

## Practice for “Data type, variables, operators, expressions, typecasting”:

1. Given three numbers, swap this in cyclic order. Suppose  $a=5$ ,  $b=6$ ,  $c=7$ . Then the output will be  $a= 7$ ,  $b= 5$ ,  $c= 6$ .
  - a. Solve it using temporary variable
  - b. Solve it without using any temporary variable
2. Given a 4 digits number, reverse the number and print it.
  - a. Solve it following any approach.
  - b. Solve it using a single equation.
3. Given a letter, print the next letter. If the letter is z, then the output will be a. OR
4. i. Given a capital letter X, print the letter N position ahead of the given letter. Consider that there are 26 letters in the alphabet. If the position of  $X+N$  goes beyond 26, rotate your counter to start from the beginning ('A').

Example:

$X='A'$ ,  $N=25$ , Output = 'Z'

$X='A'$ ,  $N=27$ , Output = 'B'

- ii. Solve the above problem, but print the letter that is N position behind from the given letter considering rotation.

## Practice for “Conditional statements”:

1. Given a 4 digit number as input, print the second largest digit.

Input : 5764

Output: 6

- 2.

প্রস্তাবিত করধাপ	প্রস্তাবিত করহার
৩ লক্ষ টাকা পর্যন্ত	শূন্য
পরবর্তী ১ লক্ষ টাকার	৫%
পরবর্তী ৩ লক্ষ টাকার	১০%
পরবর্তী ৪ লক্ষ টাকার	১৫%
পরবর্তী ৫ লক্ষ টাকার	২০%
অবশিষ্ট টাকার	২৫%

Given the table, take a salary as input and print the taxed salary.

Input: 2,50,000

Output: 2,50,000

Input: 3,50,000

Output: 3,47,500

Input: 6,00,000

Output: 5,75,000

Input: 10,00,000

Output: 9,20,000

Input: 12,00,000

Output: 10,85,000

Input: 20,00,000

Output: 17,05,000

3. Take two numbers and a operator (+, -, \*, /) as input and print the output after the operation. Use **Switch Case** to do this.

Input: 2        3        '+'

Output: 5

Input: 5.7      4.4      '-'

Output: 1.3

Input: 30       10       '\*\*'

Output: 300

Input: 40       6        '/'

Output: 6.67

4. Given a year, find out if it is a leap year.
5. Given the marks of a student and the grading policy, print his/her obtained grade.
6. Given a number, check if it is divisible by 2 or 5.
7. Given a number, check if it is divisible by 2 or 5 but not both.
8. Find out the maximum of three numbers.
9. Find out the second lowest of three numbers.
10. Given three lengths, find out if a triangle can be formed. If it can be, then print if the triangle is isosceles or equilateral or scalene.

## Practice Problems on “Loops”:

1. Write a C program to find the sum of the first n natural numbers.
2. Write a C program to calculate the factorial of a given number using a loop.
3. Write a C program to check whether a given number is prime or not.
4. Print the first 1000 prime numbers
5. Write a C program to print the following pattern using nested loops:

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

6. Write a C program to print the following pyramid pattern using nested loops:

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

7. Write a C program to generate the Fibonacci series up to the nth term.
8. Write a C program to check whether a given number is a palindrome or not.

Write the C code of a program that adds all numbers that are multiples of both 7 and 9 up to 600 (including 600) i.e. 63, 126, 189, 252, ....

[Solve it without using conditional statement]

The output of your program should be: 2835

since  $63 + 126 + 189 + 252 + 315 + 378 + 441 + 504 + 567 = 2835$

Write a C program which takes a number and prints how many digits are in that number.

Example: If the user gives 9876, your program should print 4.

Write a C program that takes a number as input from the user and tells if it is a perfect number or not.

Perfect Number: An integer number is said to be a perfect number if its factors, including 1 but not the number itself, sum to the number.

Sample Input 1:

6

Sample Output 2:

6 is a perfect number

Explanation:

6 has 4 divisors: 1, 2, 3, and 6.

If we add all factors except 6 itself,  $1 + 2 + 3 =$

	6.
<p>Print the right triangle.</p> <p>Sample Input: 5</p> <p>Sample Output:</p> <pre> A AB ABC ABCD ABCDE </pre>	<p>Print the Left Triangle.</p> <p>Sample Input : 5</p> <p>Sample Output:</p> <pre> 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 </pre>
<p>Write a C program that takes n as input. The next n lines will take an integer as input. Find out the maximum, minimum and average of the n numbers.</p> <p>You cannot use arrays.</p> <p>If the user enters 5 as an input for quantity and the enters the 5 numbers, 10, 4, -1, -100, and 1.</p> <p>The output of your program should be:</p> <p>Maximum 10</p> <p>Minimum -100</p> <p>Average is -17.2</p>	<p>Print the sum of the given sequence:</p> <p>Sample input: 5</p> <p>Sample Output:</p> $1^2 - 2^2 + 3^2 - 4^2 + 5^2 = 15$
<p>Create a butterfly pattern like the one shown on the right</p>	<p>N = 4</p> <pre> 1      1 12     21 123    321 12344321  N=3  1      1 12     21 123321 </pre>

# Practice Problems on “Array”

**Problem 1:** Floor, Ceil within an array

$A[] = \{96, 21, 58, 34, 46\}$

Number	Floor	Ceil
15	-1	21
21	21	21
30	21	34
96	96	96
98	96	-1

**Problem 2:** Print all possible pairs that have a common factor other than 1

$A[] = \{5, 3, 15, 4, 2\}$

(5, 15), (15,5), (2,4), (4,2) (3,15), (15, 3)

**Problem 3:** Implement Binary Search where Elements can be repeated. Find:

- (1) First occurrence
- (2) Last occurrence
- (3) All occurrences

# Practice Problems on “Strings”, “Functions”

1. Given 3 strings s1, s2, s3, construct a 4th string with appropriate size that will be generated by replacing all occurrences of substring s2 by the substring s3 in the string s1. Also report how many times the replacement occurred. No library functions should be used.

Sample Input:

s1 = "I scream, you scream, we all scream for ice cream!"

s2 = "scream"

s3 = "shout"

Sample Output:

Resulting string: I shout, you shout, we all shout for ice cream!

Number of replacements: 3

2. Take the size of an array and the elements of the array input from the user. Write a function that takes the array as parameter and creates two arrays that contain even numbers and odd numbers. Print the arrays inside the function.

Sample Input:

6

2 1 9 6 3 5

Sample Output:

even: {2,6}

odd: {1,9,3,5}

3. Take the size of an array and the elements of the array input from the user. Write a function that takes the array as parameter, prints the frequency of each unique element in the array and returns the element with the highest frequency.

Sample Input:

8

3 6 4 3 2 4 4 1

Sample Output:

3 : 2 times

6 : 1 times

4 : 3 times

2 : 1 times

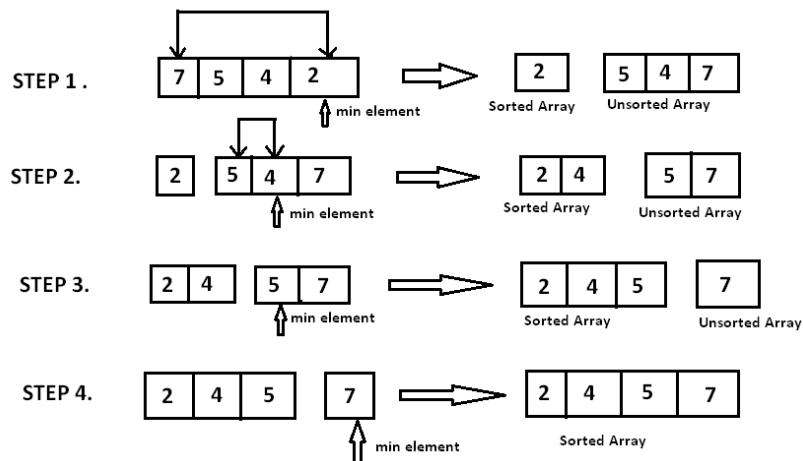
1 : 1 times

4

4. Implement the string functions below:

- int myStrlen(char s[100])
- void myStrrev(char s[100])
- int myStrcmp(char s[100], char t[100])
- void myStrcpy(char to[100], char from[100])
- void myStrcat(char s[100], char t[100], char dest[100])

5. Selection sort



6. Insertion sort





## 7. Bubble sort

# Bubble Sort

i = 0	j	0	1	2	3	4	5	6	7
	0	5	3	1	9	8	2	4	7
	1	3	5	1	9	8	2	4	7
	2	3	1	5	9	8	2	4	7
	3	3	1	5	9	8	2	4	7
	4	3	1	5	8	9	2	4	7
	5	3	1	5	8	2	9	4	7
i = 1	6	3	1	5	8	2	4	9	7
	0	3	1	5	8	2	4	7	9
	1	1	3	5	8	2	4	7	
	2	1	3	5	8	2	4	7	
	3	1	3	5	8	2	4	7	
	4	1	3	5	2	8	4	7	
i = 2	5	1	3	5	2	4	8	7	
	0	1	3	5	2	4	7	8	
	1	1	3	5	2	4	7		
	2	1	3	5	2	4	7		
	3	1	3	2	5	4	7		
i = 3	4	1	3	2	4	5	7		
	0	1	3	2	4	5	7		
	1	1	3	2	4	5			
	2	1	2	3	4	5			
i = 4	3	1	2	3	4	5			
	0	1	2	3	4	5			
	1	1	3	2	4				
i = 5	2	1	2	3	4				
	0	1	2	3	4				
i = 5	1	1	2	3					
	0	1	2	3					
		1	2						

# Practice Problems on “Pointer”

1. Write a function called `void compressString(char *str)` to compress a string using counts of repeated characters. The function should print the compressed string if it is shorter than the original; otherwise, it should print the original string. The output should be printed within the function.

Finally, write a main function to:

- Take a string input from the user.
- Call the `compressString` function.

Note:

- You may use library functions as needed to achieve the task.
- Do not use array syntax.
- Do not use nested loops.

Examples

1. **Input:** aabcccccaaa

**Output:** a2b1c5a3

2. **Input:** abc

**Output:** abc

2. Write a function that takes two integer **pointers** as parameters and swap their values using pointers. Your function should not return anything.

Function Signature: **void** swap(int\*, int\*)

3. Write a function that takes an integer pointer as a parameter and finds the factorial of a given number using pointers. It takes another integer pointer as a parameter and stores the factorial in it.

Function Skeleton:

```
void factorial(int* n, int* result)
{
    //your code goes here.
    /*result has the factorial value
}
```

4. Write a function that takes a source string pointer and a destination string pointer as parameters and copies the source string to the destination string using pointers.

Function Signature: **void** mystrcpy(char\*, char\*)

5. Write a function that takes an **array pointer** as parameter and return the sum of all the elements of the array using **recursion and pointers**.

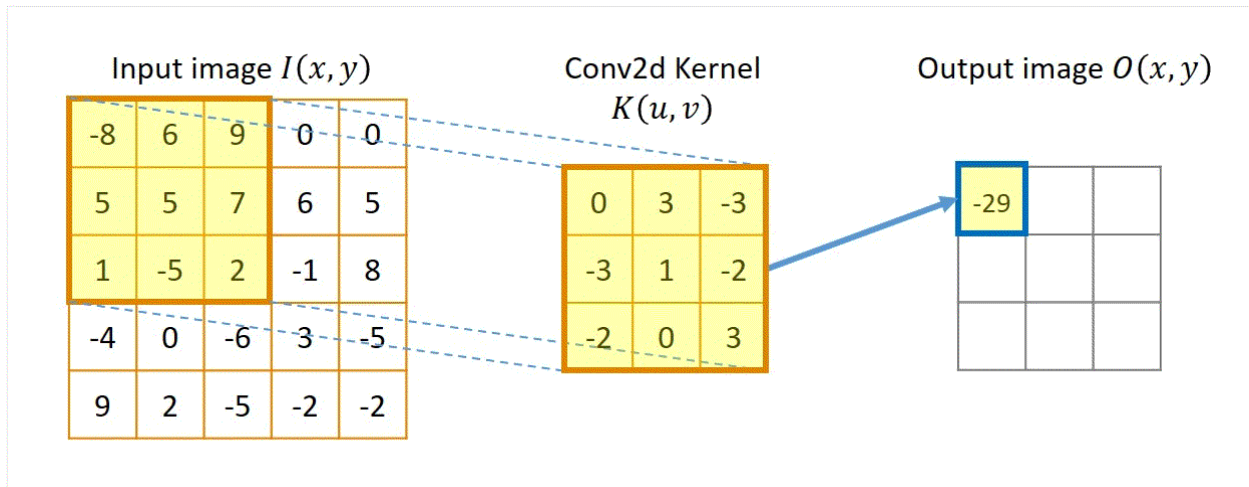
Function Signature: `int sum(int* arr, int size)`

6. Given a 2D Array of integers as input,
- Output the maximum and minimum value of the array.
  - Also given the index of two rows as input, swap the two rows and output the resulting array.
  - Also given the index of two columns as input, swap the two columns and output the resulting array.
7. Given an MxN 2D integer matrix, compute and print its transpose using pointer arithmetic only. Transpose should be stored in a new dynamically allocated matrix.
8. Given an MxN matrix, sort each column individually in ascending order, using pointer arithmetic. You may use any sorting algorithm, but must access elements via pointers only..
9. Given an integer N, create a 3D integer array, somewhat resembling the shape of a 3d rectangular pyramid, meaning that, the first dimension of the 3d array should be 1x1 matrix, second should be 2x2 and so on. This should go on up to N dimension. Fill the array with random numbers. Then take one of the 5 surfaces as input and print the numbers on that surface in appropriate shape
10. Convolution is an important operation in image processing and deep learning. It involves sliding a small matrix called a kernel (or filter) over a larger 2D input array (like an image or a matrix) to produce an output matrix that is the dot product of the overlapping parts in each step. More specifically, at each position, the kernel and the underlying subregion of the input are multiplied element-wise and summed to produce a single value in the output.

For an input matrix I of size MxN and a kernel K of size KxK (assume odd dimensions, typically (3x3), the convolution operation produces an output matrix O of size  $(M-k+1) \times (N-k+1)$

Write a C program that performs 2D convolution of a given matrix (the input array) with a given kernel using pointer arithmetic only. The function should take the input matrix, its dimensions, the kernel, and the kernel dimension as inputs, and output the result of the convolution. You

must not use nested `[ ][ ]` array access. Instead, perform all operations using pointer arithmetic `(*(ptr + i * width + j))` to simulate 2D traversal. You are expected to manually allocate space for the output matrix in main and pass it to the function. You may write helper functions to print matrices, but they should also use pointer arithmetic if accessing matrix elements.



## Practice Problems on “Recursion”

- Write a recursive function **`void printNumbersIn(int n)`** to print numbers from 1 to N in increasing order.
- Write a recursive function **`void printNumbersDe(int n)`** to print numbers from n to 1 in decreasing order.
- Write a recursive function **`int sumOfNaturals(int n)`** to find the sum of the first n natural numbers.
- Write a recursive function **`int factorial(int n)`** to find the factorial of n.
- Write a recursive function **`int pow(int x, int y)`** to calculate the power of a number (e.g.,  $x^y$ ).
- Write a recursive function **`int fibonacci(int n)`** to print the nth Fibonacci number and utilize this function to implement another recursive function **`void printFibonacciSeries(int n)`** to print the Fibonacci series from 0th to nth terms.
- Write a recursive function **`int largest(int arr[], int n)`** to return the largest element in the array.
- Write a recursive function **`int isPalindrome(char str[], int start, int end)`** to check if a given string is a palindrome.

- Write a recursive function ***int gcd (int x, int y)*** to find the greatest common divisor (GCD) of two numbers.
- Write a recursive function ***int binarySearch(int arr[], int low, int high, int key)*** to perform binary search on a sorted array.
- Implement the Tower of Hanoi problem ***void moveDisc(int n, char source, char destination, char auxiliary)*** using recursion.

# Practice Problems on “Multi-Dimensional (2D) Arrays”

- Take an integer as input indicating row/column of a square matrix and then take all the elements of the matrix as input. Write a program in C to find the sum of the upper triangular elements of the matrix.

Constraint:

1. Use dynamic memory allocation
2. **NO USE OF [] operator.**

**Expected Output:**

The given array is :

1 2 3

4 5 6

7 8 9

The elements being summed of the upper triangular matrix are: 2 3 6

The Sum of the upper triangular Matrix Elements are: 11

**Sample I/O:**

Input	Output
3 1 2 3 4 5 6 7 8 9	11
5 2 1 3 4 1 2 3 5 6 1 2 4 5 1 9 0 2 4 1 7 9 4 0 3 2	38

- Take two integers as input indicating row and column of a binary matrix and then take all the elements of the matrix as input. Write a program in C to return only the unique rows from the given binary matrix.

**Sample I/O:**

Input	Output
4 5 0 1 0 0 1 1 0 1 1 0 0 1 0 0 1 1 0 1 0 0	0 1 0 0 1 1 0 1 1 0 1 0 1 0 0
5 6 0 1 1 0 0 1 1 0 1 1 1 0 1 0 1 1 0 1 0 1 1 0 0 1 1 0 1 1 0 1	0 1 1 0 0 1 1 0 1 1 1 0 1 0 1 1 0 1

## . Matrix Compression:

**Complete** the function **compress\_matrix** that takes a 2D array as a parameter and return a new compressed 2D array. In the given array the number of row and column will always be even. **Compressing a matrix means grouping elements in 2x2 blocks and sums the elements within each block. Check the sample input output for further clarification.**

**Hint:** Generally the block consists of the (i,j), (i+1,j), (i,j+1) and (i+1, j+1) elements for 2x2 blocks.

•

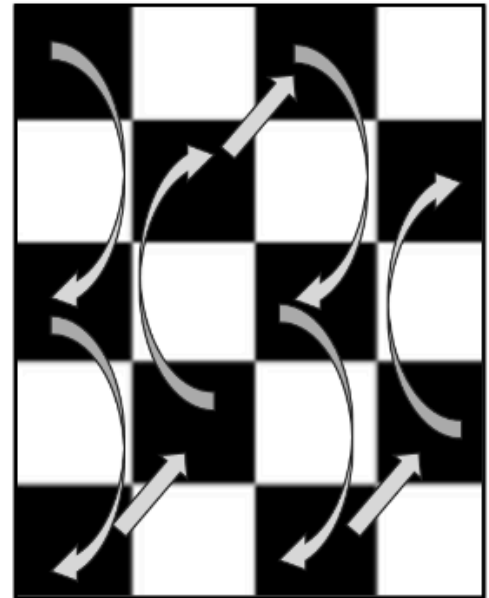
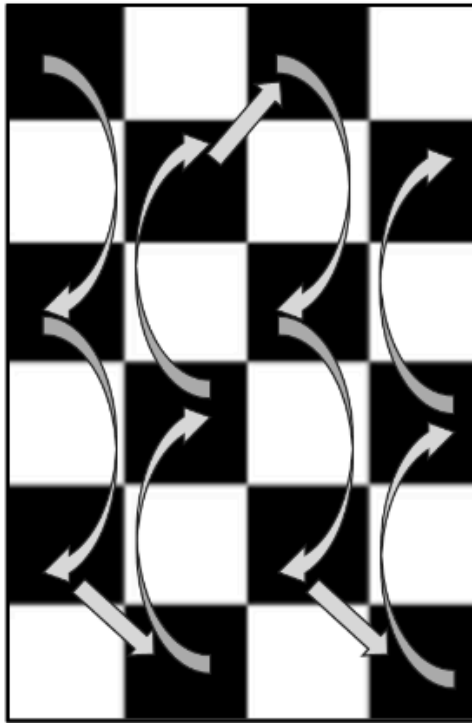


Sample Input array	All Box (No need to create these arrays)	Returned Array	Explanation
[ [1, 2, 3, 4], [5, 6, 7, 8], [1, 3, 5, 2], [-2, 0, 6, -3] ]	[[1, 2],    [[3, 4], [5, 6]]    [[7, 8]]  [[1, 3],    [[5, 2], [-2, 0]]    [[6, -3]]	[[14, 22], [ 2, 10]]	[[1+2+5+6,    3+4+7+8], [1+3+-2+0,    5+2+6+-3]]

Sample Input	All Box	Returned Array	Explanation
1 2 3 4 5 6 7 8 9 5 2 1 3 5 8 1 4 9 12 6 2 -2 3 0	1 2    3 4 5 6 7 8    9 5 2 1  3 5    8 1    4 9 12 6    2 -2    3 0	{{18, 21, 14}, {26 ,9 ,16}}	

- As a child, you often played this game while walking on a tiled floor. You walked avoiding a certain color, for example white tiles (it almost felt like if you stepped on a white tile, you would die!). Now you are in a room of  $m \times n$  dimension. The room has  $m*n$  black and white tiles. You step on the black tiles only. Your movement is like this:

OR



Now suppose you are given a 2D array which resembles the tiled floor. Each tile has a number. Can you write a method that will print your walking sequence on the floor?

**Constraint:** The first tile of the room is always black.

**Hint:** Look out for the number of rows in the matrix [Notice the transition from column 0 to column 1 in the above figures]

Sample Input	Sample Output
<pre> ----- 3   8   4   6   1   -----   7   2   1   9   3   -----   9   0   7   5   8   -----   2   1   3   4   0   -----   1   4   2   8   6   ----- </pre>	<pre> 3 9 1 1 2 4 7 2 4 9 1 8 6 </pre>

Sample Input	Sample Output
<pre> -----   3   8   4   6   1   -----   7   2   1   9   3   -----   9   0   7   5   8   -----   2   1   3   4   0   ----- </pre>	<pre> 3 9 1 2 4 7 4 9 1 8 </pre>

- Take two matrices input from the user. [You know how to take them, right? Already done before]  
Write a program in C to return the multiplication of two matrices.

# Practice Problems on “Bitwise Operator”

- Take an integer as input and determine whether it is “even” or “odd”.
- Take an integer/char as input and determine whether it is “positive” or “negative”.
- Take an integer/char as input and print the binary representation of that integer.
- Swap the content of two integers/characters using bitwise operation
- Take two integers  $r$  and  $m$ . Left rotate the bits of  $m$  by  $r$  bits.
- Take two integers  $r$  and  $m$ . Right rotate the bits of  $m$  by  $r$  bits.
- Take two integers  $n$  and  $var$ . Extract rightmost  $n$  bits of  $var$  and assign it to  $var$ . All other bits of  $var$  except the rightmost  $n$  bits will be set to zero.

# Practice Problems on “Structure”

Concepts: Structure, Array of structures, Nested structures

1. Create a structure named Student with the fields name, marks, and roll. Then, implement a function that calculates the grade of a student based on their marks, using the structure as a parameter. Additionally, create another function to display the details of a student. Write a third function that identifies and displays the student with the highest marks.

In the main function, first prompt the user to enter the number of students. Then, create an array of Student structures and gather the details for each student (name, marks, and roll number). Afterward, calculate and display the grade for each student. Finally, print the student details who got the highest marks using the third function.

2. Enter the marks of  $n$  students in Chemistry, Mathematics and Physics (each out of 100) using a **structure named Marks** having elements/members/fields roll no., name, chem\_marks, math\_marks and phy\_marks and then display the overall percentage of marks for each student.
3. Write a structure to store the roll no., name, age (between 11 to 14) and address of  $n$  students. Store the information of the students.
  - a. Write a function to print the names of all the students having age 14.
  - b. Write another function to print the names of all the students having even roll no.
  - c. Write another function to display the details of the student whose roll no is taken from input (i.e. roll no. entered by the user).
4. Write a structure to store the name, account number and balance of  $n$  customers and store their information.
  - a. Write a function to print the names of all the customers having balances less than \$200.
  - b. Write a function to add \$100 in the balance of all the customers having more than \$1000 in their balance and then print the incremented value of their balance.
5. Write a structure to store the x, y coordinates of a point. Then, write another structure that stores the two endpoints of a diagonal of a rectangle (For simplicity, you can assume that the sides of the rectangle are aligned with x and y axis). Then do the following:

Maintain an array of rectangles each identified by two endpoints of its diagonal.

Using functions-

- a. Calculate the length of the diagonal, area and perimeter of each rectangle
- b. Check if a point is inside or outside a rectangle

- c. Check if a rectangle is completely inside, outside or overlapping with another rectangle
- d. Print the of array of rectangles

## Practice Problems on “Union”

1. Create a union called 'Shape'. The union should contain three structures: circle, triangle and rectangle. Each structure should contain the necessary parameters to calculate the area of its shape (For example, the structure circle should contain a member float variable called radius).

Now, define the following function in your program:

```
double calculateArea(union Shape shape, int type) ; //if type: 1=circle, 2=triangle, 3=rectangle
```

Include necessary user input and code to demonstrate the use of this function.

## Practice Problems on “Bitfield”

1. Create a structure called date, which will contain integer variables for day, month and year (Suppose maximum value of year is 3000). Utilize bitfields within the structure to minimize wasted bits.
2. Create a structure named 'RGBColor' to represent colors using red, green, and blue components. Each component should be represented by an integer ranging from 0 to 255. Implement the structure in such a way that it minimizes memory usage using bitfields, ensuring that no additional bits are wasted.

## Practice Problems on “Enumeration”

1. Define an enumeration type called Season to represent the seasons of the year (Spring, Summer, Autumn, Winter). Now, do the following:
  - a. Write a function to determine the season based on a given month. The function should take the month as input and return the corresponding season.
  - b. Implement a function to display the season for each month of the year. The function should print the name of the month and its season.
  - c. Write a function to determine the number of days in each season. The function should calculate and display the number of days in each season.

Your program should incorporate the use of the enumeration Season wherever possible.

# Practice Problems on “File I/O”

1. Write a program in C to create and store information in a text file named with your student ID. The file content should be the following:

This is the first line  
This is the second line

Use both `fprintf` and `fputs` functions for writing to the file.

2. Now, use both `fscanf` and `fgets` function to print the content of the same file to your console.
3. Now, write a new program to open the same file and append more content to the same file. The file content should be the following:

This is the first line  
This is the second line  
This is the third line  
This is the fourth line

You should only write the last two lines in the file for this problem.

4. Write a program to find the number of lines in a text file.
5. Write a program in C to merge two files and write them to another file.