Practice Problem

Concepts to be covered:

1. - Print & Read
2. - If, if...else if...else if, etc., Switch Statements
3. - 1D-2D Arrays, DMM, Strings, etc.
4. - Loops: for, while, do-while, break, continue
5. - Functions: value, reference, static functions, recursion
6. - Pointers: NULL, Void pointers, Dangling pointers
7. - String handling functions
8. - Structures
9. - FILE
10. - Operators in C: SizeOf, comma, bitwise *, |, ^, etc. ternary operator,
11. - Miscellaneous - Macros, typedef, typecast global and static variables, static functions.
12. - DS: Arrays, LL, Stacks, Queue, Trees, Graphs.
13. More problems

I) Print & Read

Write a program to,

1. Print an integer.
2. Read an integer from the user and print it.
3. Read two integers from the user and print them.
4. Read two integers from the user and print their addition, subtraction and multiplication.
5. swap two numbers using a third variable.
6. swap two numbers without using a third variable.
7. Find the size of int, float, double and char
8. Find the roots of a quadratic equation.
9. Program to assign a character to a variable and print it.
10. Program to read a character from a user, assign it to a variable and print it.
11. Program to read an integer, float, double, long double, a character from the user and print
12. Program to read the string “Hello World” using scanf and print it. (Does it work?)

II) If, if...else, if...else if, etc., Switch Statements

Write a program to,

1. read an integer from the user and check whether the number is positive or negative.

2. read an integer from the user and check whether the number is even or odd.
3. check whether a year is a leap year.
4. find the largest number among the three numbers.
5. Compute quotient and the remainder of two numbers “a” and “b” if b is non-zero.
6. read an integer from the user and print a small number if the number is less than 10, print a large number if the number is greater than 10.
7. read a number from the user and print a small number if the number is less than 100, print a large number if the number is greater than 200, and print a very large number if the number is greater than 1000.
8. read a number from the user and is the number if between 100 and 200 print small number if between 201 and 300 print big number and if the number is between 301 and 400 print large number and if the number is greater than 401 print very large number.
9. read, a number from the user if the number if a number of between 100 and 200 divide it by 3 if the result is less than 50 print small number if it is greater than 50 print “not so big number”. If the input number is between 200 and 300 divide it by 2 if the result is less than 110 print the result else divide the result by 5 and print it. If the input is greater than 300, print a very big number.
10. read, a number if equal to 10 divide it by 2 if the number is equal to 20 divide it by 3 if the number is equal to 30 divide it by 4 if the number is equal to 50 divide it by 5.
11. read, a number if equal to 10 divide it by 2, if the number if equal to 20 divide it by 3, if the number is equal to 30 divide it by 4if the number if equal to 40 divide it by 5. Make use of a switch.
12. read a number if the number is equal to 10 or 20 or 30 divide it by 2, if the number is equal to 40 or 50 divide it by 3, else divide it by 4. Make use of a switch.

III) 1D-2D Arrays, DMM and Strings, etc.

Section A

Write a program to,

1. Read five integers from the user and print them.
2. Read five integers from the user and print them using an integer array.
3. Read five floating-point numbers from the user and print them into a float array.
4. Read five characters from the user and print them using a character array.
5. Read five integers from the user add them and print the addition.
6. Read five integers from the user and add them if the addition is greater than 10 divide the addition by 2 else divide the addition by 3 and print the answer.
7. Create an integer array of size 5 read 10 integers into this array and print the integers. (Make a comment on the output.)
8. Read a character from the user and print it.
9. Read a string from the user and print it.
10. Read a string with spaces from the user and print it (e.g. “Hello World”).

11. Read a 2x2 matrix from the user and print it
12. Read a 3x3 matrix from the user and print it
13. Read a 2x2 matrix from the user and print the addition of each row
14. Read a 3x3 matrix from the user and print the addition of each row and each column
15. Read two 3x3 matrices perform matrix addition on them and print the answer
16. Read two 2x2 matrices perform their multiplication and print the answer
17. Read two 3x3 matrices perform their multiplication and print the answer
18. Dynamically create an integer array of size 10 and read 10 numbers in it.
19. Dynamically create an integer array of size 10, read 10 integers and print the sum of all even integers. Also, release memory before program termination.
20. Dynamically create an integer array of size 10, read 10 integers and print the sum of all integers stored at even indices. Also, release memory before program termination.

Section B

1. WAP to read the dimension M, N of a 2D array
 a. dynamically allocate a 2D array of size MxN
 b. read the values for this array
 c. print this array

2. WAP to create and read a dynamic NxM 2D array from the user and
 a. accept the position of an element in this array and print that element

3. WAP to create and read a dynamic NxM 2D array from user and
 a. accept a row number from the user and print all elements from this row

4. WAP to create and read a dynamic NxM 2D array from user and
 a. accept a column number from the user and print all elements from this column

5. WAP to create and read a dynamic NxM 2D array from user and
 a. divide all elements in the first row by 5 and print the matrix
 b. divide all elements in the first column by 6 and print the matrix
 c. multiply the matrix by 5
 d. multiply the second column by 5
 e. multiply the first and the last column by 2

6. WAP to create and read a dynamic NxM 2D array from user and
 a. multiply the first row by 2 and add it to the second row

7. WAP to create and read a dynamic NxM 2D array from user and

a. multiply the first row by 2 and add it to the second row

8. WAP to create and read a dynamic NxM 2D array from user and
a. replace the first column with zeros
b. replace the last column by 1

9. WAP to create and read a dynamic NxM 2D array from user and
a. swap the first column with the last column
b. swap the first row with the last row

10. WAP to create and read a dynamic NxM matrix called A from user and
a. copy the content of A to a new matrix B. Print B
b. create a third matrix C such that C_ij = B_ij * B_ij. Print C
c. create a fourth matrix D such that D_ij = A_ij+B_ij+C_ij. Print D
d. create a fifth matrix E such that E_ij = 2A_ij + 3D_ij. Print E

IV) Loops: for, while, do-while, break, continue

Write a program to,

1. Count Down from N to 1
2. Print Even numbers from 1 to N
3. Print Odd numbers from 1 to N
4. Calculate the Sum of Even numbers up to N
5. Calculate the Sum of Odd numbers up to N
6. Sum of Digits in a Number
7. Print Squares of Numbers from 1 to N
8. Print the Table of Squares and Cubes
9. Print Multiples of 3 up to N
10. Calculate the Sum of Squares of First N Natural Numbers
11. Find the Sum of Odd Digits in a Number
12. Read five integers from the user into an integer array using for loop
13. Read five integers from the user and perform their addition, and multiplication user for loop
14. Read 10 integers from the user and perform the addition of even numbers
15. Read 10 integers from the user and perform multiplication of odd numbers
16. Read 10 integers from the user and perform addition of all even numbers and multiplication of all odd numbers
17. Read 10 integers and print them along with the size of the array
18. Print “Hello World...Go to Hell” infinite times using for loop
19. Read a number from the user and generate a multiplication table
20. Print characters from A to Z using for loop
21. Calculate the power of a number (i.e. x^y) using for loop
22. Find the largest element of an integer array using for loop
23. Calculate the average of 10 numbers using array and for loop
24. Calculate the standard deviation of 10 numbers using for loop

25. Check if a number is prime or composite
26. Count the number of digits in an integer
27. Reverse an integer
28. Find the factorial of a number
29. Print Fibonacci series
30. Check if a number is Palindrome or not
31. Convert a binary number to a decimal using while loop
32. Find the length of a string
33. Rewrite all programs written with for loop with while and do-while loop.

V) Functions: call by value & reference

Write a program with a function to,

1. Add two integers
2. Multiply three integers
3. Check if a given number is prime or composite
4. Print fibonacci series
5. Reverse an integer
6. Count the number of digits in an integer. The function should return the number of digits
7. Find the GCD and LCM of two integers
8. Concatenate two strings
9. Copy string without using strcpy()
10. Print all prime numbers in a given interval
11. To get factors of a number
12. Read two integers and a function to return the addition and multiplication of these two integers.
13. Read a 3x3 matrix and another function to calculate the trace of a matrix (trace of a matrix is the sum of diagonal entries in the matrix)

VI) Pointers: NULL, Void pointers, Dangling pointers

Write a C program to:

a. Declare an integer variable
b. Declare a pointer variable of type integer
c. Assign the pointer variable address of the integer variable
d. Print the value of the integer variable
e. Print the address of the integer variable using &
f. Print the value of the pointer variable
g. Print the address of the pointer variable
h. Modify and print the value of the integer variable using the pointer

Write a program to (demo. the size of a ptr)

a. Print the size of the integer pointer
b. Print the size of float pointer
c. Print the size of double pointer

d. Print the size of unsinged long integer pointer
e. Print the size of void pointer

Write a program to (demonstrate the effect of post/pre-increment)
a. Declare a pointer variable to an integer.
b. Print the value of the integer and pointer variable
c. Pre-increment the value of the pointer variable using brackets
d. Print the value of the pointer variable and the value of the integer variable
e. Repeat c. and d. without using brackets and observing the difference

WAP to (demo. NULL)
a. Declare a pointer variable of type integer and assign NULL to it.
b. Print the value of the pointer variable
c. Assign the address of an integer variable to the pointer variable
d. Print the value of the pointer variable

WAP to (demo. void ptr)
a. Declare a pointer of type void and assign NULL to it.
b. Print the value of the pointer variable
c. Assign the address of an integer variable to the pointer variable
d. Print the value and address of the void pointer
e. Print the value of the int variable using a void pointer (without typecast)
f. Type case void ptr to int ptr and print the value of the integer variable
g. Declare a double variable and assign its address to void ptr
h. print the value of double var using void ptr. (use typecast)
i. Repeat g. and h. with float and long int variables

WAP to (demo. double ptr)
a. Declare a pointer of type integer and assign NULL to it.
b. Assign the address of an integer variable to the pointer variable
c. Print the value and address of this pointer
d. Declare a double ptr of type integer
e. Assing the address of ptr created in a. to ptr created in d.
f. Print the value of the integer variable using double ptr.
g. Repeat d. to f. with the type for double ptr being void

WAP to
a. Declare a pointer of type integer and assign NULL to it
b. Assign the address of an integer variable to the pointer variable
c. Print the value and address of this pointer
d. Assing the value 10 to the pointer variable
e. Print the value of the pointer variable
f. Print the value stored at the address where the pointer variable is pointing to

WAP to
a. Create a static array of size 3 having values {4, 2, 1}

b. Print the address of this array
c. Create a pointer and assign NULL to it
d. Assign the base address of the array to the pointer and print it
e. Print array values using pointer arithmetic
f. Increment the value of the pointer by 1 and print it
g. Again print array values using pointer arithmetic
h. In f. decrement the value of the pointer by 1 and print it
i. Again print the array values using pointer arithmetic

WAP to
a. Create a dynamic array of size 5 and read values for this array from user
b. Print the address of this array
c. Create a pointer and assign NULL to it
d. Assign the base address of the array to the pointer and print it
e. Print array values using pointer arithmetic
f. Increment the value of the pointer by 1 and print it
g. Again print array values using pointer arithmetic
h. In f. decrement the value of the pointer by 1 and print it.
i. Again print the array values using pointer arithmetic

VII) String handling functions

Write a program to,

1. Copy one string to another string using inbuilt functions
2. Concatenate two strings using inbuilt functions
3. Get the length of a string using inbuilt functions
4. Reverse string using inbuilt functions
5. Replace all occurrences of a particular character with another character
6. Break (tokenize) a string based on space
7. Replace the substring with another substring
8. Compare if two strings are equal
9.
10. .

VIII) Structures

Write a program to,

1. Create a structure complex number with two data members int a, b;
2. Read values into the data members of the structure complex and print them.
3. Create two instances of structure complex, read data into them and print them.
4. Create two instances of structured complex read data into them and add two complex numbers, subtracting two complex numbers.
5. Write a function to accept two complex structures and return the addition of the two complex numbers.

6. Create a structure student with data members roll_number, name and marks for three subjects.
7. Read data into student structure and print it.
8. Read data for 5 students and print it.
9. Read data for 5 students and print average marks for each student along with the student's name and roll number.
10. Create an array of structures for 10 students using a for loop. Read data into this structure and print average marks for each student along with the name and roll number.

IX) FILE

Write a program to,

1. Open a file in reading mode, read a line from the file and print it
2. Open a file in write mode, write a line to the file and close the file
3. Take two file names and compare if the two files are equal or not. Make use of a user-defined function to compare two files.
4. Open a file in reading mode, read data from the file and print all the Vowels from the file.
5. Open a file in append mode and write a line to the file.

More problems

- Given a square matrix of dimension N, generate the following pattern

```
A =  1   X   Y   Z
     X   1   Y   Z
     Y   Y   1   Z
     Z   Z   Z   1
```

- Dodgson's Condensation for determinant computation
- Find all minors (sub-matrices) for a given square matrix