21CS105 PROGRAMMING FOR PROBLEM SOLVING - II

Hours Per Week :

L	T	P	C
3	-	4	5

Total Hours :

L	T	P
45	-	60

source : https:// programskills. wordpress.com

PREREQUISITE COURSES: Programming for Problem Solving - I

COURSE DESCRIPTION AND OBJECTIVES:

This course is aimed to impart knowledge on advanced concepts of C programming language and problem solving through programming. It covers strings, pointers, static and dynamic data structures, and also file manipulations. At the end of this course, students will be able to design, implement, test and debug complex programs using advanced features.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes
1	Design and implementation of string manipulation functions.
2	Creat data structure using dynamic memory and manipulation.
3	Creat text files with different access permissions and manipulations.
4	Apply suitable formatting for I/O data.
5	Develop C programs that are understandable, debuggable, maintainable and more likely to work correctly in the first attempt.

SKILLS:

- ✓ *Analysis of the problem to be solved.*
- ✓ *Select static or dynamic data structures for a given problem and manipulation of data items.*
- ✓ *Application of various file operations effectively in solving real world problems.*
- ✓ *Develop C programs that are understandable, debuggable, maintainable and more likely to work correctly in the first attempt.*

UNIT - I **L-9**

STRINGS: Character array, Reading string from the standard input device, Displaying strings on the standard output device, Importance of terminating a string, Standard string library functions.

UNIT - II **L-9**

POINTERS: Declaration, Initialization, Multiple indirection, Pointer arithmetic, Relationship between arrays and pointers, Scaling up - array of arrays, array of pointers, pointer to a pointer and pointer to an array; Dynamic memory allocation functions.

UNIT - III **L-9**

STRUCTURES: Defining a structure, Declaring structure variable, Operations on structures, Pointers to structure - declaring pointer to a structure, accessing structure members using pointer; Array of structures, Nested structures, Passing structures to functions - passing each member of a structure as a separate argument, passing structure variable by value, passing structure variable by reference/ address; Typedef and structures.

UNIT - IV **L-9**

UNIONS: Defining a union - declaring union variable, operations on union; Pointers to union - declaring pointer to a union, accessing union members using pointer; Array of union, Nested union, Typedef and union, Enumerations, Bit-fields.

UNIT - V **L-9**

FILES: Introduction to files, Streams, I/O using streams – opening a stream, closing stream; Character input, Character output, File position indicator, End of file and errors, Line input and line output, Formatted I/O, Block input and output, File type, Files and command line arguments.

LABORATORY EXPERIMENTS

LIST OF EXPERIMENTS Total hours:60

Experiment 1:

Write a C program to read a sentence from standard input (STDIN) in uppercase letters, lowercase letters, digits, symbols and spaces. Print the given sentence in the reverse order.

Example: If the given input is: Awesome program.

The program should generate the output as: margorp emosewA

Experiment 2:

There is a sequence of words in camelCase as a string of letters, having the following properties:

- It is a concatenation of one or more words consisting of English letters.
- All letters in the first word are lowercase.
- For each of the subsequent words, the first letter is uppercase and rest of the letters are lowercase.

Write a C program to read a string s, and determine the number of words in s.

Example : s=oneTwoThree

There are 3 words in the string: ‘one’, ‘Two’, ‘Three’.

Experiment 3:

Write a C program that reads a string S and remove all duplicates characters from the given string S

NOTE: 1) Order of characters in output string should be same as given in input string.

2) String S contains only lowercase characters [‘a’-’z’].

Example: S = Vignanuniversity

The program should generate the output as: Vignauersty

Experiment 4:

Today Ron is reading the book. Due to some reason, he started hating the word ‘are’ (without quotes). So he decided to replace the substring ‘are’ with ‘R’. Write a C program that reads a line of message ‘s’ and replace the substring ‘are’ with ‘R’.

Example : s= Howareyou

The program should generate the output as: HowRyou

Experiment 5:

Sherlock considers a string to be valid if all characters in the given string appear the same number of times. It is also valid if he can remove just 1 character at 1 index in the string, and the remaining characters will occur the same number of times.

Write a C program that reads a string s and determine whether it is valid or not. If valid, return YES, otherwise return NO.

Example: S=abc

This is a valid string because frequencies are {a:1,b:1,c:1}

S=abcc

This is a valid string because we can remove one c and have 1 of each character in the remaining string.

S=abccc

This string is not valid as we can only remove 1 occurrence of c. That leaves character frequencies of {a:1,b:1,c:2}

Experiment 6:

Read a string containing characters A and B only. Your task is to change it into a string such that there are no matching adjacent characters. To do this, you are allowed to delete zero or more characters in the string.

Write a C program that finds the minimum number of deletions required.

Example : S=AABAAB

Remove A at positions 0 and 3 to make S=ABABA in 2 deletions.

Input Format

The first line contains an integer (the number of queries).

The next q lines each contain a string s to analyze.

Sample Input:

```
5
AAAA
BBBBB
ABABABAB
BABABA
AAABBB
```

Sample Output:

```
3
4
0
0
4
```

Experiment 7:

Write a C program that reads a string ‘s’ and it is said to be complete if it contains all the characters from a to z..

Input Format

First line of the input contains the number of strings N. It is followed by N lines each contains a single string.

Output Format

For each test case print “YES” if the string is complete, else print “NO”

Constraints

1 <= N <= 10

The length of the string is at max 100 and the string contains only the characters a to z.

Experiment 8:

Write a C program that reads two strings and determine whether they share a common substring. or not. A substring may be as small as one character.

Example;

S1=and

S2=art

The common substring in these two strings: a.

Sample Input

```
2
hello
world
hi
world
```

Sample Output

YES
NO

Experiment 9:

Harry was studying a magic book that categorizes the magic spells into 3 categories - Good , Worst and Bad. If any spell contains all the vowels in alphabetical order then that spell is categorized as Good. If it contains the vowels in reverse alphabetical order, then that spell is categorized as Worst. All the other spells that do not fall in any of the categories before are categorized as Bad. Write a C program to categories the given words.

Input Format

The first line of input consists of an integer T denoting the number of spells that need to be classified. Each of the next T lines contains a word 'S'.

Output Format

For each string, display the category to which the spell belongs.

Sample Input:

3
discount
weak
goalkeeper

Sample Output

Good
Worst
Bad

Experiment 10:

Write a C program that completes the function void update(int *a,int *b), which reads two integers as argument, and sets a‘ with th

e sum of them, and b‘ with the absolute difference of them using pointers.

a‘ = a+b
b‘ = |a-b|

Example

The input will contain two integers, a‘ and b‘, separated by a newline.

You have to print the updated value of a and b on two different lines.

Sample Input

4
5

Sample Output

9
1

Experiment 11:

While doing the programming, Kiranmayee fixed a value for a variable by using const keyword. In continuation that, she need to create another variable of same data type which holds different value. But Kiranmayee is thinking about the space management she doesn't want to waste the memory. So, she decides to change the constant variable value. Write a C program to change the constant variable value.

Hint: We cannot change the value of a constant but using pointer we can change it.

Example: Take the initial value of constant variable value is: 10, and display it after that update the constant variable value by20 and display it.

Experiment 12:

Write a C program that creates a jagged array (adjacency list representation of a graph) with no of rows and no of columns in each row as specified by the user

Hint: Use Dynamic memory allocation (malloc() or calloc())

Example:

Input:

Enter no of rows: 3
Enter no of columns Row in 1: 3
Enter no of columns Row in 2: 5
Enter no of columns Row in 3: 2
Enter the elements row wise:
8 6 5
8 4 6 9 7

9 2

Output:

8 6 5

8 4 6 9 7

9 2

Experiment 13:

Write a C program that reads a line of message and check whether it is a ‘Pangram or Not’.

A pangram is a string that contains every letter of the alphabet.

Example

S= “The quick brown fox jumps over the lazy dog”

The string contains all letters in the English alphabet, so return ‘pangram’.

S = We promptly judged antique ivory buckles for the prize

The string does not contain the letter ‘x’, so it returns ‘not pangram’

Experiment 14:

Pooja would like to withdraw ‘X $’ from an ATM. The cash machine will only accept the transaction if X is a multiple of 5, and Pooja’s account balance has enough cash to perform the withdrawal transaction (including bank charges). For each successful withdrawal the bank charges 0.50 $. Write a C program that calculates Pooja’s account balance after the transactions.

Input

Positive integer 0 < X <= 2000. The amount of cash which Pooja wishes to withdraw.

Nonnegative number 0<= Y <= 2000 with two digits of precision - Pooja’s initial account balance.

Output

Output the account balance after the attempted transaction, given as a number with two digits of precision. If there is not enough money in the account to complete the transaction, output the current bank balance.

Example - Successful Transaction

Input:

30 120.00

Output:

89.50

Example - Incorrect Withdrawal Amount (not multiple of 5)

Input:

42 120.00

Output:

120.00

Example - Insufficient Funds

Input:

300 120.00

Output:

120.00

Experiment 15:

Write a C program that creates an array using dynamic memory allocation, and read the values. Compute sum of elements print the sum, and free the memory where the array is stored.

Input Format

The first line contains an integer, N = 5

The next line contains ‘N’ space-separated integers. Example: 10 20 30 40 50

Output Format

Print the sum of the integers in the array: Sum = 150

Experiment 16:

Joe has recently got his new collection of novels. He arranges all books in a shelf. Because of his crazy rules, he only reads book from one of the ends of the shelf. Thus, each book Joe reads is either leftmost or rightmost book from the shelf. Joe cannot read a book if it has more than K chapters, again a crazy rule. When Joe has read a book, he removes it from the shelf. Joe stops when he is unable to read any book from any end of the shelf.

Write a C program that finds, how many books are still present in the shelf?

Input:

The first line of the input contains two integers ‘N’ the total number of books and ‘K’ the maximum number of chapters in a book which Joe can read.

The next N lines denote the number of chapters in the ith book.

Note: Declare a pointer Assign memory by using Dynamic Memory allocation to hold the number of chapters in each book.

Output:

Print the number of books left in the shelf, when Joe stops reading.

Sample Input

8 4
4
3
4
2
5
1
6
4

Sample Output

3

Explanation

Joe can read books in the following order: [4,3,4,2,5,1,6 4]->[3,4,2,5,1,6, 4] -> [3,4,2,5,1,6]->[4,2,5,1,6] ->[2,5,1,6]->[5,1,6]

So the number of books left in the shelf will be 3.

Experiment 17:

Write a C program to create a struct, named Student, representing the student's details as follows: first_name, last_name, Age and standard.

Example

Read student data

john
carmack
15
10

Display the data in the following formate

```
First Name  :        john
Last Name   :        carmack
Age         :        15
Standard    :        10
```

Experiment 18:

Declare a structure POINT. Input the coordinates of point variable and write a C program to determine the quadrant in which it lies. The following table can be used to determine the quadrant.

Quadrant	X	Y
1	Positive	Positive
2	Negative	Positive
3	Negative	Negative
4	Positive	Negative

Example

Input the values for X and Y coordinate: 7 9

The coordinate point (7,9) lies in the First quadrant.

Experiment 19:

Bob and Alice both are friends. Bob asked Alice how to store the information of the books using Structures. Then Alice written a c program to store the information of books using book structure by taking different attributes like book_name, author, book_id, price. Write a C program to read and display the attributes of the books using structures.

Sample Input:

Enter number of books: 1

Enter the book name: c Programming

Enter the author name: balaguruswamy

Enter the book ID: 23413

Enter the book price: 500

Sample Output:

The details of the book are:

```
The book name is: c Programming
The author name is: balaguruswamy
The book ID is: 23413
The book price is: 500.00
```

Experiment 20:

Ramesh wants to do addition on complex numbers. He did it with regular practice but Charan asked him to do with the help of structures by following below Criteria.

Write a C program that defines a structure named ‘Complex’ consisting of two floating point members called “real and imaginary”. Let c1 and c2 are two Complex variables; compute the sum of two variables.

Example:

```
c1= 2    8
c2= 6    4
Sum=        8.000000+12.000000i
```

Experiment 21:

Customer Payment Details is a structure with members as customers_name, address, account_number, payment_status(paid(1)/ not_paid(0)), due_date, and amount. In this example, payment_date is another structure with month, day and year as integer members. So, every customer record can be considered as an array of structures.

Write a C program that displays the amount to be paid by each customer along with their names. If payment_status is 1, display NIL for such customers.

Input Format:

First line of input contains ‘n’ number of customers, followed by 8 lines of input for each customer. Each line represents (customers_name, address, account_number, amount payment_status(paid(1)/ not_paid(0)), and due_date).

Output Format: First line of output is Amount to be paid by each customer as on date: followed by n lines of output. Each line contains name of the customer followed by tab space, and amount to be paid.

Hint: Use nested structure to represent date.

Experiment 22:

Define a union, Date with 3 integer members day, month and year. Write a C program using functions to read data to validate the date (whether it is a leap year or not) and print the date.

Example, If the date is: 20 2 2010 then display ‘not a leap year’.

Experiment 23:

Define a structure called cricket that describes the following information: Player name, team name and batting average Using the structure Cricket, Declare an array player and write a C program that reads information about all the players and print team-wise list containing names of players with their batting average?

Sample Input

```
Enter Number of teams
3
Enter Player Name: Virat Kohli
Enter Team Name: Royal challengers Bangalore
Enter Batting average: 50.2
Enter Player Name: Yuzvendra Singh Chahal
Enter Team Name: Royal challengers Bangalore
Enter Batting average: 8.8
Enter Player Name: MS Dhoni
Enter Team Name: Chennai super Kings
Enter Batting average: 50.6
```

Sample output

```
Team: Royal challengers Bangalore
Player1Name: Virat Kohli
Batting average: 50.2
Player2 Name: Yuzvendra Singh Chahal
Batting average: 8.8
Team: Chennai super Kings
Player Name: MS Dhoni
Batting average: 50.6
```

Experiment 24:

Write a C program to read and display the information (number of students, Roll number, Name, Branch, Phone number) of all the students in a class. Then edit the details of the ith student and redisplay the entire information using structures/unions.

Example:

```
Input
4
4011    Ramu     CSE    8511742256
4012    swathi   CSE    7511742256
4013    tulasi   ECE    8911742256
4014    lakshmi  IT     8711742256
2
4021    sunil    EEE    8711742256
Output:
4011    Ramu     CSE    8511742256
4021    sunil    EEE    8711742256
4013    tulasi   ECE    8911742256
4014    lakshmi  IT     8711742256
```

Experiment 25:

Write a C program to print the bill details of 'N' number of customers with the following data: meter number, customer name, no of units consumed, bill date, last date to deposit and city. The bill is to be calculated according to the following conditions:

No. of units	Charges
For first 100 units	Rs.0.75 per unit
For the next 200 units	Rs.1.80 per unit
For the next 200 units	Rs. 2.75 per unit

```
Sample Input
Enter no. of customers
1
Enter Meter Number AP01213
Enter Customer Name: Karthik
Enter No. of units consumed: 200
Enter Bill date:22/01/2021
Enter Last date: 12/2/2021
Enter City: Guntur
Sample Output
Meter Number AP01213
Customer Name: Karthik
No. of units consumed: 113
Bill date:22/01/2021
Last date: 12/2/2021
City: Guntur
Total Amount: 203.400000
```

Experiment 26:

Write a C program that reads content from an existing text file and write the same in a new file by changing all lowercase alphabetic character to upper case. (Existing file may contain digit and special characters).

Example:

Input: Enter the file name.

Output: New file with updated content

Experiment 27:

Write a C program to count the occurrences of the given string in a file.

Example

Input: Enter the File name to read the string to be counted.

Output: Display the count of occurrences of the string.

Experiment 28:

Write a C Program to transfer the data from one location to another location without changing the order of the content.

Example:

Read the file name from the user. If the source file exists, Transfer the data and display the message as “Data is transferred successfully” otherwise display the message “No such file is existing in the directory.”

Experiment 29:

Write a C program that creates a student file containing {Roll No, Student Name, Address, Stream}, where the data will be inserted and display the list of students who are in CSE (Stream=CSE).

Input: A file name

Output: The attributes such as Roll_No, Student_Name, Stream, Address.

Sample Input

```
201fa4200      Raja    Guntur   CSE
201fa4201      Bala    Tenali   IT
```

Sample Output

```
201fa4200      Raja    Guntur   CSE
```

Experiment 30:

Write a C program that reads numbers and write them into a text-file. Also find odd and even numbers in that file and store it in 2 separate files named odd.txt and even.txt. All the values should be in ascending order.

Input: Enter the values.

Output: Creates a separate file for Even and Odd numbers.

Sample Input:

4 43 2 53 45

Sample Output:

Even.txt: 2 4

Odd.txt: 43 45 53

Experiment 31:

Write a C program to replace the content in the given text file.

Input: Enter the file name, line number to be replaced and the new content

Output: New file with replaced lines.

Example:

Sample Input: Enter the file name: abc.txt

Enter the line no to replace: 3

Enter the content: Files stores data presently.

Sample Output:

Line no 3 is replaced with the given content.

The content of the file abc.txt contains:

test line 1

test line 2

Files stores data presently

test line 4

TEXT BOOKS:

1. Ajay Mittal, “Programming in C - A practical Approach”, 1st edition, Pearson Education Publishers, India, 2010.
2. Reema Thareja, “Introduction to C Programming”, 2nd edition, Oxford University Press India, 2015.

REFERENCE BOOKS:

1. Behrouz A. Forouzan, Richard F.Gilberg, “Programming for Problem Solving”. 1st edition, Cengage Publishers, 2019.
2. Byron S Gottfried, “Programming with C”, 4th edition, Tata McGraw-Hill Publishers, 2018.
3. Herbert Schildt, “C: The Complete Reference”, 4th edition, Tata McGraw-Hill, 2017.