Practice Set: Functions (19 Problems)-v202

<table>
<tr><th>SL</th><th>Problem statement</th><th>Difficulty levels</th></tr>
<tr><td>1.</td><td>Function to print a message.

<table><tr><th>Sample input</th><th>Sample output</th></tr><tr><td></td><td>This is a function</td></tr></table></td><td>*</td></tr>
<tr><td>2.</td><td>Function to print an input character value.

<table><tr><th>Sample argument</th><th>Sample output</th></tr><tr><td>‘3’</td><td>Value received from main: 3</td></tr><tr><td>‘A’</td><td>Value received from main: A</td></tr></table></td><td>*</td></tr>
<tr><td>3.</td><td>Write a function in C that takes a float x as the parameter and returns the value of $1/x^2$.

<table><tr><th>Sample argument</th><th>Sample return value</th></tr><tr><td>1.0</td><td>1.0</td></tr><tr><td>2.0</td><td>0.25</td></tr><tr><td>-3.5</td><td>0.081633</td></tr></table></td><td>*</td></tr>
<tr><td>4.</td><td>Write a function in C that takes two floats x and y as the parameters and returns the absolute value of their difference. You cannot use any library function.

<table><tr><th>Sample Arguments</th><th>Sample Return Value</th></tr><tr><td>3.3 7.4</td><td>4.1</td></tr><tr><td>4.0 1.5</td><td>2.5</td></tr></table></td><td>*</td></tr>
<tr><td>5.</td><td>Write the function greet that prints the message “Hello” n times.

<table><tr><th>Sample Argument</th><th>Sample Output</th></tr><tr><td>3</td><td>Hello
Hello
Hello</td></tr><tr><td>1</td><td>Hello</td></tr></table></td><td>*</td></tr>
</table>

6. Write the function **greet** that prints the desired output.

Sample Arguments	Sample Output
2 4	Hello 2 Hello 3 Hello 4
3 7	Hello 3 Hello 4 Hello 5 Hello 6 Hello 7

7. Write a function in C that takes the base and the height of a triangle as **two float parameters** and returns the **area of the triangle**. [Formula: Area = 0.5*base*height] *

Sample Arguments	Sample Return Value
2.0 5.0	5.0
4.25 4.0	8.5

8. Function to calculate the sum of **three** numbers. *

Sample arguments	Sample return value	Sample output
80 33 27	140	Sum in Main: 140
100 -100 0	0	Sum in Main: 0

9. Function to print and return the sum of **n** numbers coming from an array. *

Sample arguments	Sample return value	Sample output in main
80 33 27 3	140	Sum in Main: 140
100 -100 2	0	Sum in Main: 0

10. Function to swap two numbers. *
(**Restriction:** Pass by value)

Sample input	Sample output
10 20	Value in func: 20 10 Value in main: 10 20

11. Function to determine only even numbers in an array of input integers. *

Sample input	Sample output
24 77 117 -512 1024	24 -512 1024
45 33 0 256	0 256

12. Function that finds and returns the minimum value in an array. **

Sample arguments	Return value	Output in main
157 -28 -37 26 10 5	-37	Minimum Value: -37
12 45 1 10 5 3 22 7	1	Minimum Value: 1

13. Function that multiplies the array elements by 2. *

Sample arguments	Sample output in main
157 -28 -37 26 10 5	314 -56 -74 52 20
12 45 1 10 5 3 22 7	24 90 2 20 10 6 44

14. Function to sort and return an input array in ascending order. **

Sample arguments	Sample output in main
10 22 -5 117 0 5	-5 0 10 22 117

15. Function **"IsPrime()"** to determine whether a number is prime or not. **

Sample argument	*Sample return value*	*Sample output in main*
1	0	Not prime
2	1	Prime
11	1	Prime
39	0	Not prime
101	1	Prime

16. Program that takes two positive integers as inputs and uses two functions to find their GCD (greatest common divisor) and LCM (least common multiple). Both functions take parameters and returns desired values. **

Sample input	*Sample output*
5 7	GCD: 1 LCM: 35
12 12	GCD: 12 LCM: 12
12 32	GCD: 4 LCM: 96

17. Function **"GeneratePrime()"** to compute the prime numbers less than **N,** where **N** is an input integer. **GeneratePrime()** uses **IsPrime()** to check whether a number is prime or not. ***

Sample input	**Sample output**
5	Prime less than 5: 2, 3
10	Prime less than 10: 2, 3, 5, 7
40	Prime less than 17: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37

18. Function **find_substr( )** that takes two string arrays (**a, b**) as parameters, returns 1 if string **b** is found anywhere in string **a**, or returns –1 if no match is found. **

(Assuming, strlen(a)>strlen(b))

Sample input (a, b)	**Sample output**
madam adam	1
telescope less	0
101010 101	1

19. You will be given exactly five positive integers. For each of the numbers, say N. you will print the smallest prime that is greater than N. **
Additional Constraints:
You have to write and use the following functions.

isPrime

parameter: an integer.

returns: 1 if the received integer is prime, 0 otherwise

nextPrime

parameter: an integer.

returns: the smallest prime that is greater than the received integer.

calls isPrime

main

reads 5 integers with scanf

calls nextPrime

Sample input	**Sample output**
1 2 3 8 21	2 3 5 11 23