

CS118

Study Session

Design and implement algorithmic solutions to problems requiring elementary processing concepts of arithmetic, basic data types and operations.

1. Without using any example code, write a Python program that collects two floats, A and B, from the user and generates and prints 5 random floats that lie between A and B, inclusive.

Design and implement algorithmic solutions to problems requiring the basic control structures of sequences, selection, and repetition.

2. Without using any example code, modify the program from #1 so that it collects a third value – an integer, N - from the user. And then have the program generate integers which lie between A and B. Print only N values - but they must be divisible by 5.

Design and implement algorithmic solutions to problems requiring array structures

Design and implement algorithmic solutions to problems requiring the application of linear search and sorting.

3. Without using any example code, modify the program from #2 so that the N values are not printed, but instead stored in a list. Sort the list manually (don't use any functions or methods) and then print the sorted list, one item per line of output.

Design and implement solutions of intermediate complexity using functions.

4. Without using any example code, modify the program from #3 so that the N values are collected and stored in the list in a function `GetList()`. The function should take two arguments: the number of values to collect, and an empty list to be filled. The return value from the function is how many values were generated in order to obtain the N required values.

Design and implement solutions of intermediate complexity requiring the use of non-numerical data such as booleans, characters, and strings.

5. Without using any example code, write a function that will find all palindrome substrings within a string, print those substrings (one per line), and return the number of palindrome substrings found.