

C Assignments

DAY-1

1. Write a program to find the angle [Degree] between two needles in a clock

Hints: input hours and min

Example Hr=3, Mn =12, Deg=24

2. Write a program:
 - a) To check Given No is prime Or not
 - b) To check Given No is Armstrong Number or not
 - c) To check given Number is perfect Number or not

Hint:1. Armstrong Number Means, Sum of, Cube of the each digit in a number is equivalent to the same Number.

For example:

$$\begin{aligned}153 &= \text{cube}(1) + \text{cube}(5) + \text{cube}(3) \\ &= 1 + 125 + 27 \\ &= 153\end{aligned}$$

Hint 2. Perfect Number Means :

sum of factors of a Number is, equivalent to the same Number

Note: Ignore Number it is as a factor.

Example: i) 6

Factors of 6 are 1,2,3
 $1+2+3=6$

ii) 28

factors are 1,2,4,7,14
 $1+2+4+7+14=28$

Perfect numbers are end with either 6 or 8

3. Write a program to find the difference between two different dates

Example: month1=12 year1=2004 date1=31

month2=3 year2=2005 date2=1

no of days=62

Note: There is no predefined function in C for this

4. Given an array of names, sort the names in alphabetical order.
5. Write a program to concatenate 2 strings without using string functions.
6. Write a program to find **Second biggest** And **Second Smallest** of Given N Number

[note :without using sorting algorithms]

7. Reading a number (from 1 to 999) in words and printing it in digits.
8. Write a program to reverse an integer and a string without using in built functions
9. Write a program insert/delete in a single dimension array.
10. Write a program to find GCD and LCM of given two Number (using function)
11. Write 3 different functions, each function should receive a 2-D array in a different way and print the array.
12. Write a program to find out max and min number in an array given using recursion.
13. Write a recursive procedure to find the sum of all elements in a given array.
14. Write a program to read a line of text and reverse using recursion.
15. Write a program that converts upper case to lower or lower case to upper, depending On the name it is invoked with, as found in argv[1].
16. Write a function that returns a substring from a string.
17. Consider a file which contains fields like a phone directory. Given a phone number, search for the existence of the number in the file and print out the name of the subscriber.
18. Write a program to compare two files, printing the first line where they differ.
19. Simulate the **cat** command of Unix
20. A program to ask for the login and password, check for authentication.
21. Implement binary search.
22. Implement insertion sort.
23. Write a program to evaluate following expression [using array multiplication/addition]
$$2A*5B + C - 6D * 5C$$

Write a program to count digits, white space, others

```
#include <stdio.h>

main() /* count digits, white space, others */
{
    int c, i, nwhite, nother, ndigit[10];

    nwhite = nother = 0;
    for (i = 0; i < 10; i++)
        ndigit[i] = 0;
    while ((c = getchar()) != EOF) {
        switch (c) {
            case '0': case '1': case '2': case '3': case '4':
            case '5': case '6': case '7': case '8': case '9':
                ndigit[c-'0']++;
                break;
            case ' ':
            case '\n':
            case '\t':
                nwhite++;
                break;
            default:
                nother++;
                break;
        }
    }
    printf("digits =");
    for (i = 0; i < 10; i++)
        printf(" %d", ndigit[i]);
    printf(", white space = %d, other = %d\n",
        nwhite, nother);
    return 0;
}
```

Data Structures Assignments

1. Write a program to reverse a single linked list (note: the original list is to be reversed).
2. Write a program to delete k-nodes from the last in a single linked list.
3. Write a program to insert /delete a element in a circular linked list
4. Implement stacks and queues using arrays.
5. Implement a priority queue.
6. Write a program to reverse a queue using stacks.
7. Implementing infix to postfix conversion using stacks.
8. Evaluate a postfix expression using stacks.
9. Write a program that simulates the behavior of CPU scheduler
There should be two queues
 - a. A wait queue where the jobs to be completed are lined up in a queue fashion
 - b. A processioning queue where the job which is being processed is keptAll the jobs will be given an equal time slot by the CPU
Every job has a percentage of job completed and total time to be executed attributes associated with it
Once the cpu time for each job is elapsed the job is put back in wait list and the next job is taken in the queue.