1A

|  |
| --- |
| 1. ***Write a program that extracts part of the given string from the specified position. For example, if the string is “Working with strings is fun,” then if from position 4, 4 characters are to be extracted then the program should return string as king. If the number of characters to e extracted is 0, then the program should extract entire string from the specified position.*** 2. ***Write a program that converts a string such as “124” into an integer like 124.*** 3. ***Write a program that generates and prints the Fibonacci words of order 0 through 5. F(0)=a, f(1)=b, f(2)=ba, f(3)=bab, f(4)=babba.*** 4. ***Write a program that receives a 10-digit integer, computes the checksum, and reports whether the ISBN number is correct or not.*** |

2A

|  |
| --- |
| **1.** Write a program in C to separate the individual characters from a string.  Test Data : Input the string : w3resource.com  *Expected Output* :  The characters of the string are :  w 3 r e s o u r c e . c o m  **2.** Write a program in C to print individual characters of string in reverse order.  Test Data : Input the string : w3resource.com  *Expected Output* :  The characters of the string in reverse are :  m o c . e c r u o s e r 3 w |

2B

|  |
| --- |
| 1. Write a program in C to count the total number of words in a string.   Test Data : Input the string : This is w3resource.com  *Expected Output* :  Total number of words in the string is : 3  **2.** Write a program in C to compare two strings without using string library functions.  Test Data : Check the length of two strings: -------------------------------- Input the 1st string : aabbcc Input the 2nd string : abcdef String1: aabbcc String2: abcdef *Expected Output* : Strings are not equal.  Check the length of two strings: -------------------------------- Input the 1st string : aabbcc Input the 2nd string : aabbcc String1: aabbcc String2: aabbcc *Expected Output* : Strings are equal. |

1B

|  |
| --- |
| 1. ***WAP that receives a Credit Card number and checks using the above rule whether the Credit Card number is valid.*** 2. Write a program in C to input a string and print it.   Test Data : Input the string : Welcome, w3resource  *Expected Output* :  The string you entered is : Welcome, w3resource  **3.** Write a program in C to find the length of a string without using library function.  Test Data : Input the string : w3resource.com  *Expected Output* :  Length of the string is : 15 |

3A

|  |
| --- |
| **1.** Write a program in C to count total number of alphabets, digits and special characters in a string.  Test Data : Input the string : Welcome to w3resource.com  *Expected Output* :  Number of Alphabets in the string is : 21  Number of Digits in the string is : 1  Number of Special characters in the string is : 4  **2.** Write a program in C to copy one string to another string.  Test Data : Input the string : This is a string to be copied.  *Expected Output* :  The First string is : This is a string to be copied.  The Second string is : This is a string to be copied.  Number of characters copied : 31 |

4A

|  |
| --- |
| **1.** Write a C program to sort a string array in ascending order.  Test Data : Input the string : w3resource  *Expected Output* :  After sorting the string appears like :  3ceeorrsuw  **2.** Write a program in C to read a string through keyboard and sort it using bubble sort.  Test Data : Input number of strings :3 Input string 3 : zero one two  *Expected Output* :  The strings appears after sorting :  one  two  zero |

4B

|  |
| --- |
| **1.** Write a program in C to extract a substring from a given string.  Test Data : Input the string : this is test string Input the position to start extraction :9 Input the length of substring :4  *Expected Output* :  The substring retrieve from the string is : " test "  **2.** Write a C program to check whether a given substring is present in the given string.  Test Data : Input the string : This is a test string. Input the substring to be search : search  *Expected Output* :  The substring is not exists in the string. |

3B

|  |
| --- |
| **1.** Write a program in C to count total number of vowel or consonant in a string.  Test Data : Input the string : Welcome to w3resource.com  *Expected Output* :  The total number of vowel in the string is : 9  The total number of consonant in the string is : 12  **2.** Write a program in C to find maximum occurring character in a string.  Test Data : Input the string : Welcome to w3resource.com.  *Expected Output* :  The Highest frequency of character 'e'  appears number of times : 4 |

5A

|  |
| --- |
| **1.** Write a program in C to read a sentence and replace lowercase characters by uppercase and vice-versa.  Test Data : Input the string : This Is A Test String.  *Expected Output* :  The given sentence is : This Is A Test String.  After Case changed the string is: tHIS iS a tEST sTRING.  **2.** Write a program in C to find the number of times a given word 'the' appears in the given string.  Test Data : Input the string : The string where the word the present more than once.  *Expected Output* :  The frequency of the word 'the' is : 3 |

6A

|  |
| --- |
| **1.** Write a program in C to Concatenate Two Strings Manually.  Test Data : Input the first string : this is string one Input the second string : this is string two  *Expected Output* :  After concatenation the string is :  this is string one this is string two  **2.** Write a program in C to find the largest and smallest word in a string.  Test Data : Input the string : It is a string with smallest and largest word.  *Expected Output* :  The largest word is 'smallest'  and the smallest word is 'a'  in the string : 'It is a string with smallest and largest word.'. |

6B

|  |
| --- |
| **1.** Write a program in C to convert a string to uppercase.  Test Data : Input a string in lowercase : the quick brown fox jumps over the lazy dog  *Expected Output* :  Here is the above string in UPPERCASE :  THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.    **2.** Write a program in C to convert a string to lowercase.  Test Data : Input a string in UPPERCASE : THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.  *Expected Output* :  Here is the above string in lowercase :  the quick brown fox jumps over the lazy dog. |

5B

|  |
| --- |
| **1.** Write a program in C to remove characters in String Except Alphabets.  Test Data : Input the string : w3resource.com  *Expected Output* :  After removing the Output String : wresourcecom  **2.** Write a program in C to Find the Frequency of Characters.  Test Data : Input the string : This is a test string Input the character to find frequency: i  *Expected Output* :  The frequency of 'i' is : 3 |

7A

|  |
| --- |
| **1.** Write a program in C to check whether a character is Hexadecimal Digit or not.  Test Data : Input a character : 7  *Expected Output* :  The entered character is a hexadecimal digit.  **2.** Write a program in C to check whether a letter is uppercase or not.  Test Data : Input a character : p  *Expected Output* :  The entered letter is not an UPPERCASE letter. |

8A

|  |
| --- |
| **1.** Write a program in C to print only the string before new line character.  Note: isprint() will only print line one, because the newline character is not printable.  *Expected Output* :  The quick brown fox  **2.** Write a program in C to check whether a letter is lowercase or not.  Test Data : Input a character : w  *Expected Output* :  The entered letter is a lowercase letter. |

8B

|  |
| --- |
| **1.** Write a program in C to read a file and remove the spaces between two words of its content.  *Expected Output* :  The content of the file is :  The quick brown fox jumps over the lazy dog  After removing the spaces the content is :  Thequickbrownfoxjumpsoverthelazydog  **2.** Write a program in C to check whether a character is digit or not.  Test Data : Input a character : 8  *Expected Output* :  The entered character is a digit. |

7B

|  |
| --- |
| **1.** Write a program in C to replace the spaces of a string with a specific character.  Test Data : Input a string : Be glad to see the back of Input replace character : \*  *Expected Output* :  After replacing the space with \* the new string is :  Be\*glad\*to\*see\*the\*back\*of\*  **2.** Write a program in C to count the number of punctuation characters exists in a string.  Test Data : Input a string : The quick brown fox, jumps over the, lazy dog.  *Expected Output* :  The punctuation characters exists in the string is : 3 |

9A

|  |
| --- |
| **1.** Write a program in C to split string by space into words.  Test Data : Input a string : this is a test string  *Expected Output* :  Strings or words after split by space are :  this  is  a  test  string .  **2.** Write a C programming to find the repeated character in a given string.  Test Data : Input a string: w3resource  *Expected Output*:  Input a string: The first repetitive character in w3resource is: r |

10A

|  |
| --- |
|  |

10B

|  |
| --- |
|  |

9B

|  |
| --- |
| **1.** Write a C programming to count of each character in a given string.  Test Data : Input a string: w3resource  *Expected Output*:  Enter a str1ing: The count of each character in the string w3resource is  w 1  3 1  r 2  e 2  s 1  o 1  u 1  c 1  **2.** Write a C programming to convert vowels into upper case character in a given string.  Test Data : Input a string : w3resource  *Expected Output*:  Input a sentence: The original string:  w3resource  After converting vowels into upper case the sentence becomes: w3rEsOUrcE |

11A

|  |
| --- |
|  |

12A

|  |
| --- |
|  |

12B

|  |
| --- |
|  |

11B

|  |
| --- |
|  |

13A

|  |
| --- |
|  |

14A

|  |
| --- |
|  |

14B

|  |
| --- |
|  |

13B

|  |
| --- |
|  |

15A

|  |
| --- |
|  |

16A

|  |
| --- |
|  |

16B

|  |
| --- |
|  |

15B

|  |
| --- |
|  |

17A

|  |
| --- |
|  |

18A

|  |
| --- |
|  |

18B

|  |
| --- |
|  |

17B

|  |
| --- |
|  |

19A

|  |
| --- |
|  |

20A

|  |
| --- |
|  |

20B

|  |
| --- |
|  |

19B

|  |
| --- |
|  |

21A

|  |
| --- |
|  |

22A

|  |
| --- |
|  |

22B

|  |
| --- |
|  |

21B

|  |
| --- |
|  |

23A

|  |
| --- |
|  |

24A

|  |
| --- |
|  |

24B

|  |
| --- |
|  |

23B

|  |
| --- |
|  |

25A

|  |
| --- |
|  |

26A

|  |
| --- |
|  |

26B

|  |
| --- |
|  |

25B

|  |
| --- |
|  |

27A

|  |
| --- |
|  |

28A

|  |
| --- |
|  |

28B

|  |
| --- |
|  |

27B

|  |
| --- |
|  |

29A

|  |
| --- |
|  |

30A

|  |
| --- |
|  |

30B

|  |
| --- |
|  |

29B

|  |
| --- |
|  |