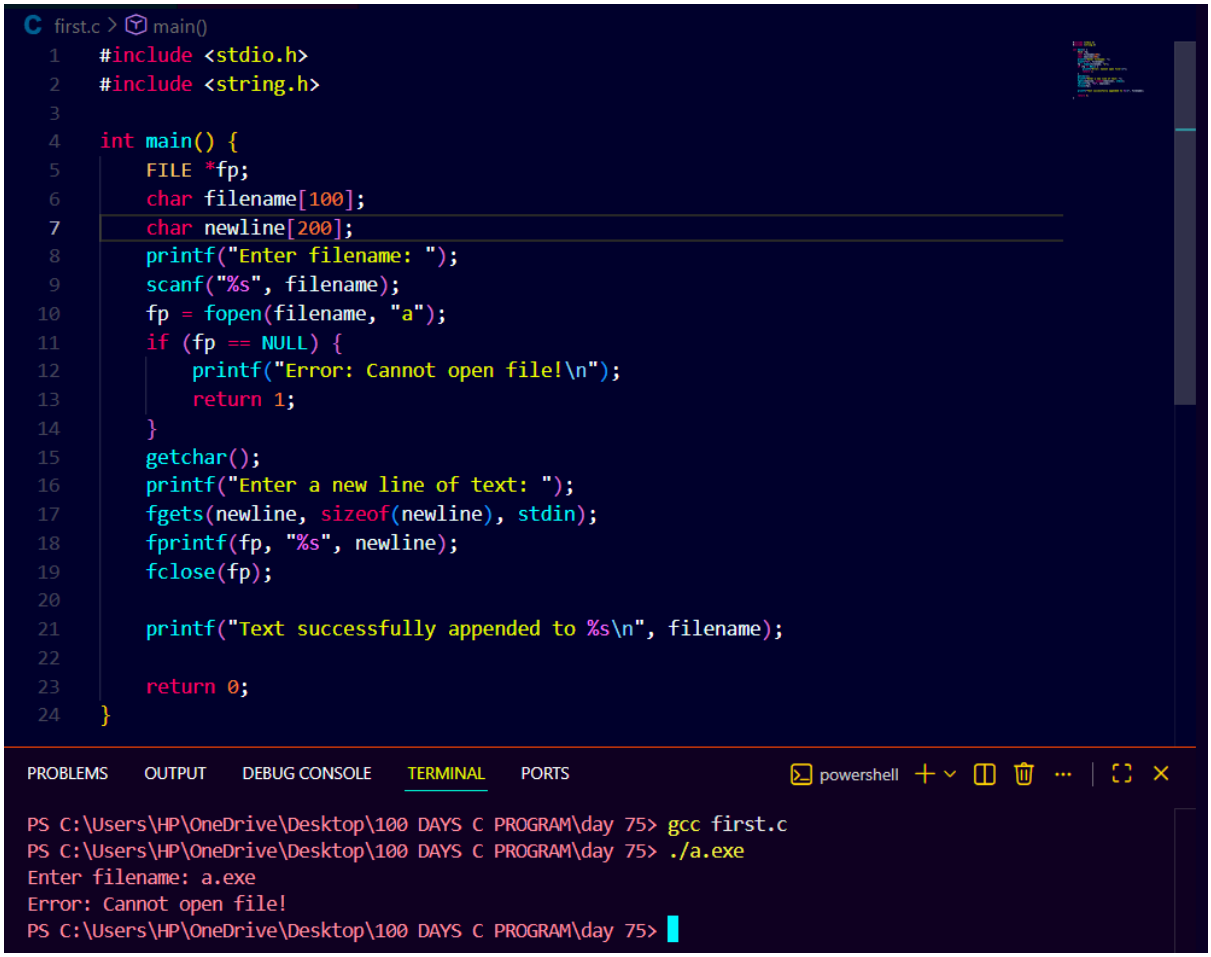
```c
#include <stdio.h>

int main() {
    char source[100], destination[100];
    FILE *src, *dest;
    int ch;
    printf("Enter source filename: ");
    scanf("%s", source);
    printf("Enter destination filename: ");
    scanf("%s", destination);
    src = fopen(source, "r");
    if (src == NULL) {
        printf("Error: Cannot open source file!\n");
        return 1;
    }
    dest = fopen(destination, "w");
    if (dest == NULL) {
        printf("Error: Cannot open destination file!\n");
        fclose(src);
        return 1;
    }
    while ((ch = fgetc(src)) != EOF) {
        fputc(ch, dest);
    }
    fclose(src);
    fclose(dest);

    printf("File copied successfully from %s to %s\n", source, destination);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 74> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 74> ./a.exe
Enter source filename: info.txt
Enter destination filename: day 72
Error: Cannot open source file!
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 74>
```

DAY-75

📁 Q125 (File Handling)

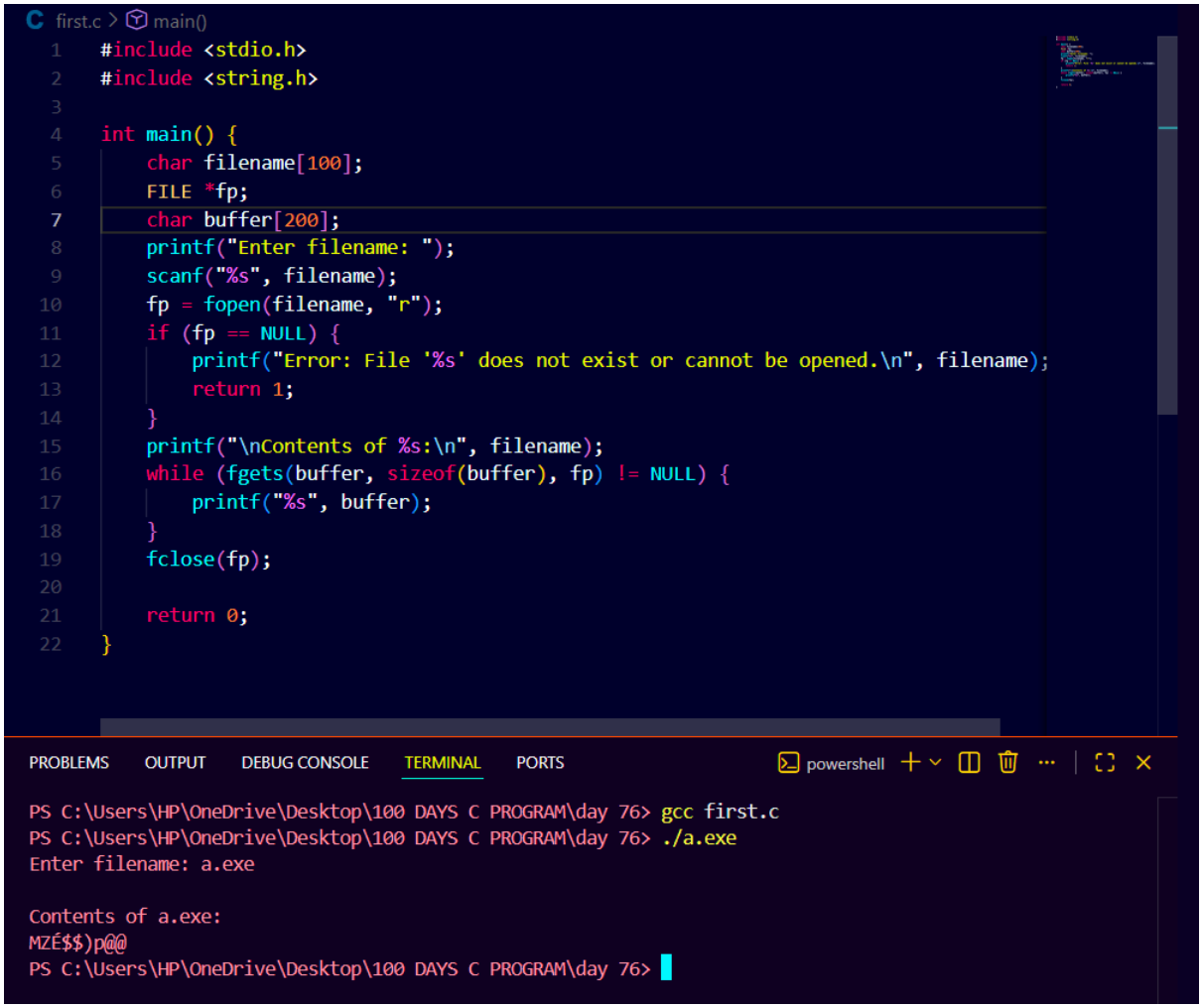
Open an existing file in append mode and allow the user to enter a new line of text. Append the text at the end without overwriting existing content.



DAY-76

📁 Q126 (File Handling)

Ask the user for a filename. Check if it exists by trying to open it in read mode. If the file pointer is NULL, print an error message; otherwise, read and display its content.



DAY-77

📁 Q127 (File Handling)

Write a program that reads text from input.txt, converts all lowercase letters to uppercase, and writes the result to output.txt.

first.c

```c
#include <stdio.h>
#include <ctype.h>

int main() {
    FILE *infile, *outfile;
    int ch;
    infile = fopen("input.txt", "r");
    if (infile == NULL) {
        printf("Error: Could not open input.txt\n");
        return 1;
    }
    outfile = fopen("output.txt", "w");
    if (outfile == NULL) {
        printf("Error: Could not open output.txt\n");
        fclose(infile);
        return 1;
    }
    while ((ch = fgetc(infile)) != EOF) {
        fputc(toupper(ch), outfile);
    }
    fclose(infile);
    fclose(outfile);

    printf("Conversion successful! Check output.txt for the result.\n");
    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 77> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 77> ./a.exe
Error: Could not open input.txt
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 77>
```

DAY-78

📁 Q128 (File Handling)

Read a text file and count how many vowels and consonants are in the file. Ignore digits and special characters.

first.c > ...

```c
#include <stdio.h>
#include <ctype.h>

int main() {
    FILE *file;
    char filename[100];
    char ch;
    int vowels = 0, consonants = 0;
    printf("Enter the filename: ");
    scanf("%s", filename);
    file = fopen(filename, "r");
    if (file == NULL) {
        printf("Could not open file %s\n", filename);
        return 1;
    }
    while ((ch = fgetc(file)) != EOF) {
        if (isalpha(ch)) {
            ch = tolower(ch);
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
                vowels++;
            else
                consonants++;
        }
    }

    fclose(file); printf("Vowels: %d\n", vowels);
    printf("Consonants: %d\n", consonants); return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 78> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 78> ./a.exe
Enter the filename: a.exe
Vowels: 0
Consonants: 2
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 78>
```

DAY-79

📁 Q129 (File Handling)

A file numbers.txt contains a list of integers separated by spaces. Read all integers, compute their sum and average, and print both.

first.c

```c

int main() {
    FILE *file;
    char filename[] = "numbers.txt";
    int num, sum = 0, count = 0;
    double average;
    file = fopen(filename, "r");
    if (file == NULL) {
        printf("Could not open file %s\n", filename);
        return 1;
    }
    while (fscanf(file, "%d", &num) == 1) {
        sum += num;
        count++;
    }

    fclose(file);

    if (count > 0) {
        average = (double)sum / count;
        printf("Sum = %d\n", sum);
        printf("Average = %.2f\n", average);
    } else {
        printf("No integers found in the file.\n");
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 79> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 79> ./a.exe
Could not open file numbers.txt
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 79>
```

DAY-80

📁 Q130 (File Handling)

Store multiple student records (name, roll number, marks) into a file using fprintf(). Then read them using fscanf() and display each record.

first.c > main()

```c
#include <stdio.h>

int main() {
    FILE *file;
    int n, i;
    char name[50];
    int roll;
    float marks;
    file = fopen("students.txt", "w");
    if (file == NULL) {
        printf("Error opening file for writing.\n");
        return 1;
    }printf("Enter number of students: ");
    scanf("%d", &n); for (i = 0; i < n; i++) {
        printf("\nEnter details of student %d\n", i + 1);
        printf("Name: ");
        scanf("%s", name);
        printf("Roll Number: ");
        scanf("%d", &roll);
        printf("Marks: ");
        scanf("%f", &marks);
        fprintf(file, "%s %d %.2f\n", name, roll, marks);
    }

    fclose(file);
    file = fopen("students.txt", "r");
    if (file == NULL) {
        printf("Error opening file for reading.\n");
        return 1;
    }printf("\n--- Student Records ---\n");
    while (fscanf(file, "%s %d %f", name, &roll, &marks) == 3) {
        printf("Name: %s | Roll: %d | Marks: %.2f\n", name, roll, marks);
    }

    fclose(file);return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 80> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 80> ./a.exe
Enter number of students: 4

Enter details of student 1
Name: swastik
Roll Number: 5
Marks: 25

Enter details of student 2
Name: gautam
Roll Number: 6
Marks: 90

Enter details of student 3
Name: yash
Roll Number: 7
Marks: 97

Enter details of student 4
Name: guru
Roll Number: 2
Marks: 100

--- Student Records ---
Name: swastik | Roll: 5 | Marks: 25.00
Name: gautam | Roll: 6 | Marks: 90.00
Name: yash | Roll: 7 | Marks: 97.00
Name: guru | Roll: 2 | Marks: 100.00
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 80>
```

DAY-81

🧱 Q131 (Enum)

Create an enumeration for days (SUNDAY to SATURDAY) and print each day with its integer value.

first.c

```c
#include <stdio.h>

enum Days {
    SUNDAY,
    MONDAY,
    TUESDAY,
    WEDNESDAY,
    THURSDAY,
    FRIDAY,
    SATURDAY
};

int main() {
    enum Days day;

    printf("Days of the Week with Integer Values:\n");
    for (day = SUNDAY; day <= SATURDAY; day++) {
        switch(day) {
            case SUNDAY:    printf("SUNDAY = %d\n", day); break;
            case MONDAY:    printf("MONDAY = %d\n", day); break;
            case TUESDAY:   printf("TUESDAY = %d\n", day); break;
            case WEDNESDAY: printf("WEDNESDAY = %d\n", day); break;
            case THURSDAY:  printf("THURSDAY = %d\n", day); break;
            case FRIDAY:    printf("FRIDAY = %d\n", day); break;
            case SATURDAY:  printf("SATURDAY = %d\n", day); break;
        }
    }

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 81> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 81> ./a.exe
Days of the Week with Integer Values:
SUNDAY = 0
MONDAY = 1
TUESDAY = 2
WEDNESDAY = 3
THURSDAY = 4
FRIDAY = 5
SATURDAY = 6
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 81>
```

DAY-82

Q132 (Enum)

Define an enum for traffic lights (RED, YELLOW, GREEN) and print 'Stop', 'Wait', or 'Go' based on its value.

first.c > main()

```c
#include <stdio.h>

enum TrafficLight { RED, YELLOW, GREEN };

int main() {
    enum TrafficLight signal;


    for (signal = RED; signal <= GREEN; signal++) {
        switch(signal) {
            case RED:
                printf("Signal: RED -> Stop\n");
                break;
            case YELLOW:
                printf("Signal: YELLOW -> Wait\n");
                break;
            case GREEN:
                printf("Signal: GREEN -> Go\n");
                break;
        }
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 82> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 82> ./a.exe
Signal: RED -> Stop
Signal: YELLOW -> Wait
Signal: GREEN -> Go
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 82>
```

DAY-83

Q133 (Enum)

Create an enum for months and print how many days each month has.

first.c

```c
#include <stdio.h>

enum Months {
    JANUARY = 1,
    FEBRUARY,
    MARCH,
    APRIL,
    MAY,
    JUNE,
    JULY,
    AUGUST,
    SEPTEMBER,
    OCTOBER,
    NOVEMBER,
    DECEMBER
};
int main() {
    enum Months month;printf("Days in each month:\n");
    for (month = JANUARY; month <= DECEMBER; month++) {
        switch(month) {
            case JANUARY:   printf("January   -> 31 days\n"); break;
            case FEBRUARY:  printf("February  -> 28/29 days\n"); break;
            case MARCH:     printf("March     -> 31 days\n"); break;
            case APRIL:     printf("April     -> 30 days\n"); break;
            case MAY:       printf("May       -> 31 days\n"); break;
            case JUNE:      printf("June      -> 30 days\n"); break;
            case JULY:      printf("July      -> 31 days\n"); break;
            case AUGUST:    printf("August    -> 31 days\n"); break;
            case SEPTEMBER: printf("September -> 30 days\n"); break;
            case OCTOBER:   printf("October   -> 31 days\n"); break;
            case NOVEMBER:  printf("November  -> 30 days\n"); break;
            case DECEMBER:  printf("December  -> 31 days\n"); break;
        }
    }return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 83> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 83> ./a.exe
Days in each month:
January   -> 31 days
February  -> 28/29 days
March     -> 31 days
April     -> 30 days
May       -> 31 days
June      -> 30 days
July      -> 31 days
April     -> 30 days
May       -> 31 days
June      -> 30 days
April     -> 30 days
May       -> 31 days
April     -> 30 days
April     -> 30 days
April     -> 30 days
May       -> 31 days
June      -> 30 days
July      -> 31 days
August    -> 31 days
September -> 30 days
October   -> 31 days
November  -> 30 days
December  -> 31 days
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 83>
```

DAY-84

🧱 Q134 (Enum)

Define an enum with SUCCESS, FAILURE, and TIMEOUT, and print messages accordingly.

first.c

```c
#include <stdio.h>

enum Status { SUCCESS, FAILURE, TIMEOUT };

int main() {
    enum Status result;

    for (result = SUCCESS; result <= TIMEOUT; result++) {
        switch(result) {
            case SUCCESS:
                printf("Status: SUCCESS -> Operation completed successfully.\n");
                break;
            case FAILURE:
                printf("Status: FAILURE -> Operation failed.\n");
                break;
            case TIMEOUT:
                printf("Status: TIMEOUT -> Operation timed out.\n");
                break;
        }
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 84> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 84> ./a.exe
Status: SUCCESS -> Operation completed successfully.
Status: FAILURE -> Operation failed.
Status: TIMEOUT -> Operation timed out.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 84>
```

DAY-85

🧱 Q135 (Enum)

Assign explicit values starting from 10 and print them.

first.c

```c
#include <stdio.h>

enum Status {
    SUCCESS = 10,
    FAILURE,
    TIMEOUT
};

int main() {
    enum Status s;

    printf("Enum values:\n");
    for (s = SUCCESS; s <= TIMEOUT; s++) {
        switch(s) {
            case SUCCESS:
                printf("SUCCESS = %d\n", s);
                break;
            case FAILURE:
                printf("FAILURE = %d\n", s);
                break;
            case TIMEOUT:
                printf("TIMEOUT = %d\n", s);
                break;
        }
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 85> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 85> ./a.exe
Enum values:
SUCCESS = 10
FAILURE = 11
TIMEOUT = 12
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 85>
```

DAY-86

🧱 Q136 (Enum)

Use enum to represent menu choices (ADD, SUBTRACT, MULTIPLY) and perform operations using switch.

first.c

```c
#include <stdio.h>

enum Menu { ADD = 1, SUBTRACT, MULTIPLY };

int main() {
    enum Menu choice;
    int a, b, result;

    printf("Enter two integers: ");
    scanf("%d %d", &a, &b);

    printf("\nMenu:\n");
    printf("1. ADD\n");
    printf("2. SUBTRACT\n");
    printf("3. MULTIPLY\n");
    printf("Enter your choice (1-3): ");
    scanf("%d", (int*)&choice);

    switch(choice) {
        case ADD:
            result = a + b;
            printf("Result of addition: %d\n", result);
            break;
        case SUBTRACT:
            result = a - b;
            printf("Result of subtraction: %d\n", result);
            break;
        case MULTIPLY:
            result = a * b;
            printf("Result of multiplication: %d\n", result);
            break;
        default:
            printf("Invalid choice!\n");
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 86> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 86> ./a.exe
Enter two integers: 45
76

Menu:
1. ADD
2. SUBTRACT
3. MULTIPLY
Enter your choice (1-3): 3
Result of multiplication: 3420
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 86>
```

DAY-87

🧱 Q137 (Enum)

Create an enum for user roles (ADMIN, USER, GUEST) and display messages based on role.

```c
#include <stdio.h>

enum UserRole { ADMIN = 1, USER, GUEST };

int main() {
    enum UserRole role;

    printf("Select role:\n");
    printf("1. ADMIN\n");
    printf("2. USER\n");
    printf("3. GUEST\n");
    printf("Enter your choice (1-3): ");
    scanf("%d", (int*)&role);

    switch(role) {
        case ADMIN:
            printf("Welcome, Admin! You have full access.\n");
            break;
        case USER:
            printf("Welcome, User! You have limited access.\n");
            break;
        case GUEST:
            printf("Welcome, Guest! You can browse only.\n");
            break;
        default:
            printf("Invalid role selected.\n");
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 87> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 87> ./a.exe
Select role:
1. ADMIN
2. USER
3. GUEST
Enter your choice (1-3): 3
Welcome, Guest! You can browse only.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 87>
```

DAY-88

Q138 (Enum)

Print all enum names and integer values using a loop.

```c
#include <stdio.h>

enum Days {
    SUNDAY = 0,
    MONDAY,
    TUESDAY,
    WEDNESDAY,
    THURSDAY,
    FRIDAY,
    SATURDAY
};

int main() {

    const char *dayNames[] = {
        "SUNDAY", "MONDAY", "TUESDAY", "WEDNESDAY",
        "THURSDAY", "FRIDAY", "SATURDAY"
    };

    printf("Enum names with their integer values:\n");
    for (int i = SUNDAY; i <= SATURDAY; i++) {
        printf("%s = %d\n", dayNames[i], i);
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 88> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 88> ./a.exe
Enum names with their integer values:
SUNDAY = 0
MONDAY = 1
TUESDAY = 2
WEDNESDAY = 3
THURSDAY = 4
FRIDAY = 5
SATURDAY = 6
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 88>
```

DAY-89

Q139 (Enum)

Show that enums store integers by printing assigned values.

first.c

```c
#include <stdio.h>

enum Status {
    SUCCESS = 10,
    FAILURE = 20,
    TIMEOUT = 30
};

int main() {
    enum Status s;

    printf("Enum values:\n");
    printf("SUCCESS = %d\n", SUCCESS);
    printf("FAILURE = %d\n", FAILURE);
    printf("TIMEOUT = %d\n", TIMEOUT);


    s = SUCCESS;
    printf("\nAssigned variable 's' = %d\n", s);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 89> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 89> ./a.exe
Enum values:
SUCCESS = 10
FAILURE = 20
TIMEOUT = 30

Assigned variable 's' = 10
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 89>
```

DAY-90

🧱 Q140 (Enum)

Define a struct with enum Gender and print person's gender.

first.c > ...

```c
#include <stdio.h>
enum Gender { MALE, FEMALE, OTHER };
struct Person {
    char name[50];
    int age;
    enum Gender gender;
};

int main() {
    struct Person p1;
    printf("Enter name: ");
    scanf("%s", p1.name);
    printf("Enter age: ");
    scanf("%d", &p1.age);

    printf("Select gender (0 = MALE, 1 = FEMALE, 2 = OTHER): ");
    scanf("%d", (int*)&p1.gender);
    printf("\n--- Person Details ---\n");
    printf("Name: %s\n", p1.name);
    printf("Age: %d\n", p1.age);
    switch(p1.gender) {
        case MALE:
            printf("Gender: Male\n");
            break;
        case FEMALE:
            printf("Gender: Female\n");
            break;
        case OTHER:
            printf("Gender: Other\n");
            break;
        default:
            printf("Invalid gender selected.\n");
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 90> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 90> ./a.exe
Enter name: vinit kumar
Enter age: Select gender (0 = MALE, 1 = FEMALE, 2 = OTHER):
--- Person Details ---
Name: vinit
Age: 0
Invalid gender selected.
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 90>
```

DAY-91

🧱 Q141 (Enum)

Define a structure Student with name, roll_no, and marks, then read and print one student's data.

first.c

```c
#include <stdio.h>

struct Student {
    char name[50];
    int roll_no;
    float marks;
};

int main() {
    struct Student s1;
    printf("Enter student name: ");
    scanf("%s", s1.name);

    printf("Enter roll number: ");
    scanf("%d", &s1.roll_no);

    printf("Enter marks: ");
    scanf("%f", &s1.marks);
    printf("\n--- Student Details ---\n");
    printf("Name: %s\n", s1.name);
    printf("Roll Number: %d\n", s1.roll_no);
    printf("Marks: %.2f\n", s1.marks);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 91> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 91> ./a.exe
Enter student name: vinit
Enter roll number: 6
Enter marks: 100

--- Student Details ---
Name: vinit
Roll Number: 6
Marks: 100.00
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 91>
```

DAY-92

🧱 Q142 (Enum)

Store details of 5 students in an array of structures and print all.

first.c

```c
#include <stdio.h>

struct Student {
    char name[50];
    int roll_no;
    float marks;
};

int main() {
    struct Student students[5];
    int i;
    for (i = 0; i < 5; i++) {
        printf("\nEnter details of student %d\n", i + 1);
        printf("Name: ");
        scanf("%s", students[i].name);
        printf("Roll Number: ");
        scanf("%d", &students[i].roll_no);
        printf("Marks: ");
        scanf("%f", &students[i].marks);
    }
    printf("\n--- Student Records ---\n");
    for (i = 0; i < 5; i++) {
        printf("Student %d:\n", i + 1);
        printf("Name: %s\n", students[i].name);
        printf("Roll Number: %d\n", students[i].roll_no);
        printf("Marks: %.2f\n\n", students[i].marks);
    }

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 92> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 92> ./a.exe

Enter details of student 1
Name: vinit
Roll Number: 3
Marks: 98

Enter details of student 2
Name: swastik
Roll Number: 4
Marks: 3

Enter details of student 3
Name: gautam
Roll Number: 5
Marks: 47

Enter details of student 4
Name: hardik
Roll Number: 7
Marks: 89

Enter details of student 5
Name: bipin
Roll Number: 9
Marks: 76

--- Student Records ---
Student 1:
Name: vinit
Roll Number: 3
Marks: 98.00

Student 2:
Name: swastik
Roll Number: 4
Marks: 3.00

Student 3:
```

DAY-93

🧱 Q143 (Enum)

Find and print the student with the highest marks.

first.c

```c
#include <stdio.h>
struct Student {
    char name[50];
    int roll_no;
    float marks;
};

int main() {
    struct Student students[5];
    int i, maxIndex = 0;
    for (i = 0; i < 5; i++) {
        printf("\nEnter details of student %d\n", i + 1);
        printf("Name: ");
        scanf("%s", students[i].name);
        printf("Roll Number: ");
        scanf("%d", &students[i].roll_no);
        printf("Marks: ");
        scanf("%f", &students[i].marks);
    }
    for (i = 1; i < 5; i++) {
        if (students[i].marks > students[maxIndex].marks) {
            maxIndex = i;
        }
    }
    printf("\n--- Student with Highest Marks ---\n");
    printf("Name: %s\n", students[maxIndex].name);
    printf("Roll Number: %d\n", students[maxIndex].roll_no);
    printf("Marks: %.2f\n", students[maxIndex].marks);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 93> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 93> ./a.exe

Enter details of student 1
Name: virat
Roll Number: 1
Marks: 1

Enter details of student 2
Name: alex
Roll Number: 2
Marks: 2

Enter details of student 3
Name: papu
Roll Number: 4
Marks: 4

Enter details of student 4
Name: tanisq
Roll Number: 6
Marks: 6

Enter details of student 5
Name: bipin
Roll Number: 3
Marks: 3

--- Student with Highest Marks ---
Name: tanisq
Roll Number: 6
Marks: 6.00
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 93>
```

DAY-94

🧱 Q144 (Enum)

Write a function that accepts a structure as parameter and prints its members.

first.c

```c
#include <stdio.h>

struct Student {
    char name[50];
    int roll_no;
    float marks;
};


void printStudent(struct Student s) {
    printf("\n--- Student Details ---\n");
    printf("Name: %s\n", s.name);
    printf("Roll Number: %d\n", s.roll_no);
    printf("Marks: %.2f\n", s.marks);
}

int main() {
    struct Student s1;
    printf("Enter student name: ");
    scanf("%s", s1.name);
    printf("Enter roll number: ");
    scanf("%d", &s1.roll_no);
    printf("Enter marks: ");
    scanf("%f", &s1.marks);
    printStudent(s1);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 94> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 94> ./a.exe
Enter student name: papu
Enter roll number: 1
Enter marks: 1

--- Student Details ---
Name: papu
Roll Number: 1
Marks: 1.00
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 94>
```

DAY-95

🧱 Q145 (Enum)

Return a structure containing top student's details from a function.

first.c

```c
#include <stdio.h>
#include <string.h>
struct Student {
    int id;
    char name[50];
    float marks;
};


struct Student getTopStudent() {
    struct Student top;

    top.id = 101;
    strcpy(top.name, "Amit Sharma");
    top.marks = 95.6;

    return top;
}

int main() {
    struct Student s = getTopStudent();

    printf("Top Student Details:\n");
    printf("ID: %d\n", s.id);
    printf("Name: %s\n", s.name);
    printf("Marks: %.2f\n", s.marks);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 95> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 95> ./a.exe
Top Student Details:
ID: 101
Name: Amit Sharma
Marks: 95.60
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 95>
```

DAY-96

🧱 Q146 (Enum)

Create Employee structure with nested Date structure for joining date and print details.

```c
#include <stdio.h>
#include <string.h>
struct Date {
    int day;
    int month;
    int year;
};
struct Employee {
    int id;
    char name[50];
    float salary;
    struct Date joiningDate;
};
int main() {

    struct Employee emp;

    emp.id = 1001;
    strcpy(emp.name, "Ravi Kumar");
    emp.salary = 55000.75;

    emp.joiningDate.day = 15;
    emp.joiningDate.month = 7;
    emp.joiningDate.year = 2021;

    // Print Employee details
    printf("Employee Details:\n");
    printf("ID: %d\n", emp.id);
    printf("Name: %s\n", emp.name);
    printf("Salary: %.2f\n", emp.salary);
    printf("Joining Date: %02d-%02d-%d\n",
           emp.joiningDate.day,
           emp.joiningDate.month,
           emp.joiningDate.year);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 96> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 96> ./a.exe
Employee Details:
ID: 1001
Name: Ravi Kumar
Salary: 55000.75
Joining Date: 15-07-2021
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 96>
```

DAY-97

Q147 (Enum)

Store employee data in a binary file using fwrite() and read using fread().

first.c

```c
#include <stdio.h>
#include <string.h>
struct Date {
    int day;
    int month;
    int year;
};
struct Employee {
    int id; char name[50];
    float salary; struct Date joiningDate;
};
int main() {
    struct Employee emp1, emp2;  emp1.id = 1001;
    strcpy(emp1.name, "Ravi Kumar");
    emp1.salary = 55000.75; emp1.joiningDate.day = 15;
    emp1.joiningDate.month = 7;emp1.joiningDate.year = 2021;
    FILE *fp = fopen("employee.dat", "wb");
    if (fp == NULL) {
        printf("Error opening file for writing!\n");  return 1;
    }
    fwrite(&emp1, sizeof(struct Employee), 1, fp);
    fclose(fp);
    fp = fopen("employee.dat", "rb");
    if (fp == NULL) {
        printf("Error opening file for reading!\n");
        return 1;
    }fread(&emp2, sizeof(struct Employee), 1, fp);fclose(fp); printf("Employee Details (Read from File):\n");
    printf("ID: %d\n", emp2.id);
    printf("Name: %s\n", emp2.name);
    printf("Salary: %.2f\n", emp2.salary);
    printf("Joining Date: %02d-%02d-%d\n",
           emp2.joiningDate.day,
           emp2.joiningDate.month, emp2.joiningDate.year);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 97> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 97> ./a.exe
Employee Details (Read from File):
ID: 1001
Name: Ravi Kumar
Salary: 55000.75
Joining Date: 15-07-2021
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 97>
```

DAY-98

Q148 (Enum)

Take two structs as input and check if they are identical.

```c
#include <stdio.h>
#include <string.h>


struct Student {
    int id;
    char name[50];
    float marks;
};


int areIdentical(struct Student s1, struct Student s2) {
    if (s1.id == s2.id &&
        strcmp(s1.name, s2.name) == 0 &&
        s1.marks == s2.marks) {
        return 1;
    }
    return 0;
}

int main() {
    struct Student a, b;


    printf("Enter details for Student A:\n");
    printf("ID: ");
    scanf("%d", &a.id);
    printf("Name: ");
    scanf("%s", a.name);
    printf("Marks: ");
    scanf("%f", &a.marks);


    printf("\nEnter details for Student B:\n");
    printf("ID: ");
    scanf("%d", &b.id);
    printf("Name: ");
    scanf("%s", b.name);
    printf("Marks: ");
    scanf("%f", &b.marks);


    if (areIdentical(a, b)) {
        printf("\nBoth structs are IDENTICAL.\n");
    } else {
        printf("\nStructs are DIFFERENT.\n");
    }

    return 0;
}
```

DAY-99

🧱 Q149 (Enum) Use malloc() to allocate structure memory dynamically and print details.

first.c > ...

```c
#include <stdio.h>


struct Student {
    int id;
    char name[50];
    float marks;
};

int main() {

    struct Student *s1 = (struct Student *)malloc(sizeof(struct Student));
    struct Student *s2 = (struct Student *)malloc(sizeof(struct Student));

    if (s1 == NULL || s2 == NULL) {
        printf("Memory allocation failed!\n");
        return 1;
    }


    printf("Enter details for Student A:\n");
    printf("ID: ");
    scanf("%d", &s1->id);
    printf("Name: ");
    scanf("%s", s1->name);
    printf("Marks: ");
    scanf("%f", &s1->marks);


    printf("\nEnter details for Student B:\n");
    printf("ID: ");
    scanf("%d", &s2->id);
    printf("Name: ");
    scanf("%s", s2->name);
    printf("Marks: ");
    scanf("%f", &s2->marks);

```

```c

    printf("\n--- Student A ---\n");
    printf("ID: %d\n", s1->id);
    printf("Name: %s\n", s1->name);
    printf("Marks: %.2f\n", s1->marks);

    printf("\n--- Student B ---\n");
    printf("ID: %d\n", s2->id);
    printf("Name: %s\n", s2->name);
    printf("Marks: %.2f\n", s2->marks);


    free(s1);
    free(s2);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 99> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 99> ./a.exe
Enter details for Student A:
ID: 43
Name: papu
Marks: 56

Enter details for Student B:
ID: rahul
Name: Marks: 5

--- Student A ---
ID: 43
Name: papu
Marks: 56.00

--- Student B ---
ID: 13834016
Name: rahul
Marks: 5.00
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 99>
```

DAY-100

🧱 Q150 (Enum)

Use pointer to struct to modify and display data using -> operator.

first.c > main()

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct Student {
    int id;
    char name[50];
    float marks;
};
int main() {
    struct Student *s = (struct Student *)malloc(sizeof(struct Student));
    if (s == NULL) {
        printf("Memory allocation failed!\n");
        return 1;
    }
    s->id = 101;
    strcpy(s->name, "Vinit");
    s->marks = 92.5;


    printf("Student Details:\n");
    printf("ID: %d\n", s->id);
    printf("Name: %s\n", s->name);
    printf("Marks: %.2f\n", s->marks);


    s->marks = 95.0;
    printf("\nAfter updating marks:\n");
    printf("ID: %d\n", s->id);
    printf("Name: %s\n", s->name);
    printf("Marks: %.2f\n", s->marks);


    free(s);

    return 0;
}
```

```
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 100> gcc first.c
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 100> ./a.exe
Student Details:
ID: 101
Name: Vinit
Marks: 92.50

After updating marks:
ID: 101
Name: Vinit
Marks: 95.00
PS C:\Users\HP\OneDrive\Desktop\100 DAYS C PROGRAM\day 100>
```