Lab 3: Divide and conquer algorithms

Q1. The multiplication of large integers problem.

Problem statement: Apply the Divide-and-conquer approach to implement the multiplication of two large integers, u and v.

Input: The user inputs two strings, each of which denotes a large integer, from the keyboard.

Output: Print to the console the string that represents the product of the given two large integers.

Example of input and output:

Input from user keyboard	Output to console
123 456	56088

Note that you only need to implement the upgraded version shown in the lecture.

Q2. The multiplication of two positive integers of n bits.

Problem statement: Apply the Divide-and-conquer approach to implement the multiplication of two positive integers, x and y, of n bits. Assuming that n is the power of 2.

Input: The user inputs two positive integers, x and y.

Output: Print to the console the product of the given two large integers.

Example of input and output:

Input from user keyboard	Output to console
123 456	56088

Q3. Find the contiguous subarray of largest sum.

Problem statement: Given an array of n integers, $\{a_1, a_2, .., a_n\}$. Apply the Divide-and-conquer approach to find the subsequence $\{a_i, a_{i+1}, .., a_{j-1}, a_j\}$ such that $\sum_{k=i}^{j} a_k$ is largest, where $1 \leq i \leq k \leq j \leq n$. If all integers in the sequence are negative, the subsequence is empty and the result is 0.

Input: Read the input data from a text file whose format is as follows

- 1st line: a positive integer n to indicate the size of the input array
- 2nd line: n integers, two consecutive numbers are separated by a single space " "

Output: Print the output data to the console as follows

- The subsequence with the largest sum of elements
- The sum of the above subsequence

Example of input and output:

Input text file	Output to console
8 -2 -3 4 -1 -2 1 5 -3	4 -1 -2 1 5 7

Q4. The Strassen's matrix multiplication problem.

Problem statement: Apply the Divide-and-conquer approach to implements the Strassen's method to multiply two matrices, A and B, of size $n \times n$, where $n = 2^k$.

Input: Read the input data from a text file whose format is as follows

- 1st line: a positive integer n to indicate the size of the matrices ($n > 2$)
- n next lines present the matrix A, each of which has n integers. Two consecutive numbers are separated by a single space " ".
- n last line shows the matrix B with the similar format as A's.

Output: Print to the console the product of the given two matrices.

Example of input and output:

Input from user keyboard	Output to console

2
1 3
5 7
2 4
6 8
	20 28
	52 76

Q5. The closest-pair problem.

Problem statement: Let P be a list of $n > 1$ points in the Cartesian plane: $P = \{p_1, p_2, \ldots, p_n\}$. Apply the Divide-and-conquer approach to find a pair of points with the smallest distance between them.

Input: Read the input data from a text file whose format is as follows

- 1st line: a positive integer n to indicate the number of points in the Cartesian plane.
- n next lines: Each line represents the two-dimensional coordinate of a point, in which the two coordinate values are separated by single space " ".

Output: Print the output data to the console the pair of points with the smallest distance.

Example of input and output:

Input text file	Output to console

6
2 3
12 30
40 50
5 1
12 10
3 4
	2 3
	3 4
	1.41421

Q6. The change-making problem.

Problem statement: Given k denominations: $d_1 < d_2 <..< d_k$ where $d_1 = 1$. Apply the Divide-and-conquer approach to find the minimum number of coins (of certain denominations) that add up to a given amount of money n.

Input: Read the input data from a text file whose format is as follows

- 1st line: a positive integer k to indicate the number of denominations.
- 2nd line: k positive integers describing k denominations, sorted descending, two consecutive numbers are separated by a single space " ". The last value must be one.

- 3rd line: a positive integer n to indicate the amount of money required exchange.

Output: If there exists a solution, print the amount of each denomination to the console. Otherwise, output the string "No solution".

Example of input and output:

Input text file	Output to console

4
25 10 5 1
72
	25: 2
	10: 2
	5: 0
	1: 2

Note that you only need to implement the upgraded version shown in the lecture.

Regulations for completing the lab work

- Each question must be implemented as an independent program in a single C++ file (of format .cpp).
- The program must receive input and return output as specified Submissions with wrong regulation will result in a "0" (zero).
- Plagiarism and Cheating will result in a "0" (zero) for the entire course.
- Contact: Here.