

INDIAN STATISTICAL INSTITUTE

MTech(CS) I year 2020-2021

Subject: Computing Laboratory

Assignment 1

Total: $3 \times 20 = 60$ marks Deadline: 07:00am, 27 January, 2021

SUBMISSION INSTRUCTIONS

1. Please make sure that your programs adhere strictly to the specified input and output format.
2. Please adhere to the file naming conventions discussed in class.

IMPORTANT: Insert a single alpha-numeric string of your choice, 6-8 characters long, in the name given above as shown in the examples below. Think of this string as something like a security password, except that you are not required to remember the string.

Examples: `cs20XX-assign1-abcdef-prog1.py`, `mnoqr-cs20XX-assign1-prog2.py`,
`cs20XX-assign1-prog2-uvwxyz.py`.

3. Please upload your programs to `~dfs/2020/assign1/cs20XX/` on the ISI server (192.168.64.35), as advised in class.

NOTE: Unless otherwise specified, all programs should take the required input from stdin, and print the desired output to stdout.

Q1. Inversion Pair Count. Two elements $A[i]$ and $A[j]$ of a list A are said to form an *inversion pair* if $A[i] > A[j]$ but $i < j$. Write a program to count the number of inversion pairs in a list A containing distinct integers.

Input format: Your program should take the list of distinct integers as command-line arguments.

Output format: Your program should print the number of inversion pairs on the screen.

Sample input 1: 8 4 2 1

Sample output 1: 6

Explanation: The 6 inversion pairs are (8, 4), (4, 2), (8, 2), (8, 1), (4, 1), (2, 1).

Sample input 2: 3 1 2

Sample output 2: 6

Explanation: The 2 inversion pairs are (3, 1) and (3, 2).

Q2. Fast Multiplication. Given two binary strings representing two integers, find the product of the two integers using Karatsuba's algorithm.

Input format: The input will consist of a single line containing the two binary strings, separated by a space. You may assume that the lengths of the two strings are the same.

Output format: Your program should print the product on the screen **in decimal**.

Sample input 1: 1100 1010

Sample output 1: 120

Q3. Maximum Sub-sequence Sum. Given a sequence of N integers stored in a list, determine the maximum sum that can be formed by adding any contiguous sub-sequence of the list. The problem is easy if all the numbers are positive: all numbers in the sequence should be added. The difficulty arises when there are negative integers: should you take a negative integer in the hope that nearby positive integers will compensate for it? Note that the maximum sub-sequence sum will always be non-negative, because when **no** elements are selected (i.e., an *empty* sub-sequence is selected), the sum obtained is 0.

Input format: The input will consist of a single line containing the integers.

Output format: Your program should print the starting index and the ending index of the contiguous sub-sequence, along with the sum. All three integers should be printed on one line, separated by spaces.

Sample input 1: 12 -34 40 6 -10 56 12 -1 -15 10 4

Sample output 1: 2 6 104

Explanation: The maximum possible sum is 104, achieved by taking elements 2 through 6.