You are allowed to discuss with others but are not allowed to use any references other than the course notes and the three reference books. Please list your collaborators for each question. You must write your own solutions. See the course outline for the homework policy.

There are totally 54 marks. The full mark is 50. This homework is counted 8% of the course grade.

1. **Programming Problem: Road Trip** (20 marks)

   Alice has just earned her driving license and she can't wait to go on a road trip!

   Alice starts at city 1 and will visit cities $2, 3, \ldots, N$ in order. It takes T_i time to drive from city i to city $i+1$ ($1 \leq i < N$).

   The most important thing for Alice to keep in mind of is her car's fuel level – if the car runs out of fuel in the middle of the road, Alice will be stranded! The car's fuel tank capacity is C, and the fuel level is initially full. Each unit of time of driving decreases fuel level by 1. Fortunately, at each city there is a gas station. Adding one unit of fuel to the car at city i takes R_i time and costs D_i dollars. Alice can refuel as much as she wants, as long as the fuel level does not exceed C.

   Alice is travelling on a budget and will bring M dollars with her on the trip. Please determine the minimum time needed to drive from city 1 to city N.

   Input: The first line has three integers N, M, and C, respectively the number of cities, the budget, and the fuel tank capacity. The second line has $N-1$ integers $T_1, T_2, \ldots, T_{N-1}$, the time needed to drive from city i to city $i+1$. The third line has N integers $R_1, R_2, \ldots, R_N$, the unit time cost of fuel in each city. The fourth line has N integers $D_1, D_2, \ldots, D_N$, the unit money cost of fuel in each city.

   It is guaranteed that $2 \leq N \leq 20$, $0 \leq M \leq 2000$, and $1 \leq C \leq 500$; $1 \leq T_i \leq 500, 1 \leq R_i \leq 50$, and $1 \leq D_i \leq 2000$. Furthermore, for test cases worth 90% of the points, it is guaranteed that $0 \leq M \leq 500$ and $1 \leq C \leq 100$.

   Output: If it is impossible to reach city N, output `-1` on a single line. Otherwise, output two lines. On the first line, output the minimum time needed to travel to city N. On the second line, output N integers $f_1, f_2, \ldots, f_N$, where f_i is the amount of fuel added to the car at city i.

   If there are multiple refueling plans that minimize travel time, output any of them.

   Sample Input 1:
   ```
   4 500 100
   50 30 25
   1 1 20 1
   16 17 16 15
   ```
   Sample Output 1:
   ```
   110
   0 5 0 0
   ```

   Sample Input 2:
   ```
   4 80 100
   ```

```
50 30 25
1 1 20 1
16 17 16 15
```

Sample Output 2:

```
205
0 0 5 0
```

Sample Input 3:

```
4 80 100
50 30 26
1 1 20 1
16 17 16 15
```

Sample Output 3:

```
-1
```

2. **Written Problem: Facility Location** (12 marks)

   Suppose there is a new town where everyone lives in a long street. The government has the budget to build k hospitals. There are m possible locations and the government would like to choose a subset of k locations so as to minimize the total distance to the people. You are asked to solve the following problem.

   Input: n nonnegative numbers $a_1 \leq a_2 \leq \ldots \leq a_n$ representing the locations of the people, m nonnegative numbers $p_1 \leq p_2 \leq \ldots \leq p_m$ representing the possible locations of the hospitals, and an integer $1 \leq k \leq m$. We assume that $m \leq n$.

   Output: The minimum total distance to the people. Suppose $S = \{l_1, \ldots, l_k\} \subseteq \{1, \ldots, m\}$ is a subset of k hospitals. We assume that every person will go to the nearest hospital, and so we define $\text{dist}(i, S) = \min_{j \in S} |a_i - p_j|$ for person i. The objective is to choose a subset S of k hospitals so as to minimize the total distance defined as $\sum_{i=1}^{n} \text{dist}(i, S)$. You just need to output one number, the minimum total distance. There is no need to output an optimal subset S.

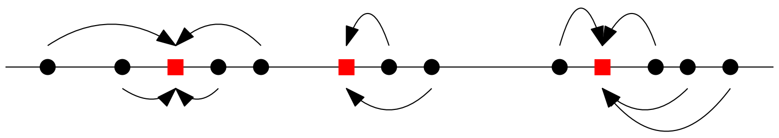

Figure 1: The black circles are people. The red squares are hospitals. Each person goes to the nearest hospital and this is how the distances are defined. In this example $k = 3$.

   Design the fastest algorithm that you can to solve this problem. Prove the correctness and analyze the time complexity.

3. **Written Problem: 3-Independent Set** (12 marks)

   Given an undirected graph $G = (V, E)$, a subset $S \subseteq V$ is a 3-independent set if $\text{dist}(u, v) \geq 3$ for every pair of vertices $u, v \in S$, where $\text{dist}(u, v)$ is the shortest path distance from u to v in G. We are interested in finding a 3-independent set of maximum total weight, but this problem is NP-hard on general graphs. Show that this problem is easy on trees.

   Input: A tree $T = (V, E)$ in the adjacency list representation, a weight w_v for each vertex $v \in V$.

   Output: A 3-independent set $S \subseteq V$ that maximizes the total weight $\sum_{v \in S} w_v$.

   Design the fastest algorithm that you can to solve this problem. Prove the correctness and analyze the time complexity.

4. **Written Problem: Trading** (10 marks)

   Suppose somehow we could use 1 unit of metal to trade for 2 units of wood, and 1 unit of wood to trade for 0.4 unit of glass, and 1 unit of glass to trade for 1.5 units of metal. Then we can start with 1 unit of metal, and end up with $1 \times 2 \times 0.4 \times 1.5 = 1.2$ units of metal, and make profit by just trading. Help your company to solve the following problem.

   Input: n items $a_1, \ldots, a_n$ and an $n \times n$ table T of trading rates, such that one unit of item a_i can trade for $T[i, j]$ units of item a_j, where $T[i, j] \geq 0$ for $1 \leq i, j \leq n$.

   Output: Whether or not there exists a sequence of items $a_{i_1}, \ldots, a_{i_k}$ such that $T[i_1, i_2] \cdot T[i_2, i_3] \cdot \cdots \cdot T[i_{k-1}, i_k] \cdot T[i_k, i_1] > 1$.

   Design the fastest algorithm that you can to solve this problem. Prove the correctness and analyze the time complexity.