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

abc 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

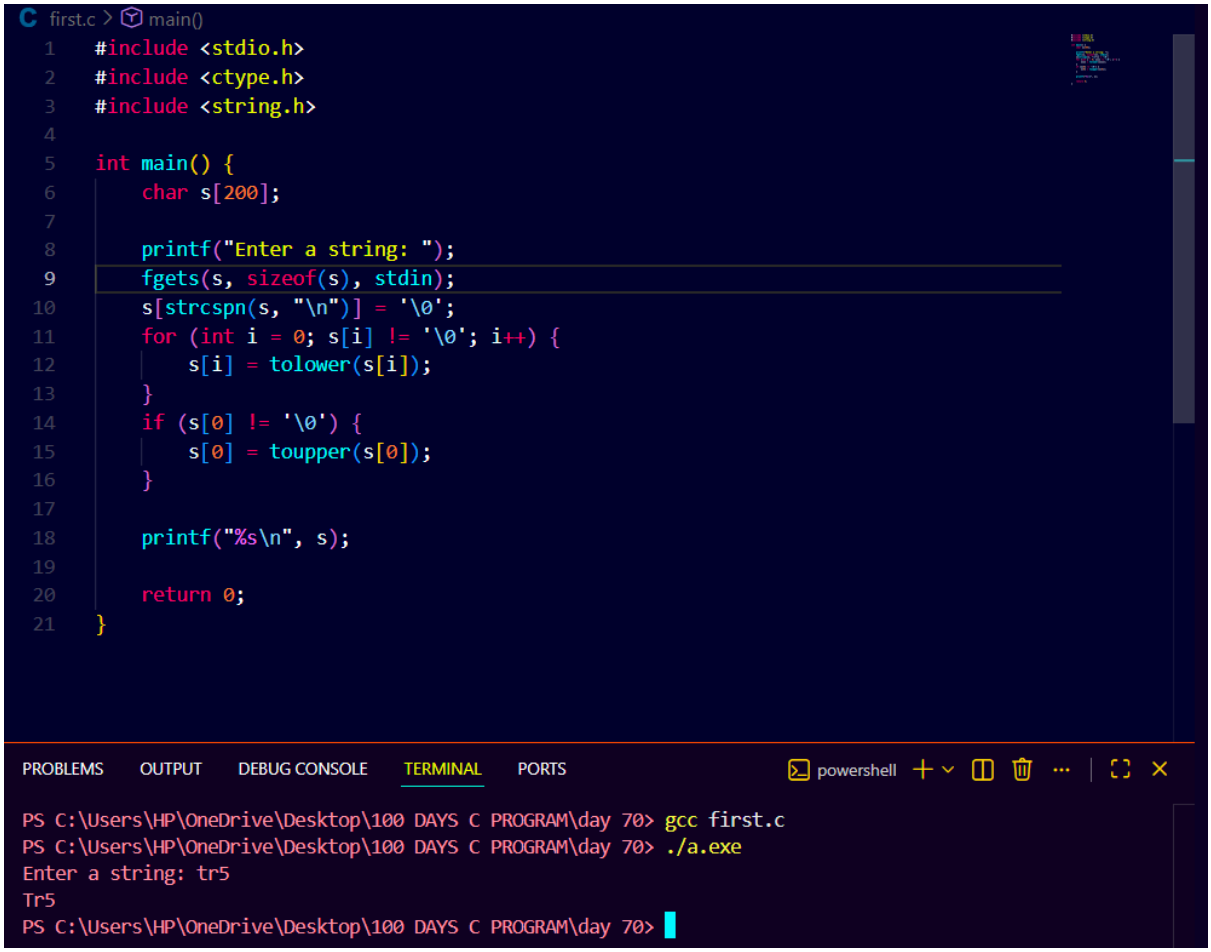
Q119 (Logic Enhancers)

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)

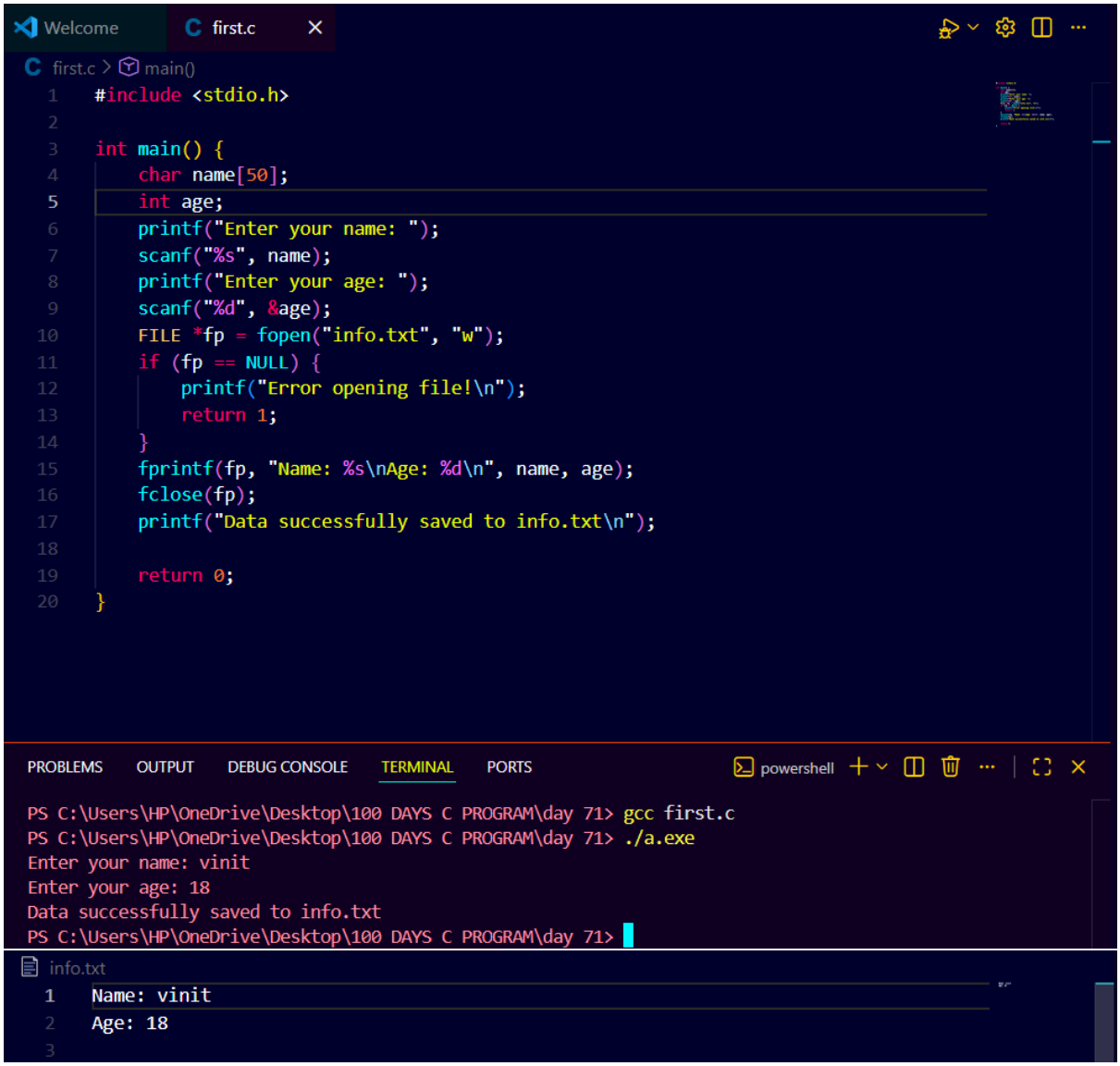
Write a program to take a string input. Change it to sentence case.



DAY-71

📁 Q121 (File Handling)

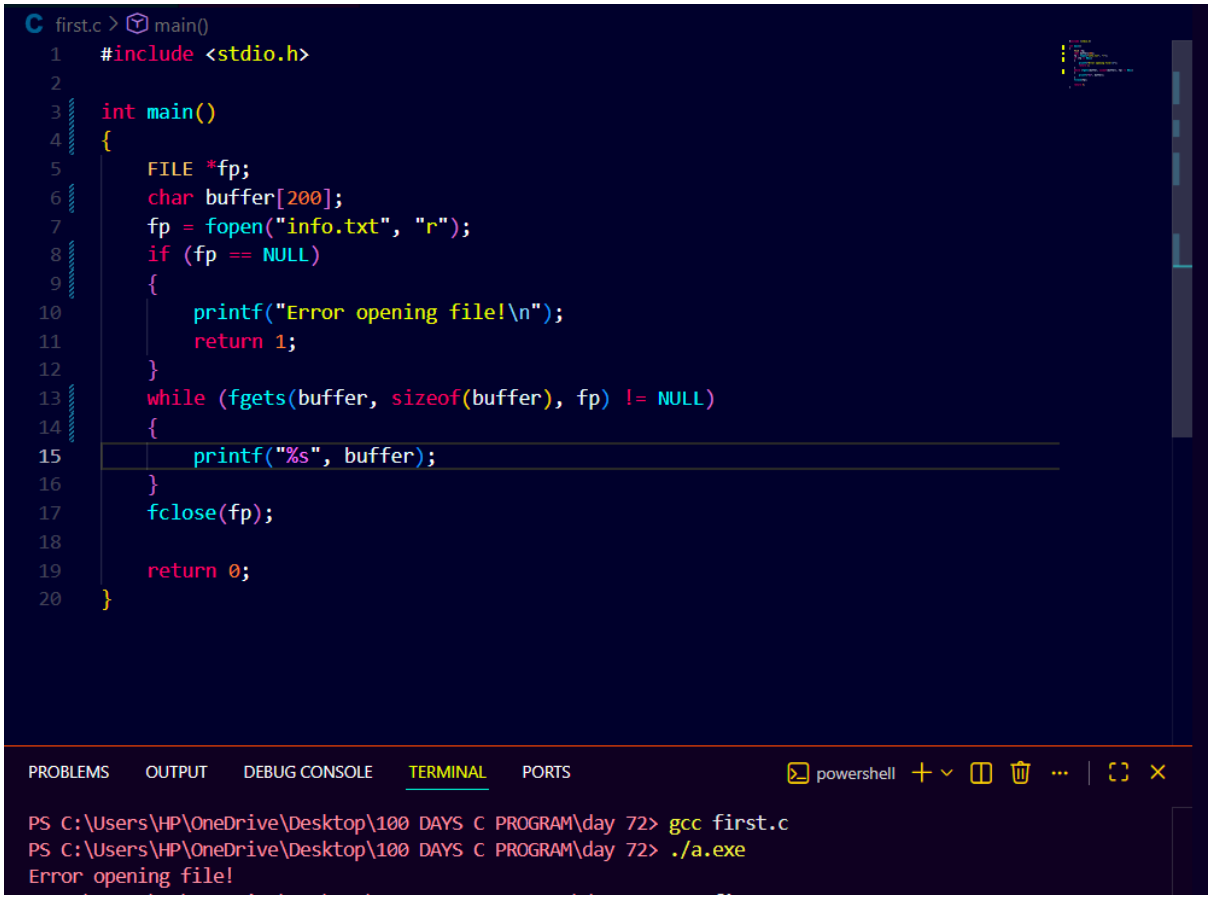
Write a C program that creates a text file named info.txt in write mode. The program should take the user's name and age as input, and write them to the file using fprintf(). After writing, display a message confirming that the data was successfully saved.



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

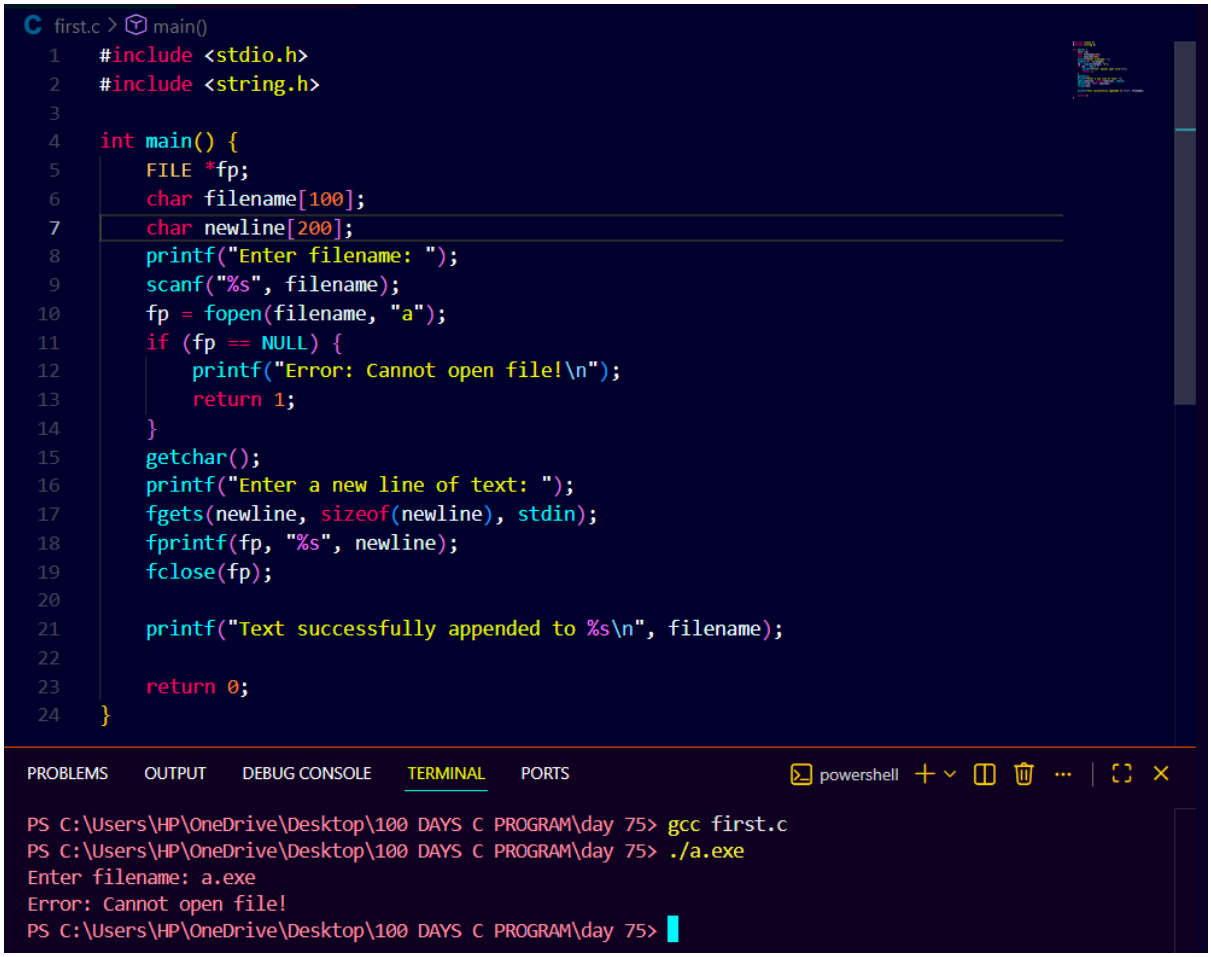
```c
#include <stdio.h>

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)

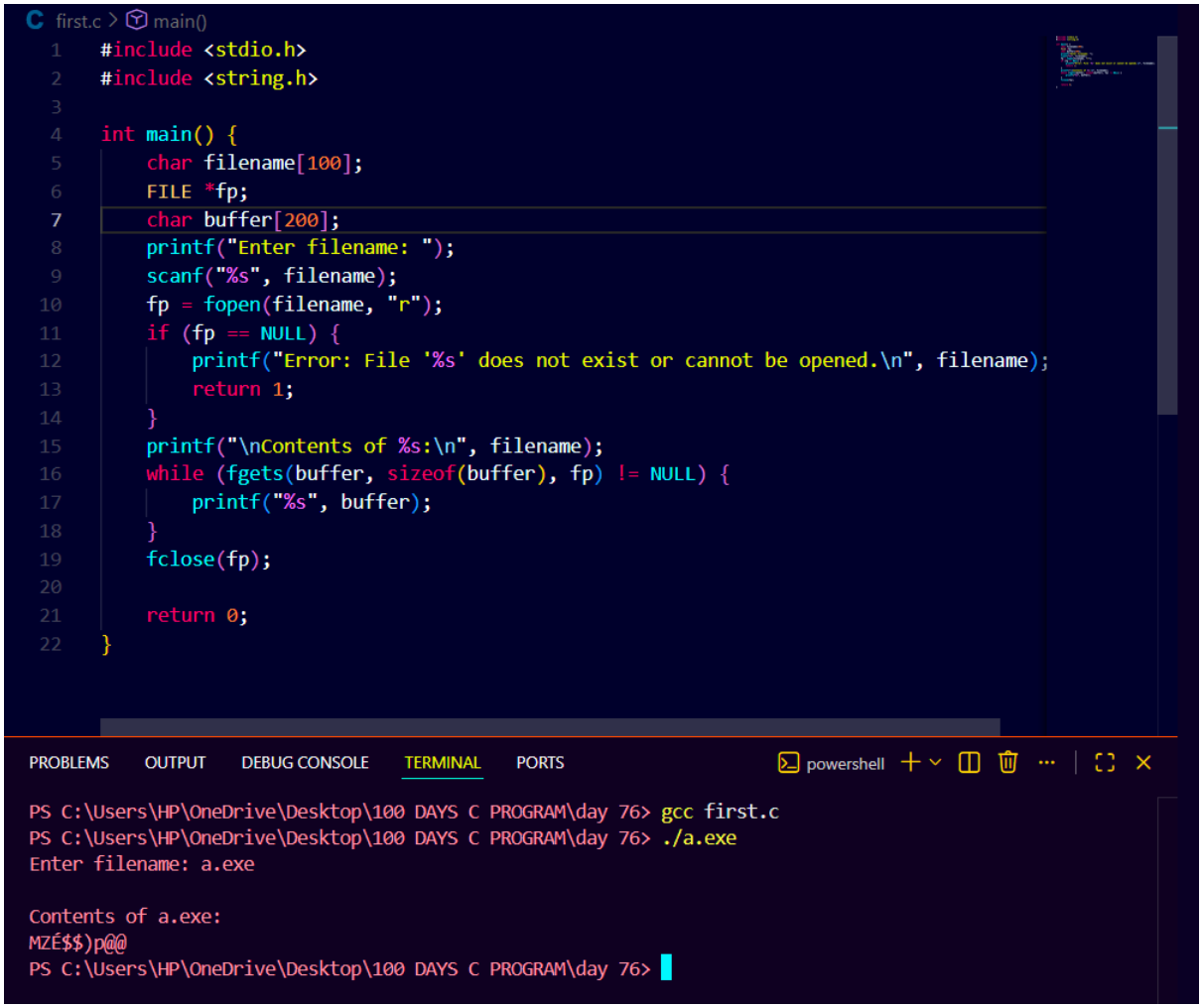
Open an existing file in append mode and allow the user to enter a new line of text. Append the text at the end without overwriting existing content.



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.

day98Q1.c > ...

```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>
```