Assignment 2

Problem 1:

Given an array of ‘n’ Integers. Write a program to find out the kth smallest element where k is given as input.

Input: First line denotes the value of t where t is no of test cases. In each test case first line denotes the values of ‘n’ and ‘k’ seperated by spaces. The second line of each test case consists of ‘n’ integers.

Output: kth smallest integer needs to be printed.

Constraints:

1 < t < 100
1 < n < 10^6

Sample Input:
1
5 3
5 8 1 6 4

Sample Output:
5

Problem 2:

Given two strings of any length. Write a program to find the longest common subsequence.

Input: first line denotes the value of ‘t’ where t is no of test cases. Each test case consists of two strings in two different lines.

Output: Length of Longest common subsequence needs to be printed.

Constraints

1 < t < 100
1 < len(input_str) < 10^5

Sample Input

1
ABACCD
ACDF

Sample Output

3

Problem 3:

Write a program to print the Longest common subsequence instead of its length

Input/output formats and constraints are exactly same as they are in problem 2

Sample Input:

1
ABACCD
ACDF

Sample Output:

ACD

Note: All the Test Cases are of type having only one longest common subsequence. No need to worry about multiple possibilities of LCS.

Problem 4:

Write a program to find out the length of a shortest path between source and destination where a graph G is given as input in the format described below.

Input:
In the first line integer ‘t’ indicates number of test cases.
In each test case first line indicates integer ‘n, m’ seperated by space where n is no of vertices(including source and destination) and m stands for the number of edges.
After that each line indicates integers: a b w where there is an edge of weight w from vertex a to vertex b.

Note: Vertices are named from 1 to n where source being 1 and destination being n

Output:

length of the shortest path from source to destination.

Sample Input:

```
1
7 8
1 2 1
1 3 1
2 4 2
2 5 3
3 6 2
4 7 3
5 7 4
6 7 2
```

Sample Output:

```
5
```

Problem 5

Write a program to find out codes for given charecters based on their frquencies

Input:
First Line contains T, the number of test cases.
For each test case, first line has the number of symbols, frequencies that will be given as input N
From next N lines read symbol followed by frequency.

Output:
Print the huffman code for each symbol in the order of the input symbols.

Note: While constructing the tree, keep less frequent child on left side whereas high frequent child on right side. No two nodes of a tree will have equal frquencies.

Sample Input:

```
1
5
A 24
B 12
C 10
```

D 8
E 9

Sample Output:

0
111
110
100
101

Bonus Question:

Huffman Coding: Write a program with two functions.
Function 1: Takes input paragraph from input file and generated a new ‘encoded.txt’ file with the ASCII Format of input.

Function2: Takes ‘encoded.txt’ file as input and generates an output file with decoded string which shoulp match the given input paragraph.