Structures and Unions

1. Write a simple structure called student containing fields (int id, char gender and char name[size]; Create a variable of this structure(ex: std1), fill in the fields of this structure variable and use a pointer to std1 which changes the member values for std1.

2. Define a structure type struct personal that would contain person name, date of joining and salary using this structure to read this information of 5 people and print the same on screen.

3. Define structure data type called time_struct containing three member's integer hour, integer minute and integer second. Develop a program that would assign values to the individual number and display the time in the following format: 16: 40:51

4. Define a structure called cricket that will describe the following information:

Player name
Team name
Batting average

Using cricket, declare an array player with 50 elements and write a C program to read the information about all the 50 players and print team-wise list containing names of players with their batting average.

5. Design a structure student_record to contain name, branch and total marks obtained. Develop a program to read data for 10 students in a class and print them.

File handling

1. Write a program in C to write multiple lines in a text file.
   Test Data :
   Input the number of lines to be written : 4
   The lines are
   test line 1
   test line 2
   test line 3
   Expected Output :
   The content of the file test.txt is :
   test line 1
   test line 2
   test line 3

2. Write a program in C to read the file and store the lines into an array.
   Test Data :
   Input the file name to be opened : test.txt
   Expected Output :
   The content of the file test.txt are :
   test line 1
   test line 2
   test line 3
   test line 4

3. Write a program in C to count a number of words and characters in a file.
   Test Data :
   Input the file name to be opened : test.txt
   Expected Output :
   The content of the file test.txt are :
   test line 1
   test line 2
   test line 3
   test line 4
   The number of words in the file test.txt are : 12
   The number of characters in the file test.txt are : 36

4. Write a program in C to copy a file in another name.
   Assume that the content of the file test.txt is :
   test line 1
   test line 2
   test line 3

   Test Data :
   Input the source file name : test.txt
   Input the new file name : test1.txt
   Expected Output :

   The file test.txt copied successfully in the file test1.txt.

   If you read the new file you will see the content of the file :

   test line 1
   test line 2
   test line 3

5. Write a program in C to replace a specific line with another text in a file.

Assume that the content of the file test.txt is :

test line 1
test line 2
test line 3

Test Data :

Input the file name to be opened : test.txt
Input the content of the new line : Yes, I am the new text instead of test line 2
Input the line no you want to replace : 2

Expected Output :

test line 1
Yes, I am the new text instead of test line 2
test line 1

Preprocessor directives

1. Define a variable inside a header file, say “custom.h”, use this header file in your program and then print the value of the variable defined in custom.h.
2. Using #define, declare a value, say PI = 3.14, and use it to calculate area of a circle, given the radius.
3. Write a for loop to print numbers from 1 to 20. The for loops has to be written as a multiline macro

   Ex:
   forlo(1, 20); => prints numbers from 1 to 20;
   forlo here is a multiline macro

4. Give example usage of conditional compilation using #ifdef, #ifndef and #undef.
5. Concat two numbers to create a single number using ## operator:
   Ex: 11882, 619 => 61911882

Storage Class Specifier

1. Using Static keyword, keep a count of the number of times a function was called.
2. Using extern keywords, declare variable in another file(second.c) , and in another file(first.c), utlize that variable.
   Ex:

```
first.c
        extern int myvar;
        printf(“my var is”, myvar)
second.c
        extern int myvar = 106;
```

Compile using: gcc -o bin first.c second.c

3. Can two functions have variables with the same name? Give example of they both have different scopes.