Programming Design, Spring 2016
Midterm Project

Instructor: Ling-Chieh Kung
Department of Information Management
National Taiwan University

IEDO Airlines is a big airline company based in Taiwan. Most of flights operated by IEDO connect TPE (Taiwan Taoyuan international airport) and another airport somewhere in the world. The Ground Service Department (GSD) of IEDO is responsible for managing all the customer-related affairs at TPE. In GSD, the most critical resource is of course the agents, and the most critical task in GSD is agent scheduling, i.e., to assign agents the right tasks to do at the right time. With hundreds of agents, 24 hours a day and 7 days a week service, and various kinds of tasks, agent scheduling is a big challenge.

Robin was hired into IEDO two weeks ago as the head of the Operations Research team. She was asked to help GSD improve the efficiency of agent scheduling. In the past two weeks, she found that GSD heads are not helpful: The director of GSD, Brook, plays violin all the days, and the vice director, Zoro, only cares about drinking and fighting. Moreover, the head of the agent group, Sanji, spends all his time improving his cooking skill. The three managers together made agents scheduled in a very bad way: Sometimes a lot of customers are served by no agent while sometimes a lot of agents have nothing to do. That is why the president of IEDO, Luffy, needs Robin to take over the job. The good news is that Brook, Zoro, and Sanji completely trust her and will not make any suggestion (which will most likely be useless).

A senior agent Vivi is quite experienced and knowledgeable. She knows all the rules and objectives about agent scheduling. Vivi explained all she knows to Robin as follows.

1 Basics of agent scheduling

1.1 Boarding and check-in

There are two missions in GSD: *boarding* and *check-in*. Boarding is for agents to help customers board their flights at boarding gates right before airplanes take off. Check-in is for agents to verify the validity of customers' passport and tickets, issue boarding passes, and check customers' luggage, etc., at check-in counters. GSD currently has $N + M$ agents. These agents are split into two divisions, one for boarding and one for check-in. The boarding division owns M agents while the check-in division owns N agents. An agent in one division does not do the tasks in the other division.

1.2 Skills and groups

IEDO helps three foreign airline companies (to earn money, of course) for serving their customers in TPE. They are ANTS airlines, SVVRL airlines, and KMM airlines. Boarding tasks for different airlines are similar. However, check-in is more complicated because it requires an agent to operate information systems that are different from airlines to airlines. An agent must be trained to operate an airline's system before she can do the check-in tasks for that airlines company. Agents who can operate the same set of airlines check-in systems are said to be in the same *group*. Following this logic, we say that all agents in the boarding division are in the same group.

Currently, all check-in agents knows either the IEDO system only or the IEDO system plus one additional system. Therefore, we have four check-in groups and one boarding groups. The Chief Human Resource Officer (CHRO) Usopp has labeled the numbers of agents in the four check-in groups IEDO, ANTS, SVVRL, and KMM are N_I, N_A, N_S, and N_K, respectively. For example, there are N_A agents who are able to operate the systems of IEDO airlines and ANTS airlines. Obviously, $N_I + N_A + N_S + N_K = N$.

1.3 Tasks and demands

IEDO divides each day into three *shifts*: 6–14, 14–22, and 22–6. If an agent works on a day, she/he will work in one of the three shifts; otherwise she/he will has a *day off*. IEDO has a special position, Chief Medicine Officer (CMO). The CMO Chopper insists that one cannot work in two shifts in the same day. However, to make the agent scheduling problem easier, Chopper agrees that an agent may be assigned to work in shift 22–6 in a day and then in shift 6–14 in the next day. He will be responsible for making these agents become energetic again after the long time of work. Flights and agents are scheduled in a weekly basis. One must have two days off in a week.

There are five kinds of *tasks*: Check-in for the four airlines companies and boarding. Before each season starts, the schedule of flights will be determined. According to the number of flights, the capacities of airplanes, their destinations, etc., the number of agents needed for a task can be calculated for each shift in each day.

Calculating all these agent *demands* is of course uneasy. Fortunately, the Chief Information Officer (CIO) Franky has built a powerful information system to do this. Before Robin schedules agents, she will get the demand information as the number of agents required for each task in each shift of each day. More precisely, let's label 6–14, 14–22, and 22–6 as shifts 1, 2, and 3, respectively, and Monday, Tuesday, ..., and Sunday as days 1, 2, ..., and 7, respectively. Robin will get D^t_{ds} as the number of agents required for task t in shift s of day d, where $t \in \{I, A, S, K, B\}$ means check-in for IEDO, check-in for ANTS, check-in for SVVRL, check-in for KMM, and boarding, $s \in \{1, 2, 3\}$, and $d \in \{1, 2, ..., 7\}$. Note that even if an agent who can operate two types of systems works in a shift, she/he can be assigned to only one of the airlines companies.

1.4 Wages and agent hiring

If the current 1000 full-time agents are not enough to satisfy all agent demands, more agents may be hired. IEDO may hire either full-time or part-time agents. A full-time agent works five days a week, one shift in a day. The wages for a boarding agent and an IEDO-only check-in agent are different. Moreover, knowing one additional check-in system gives an agent a higher wage. A part-time agent can work for any number of shifts in a week. However, a part-time agent can only be trained to operate the IEDO check-in system or do the boarding task. The number of new agents, full-time or part-time, that can be hired is unlimited.

The Chief Financial Officer (CFO) Nami provides the salary information to Robin. The weekly wages for a full-time agent for boarding, IEDO-only check-in, and IEDO-plus-one check-in are F^B, F^I, and F^{II}, respectively. The wages per shift for a part-time agent for boarding and IEDO-only check-in are P^B and P^I, respectively.

Finally, an extra compensation C per week must be given to full-time agents whose two off days are separated (Sunday and Tuesday, Wednesday and Saturday, etc.). If a full-time agent's two off days are consecutive (Sunday and Monday, Tuesday and Wednesday, Saturday and Sunday, etc.), she/he does not get the compensation.

2 Objectives of agent scheduling

Robin, Nami, and Usopp spent a few hours to come up with several restrictions and objectives of agent scheduling. The current $N + M$ agents cannot be fired. Their skills are also fixed and cannot be changed. In fact, Usopp believes that the current $N + M$ agents are far from enough to satisfy all the agent demands. Usopp really wants to hire more agents (if possible, he actually wants 8000 agents!), but he needs to convince Nami that new agents are really needed.

Robin is here to do the job. If an optimal (or at least near-optimal) schedule still cannot satisfy all the agent demands, new agents are really needed. Robin needs to find an optimal/near-optimal schedule and convince Nami that it is indeed optimal/near-optimal. Therefore, Robin's agent scheduling should answer the following questions: *If, beside the current 1000 agents, IEDO can hire both full-time and* *part-time agents, how to satisfy all demands while minimizing the total wages?* Please note that for a schedule to be executable, it must be detailed enough to tell each single agent when to work and which task to work for. Suggesting only a number of additional agents to hire is not sufficient.

3 Tasks for this project

3.1 Problem 1: agent scheduling for the boarding division

You will be given all the information mentioned above for you to do agent scheduling for the boarding division. The data will be given in the following format. In each input file, there are three lines of integers:

- In the first line, there is one single integer M, the number of full-time agents already hired in the boarding division. It is known that $100 \leq M \leq 10000$.
- In the second line, there are 21 integers D^B_{11}, D^B_{12}, D^B_{13}, D^B_{21}, ..., and D^B_{73}, where D^B_{ds} is the number of agents required for boarding in shift $s \in \{1, 2, 3\}$ of day $d \in \{1, 2, ..., 7\}$. It is known that $0 \leq D^B_{ds} \leq 1000$.
- In the third line, there are three integers F^B, P^B, and C, which are the weekly wage for a full-time agent in the boarding division, the per shift wage for a part-time agent in the boarding division, and the per week compensation for a full-time agent having nonconsecutive days off. It is known that $100 \leq F^B \leq 1000$, $10 \leq P^B \leq 500$, and $0 \leq C \leq 500$.

In each line, two consecutive integers are separated by a white space.

As an example, consider the following input:

```
300
110 90 40 80 60 30 150 100 50 30 20 30 70 60 20 160 120 50 120 100 30
200 60 80
```

The first line means that there are 300 full-agents. The second line tells us that we need 110 in 6–14 on Monday, 90 in 14–22 on Monday, ..., and 30 in 22–6 on Sunday (22:00 Sunday to 6:00 on Monday). The last line tells us that the weekly wage for a full-time agent, the per shift wage for a part-time agent, and the per week compensation for nonconsecutive days off are \$200, \$60, and \$80, respectively.

Given the input, you should try to find a hiring plan and an agent schedule to minimize the *total additional wage* paid to hire *new agents* plus *all compensations* while satisfying all demands. In short, you want to minimize the total cost of your schedule (excluding the sunk wage for existing full-time agents). Of course, you may have no idea about how to find an optimal solution (i.e. a cost-minimizing schedule). In this project, you only need to try your best to make your schedule as good as possible. The points you earn for each instance is $2(\frac{z^*}{z})$, where z is the total cost of your schedule and z^* is that of a cost-minimizing schedule, as long as your plan is feasible. If your plan is infeasible, you will get 0.

To output your plan, follow the rules below:

- In the first line, output two integers f and p, the number of new full-time and part-time agents hired, respectively.
- In the second line, output 21 integers x^B_{12}, x^B_{13}, ..., x^B_{17}, x^B_{23}, x^B_{24}, ..., and x^B_{67}, where x^B_{ij} means the number of full-time agents whose days off are days i and j, $i = 1, ..., 6$, $j = i + 1, ..., 7$, respectively.
- In the third line, output 21 integers y^B_{11}, y^B_{12}, y^B_{13}, y^B_{21}, ..., and y^B_{73}, where y^B_{ds} is the number of full-time agents scheduled in shift $s \in \{1, 2, 3\}$ of day $d \in \{1, 2, ..., 7\}$.
- In the fourth line, output 21 integers p^B_{11}, p^B_{12}, p^B_{13}, p^B_{21}, ..., and p^B_{73}, where p^B_{ds} is the number of part-time agents scheduled in shift $s \in \{1, 2, 3\}$ of day $d \in \{1, 2, ..., 7\}$.

In each line, two consecutive integers are separated by a white space.

For the example above, an optimal schedule can be expressed in the following way:

```
0 120
130 0 0 0 0 20 0 0 0 0 0 0 0 0 0 150 0 0 0 0 0
20 90 40 80 60 30 150 100 50 100 20 30 70 60 20 130 120 50 150 100 30
90 0 0 0 0 0 0 0 0 0 0 0 0 0 0 30 0 0 0 0 0
```

This means to hire no new full-time agent and 120 part-time agents (one for a shift). 130 full-time agents have days off on days 1 and 2, 20 full-time agents have days off on days 1 and 7, and 150 full-time agents have days off on days 4 and 5. 90 and 30 part-time agents are hired to work in shift 1 of days 1 and 6, respectively. As no full-time agent has nonconsecutive days off, the total cost is $\$200 \times 0 + \$60 \times 120 = \$7200$.

Your output needs not to give an optimal schedule. PDOGS will read your input and check feasibility. If it is feasible, PDOGS will compute the total wage z of your schedule. You then obtain $2(\frac{7200}{z})$ points.

3.2 Problem 2: agent scheduling for the check-in division

You will be given all the information mentioned above for you to do agent scheduling for the check-in division. The data will be given in the following format. In each input file, there are three lines of integers:

- In the first line, there are four integers N_I, N_A, N_S, and N_K, the number of full-time agents already hired in the check-in division in the IEDO, ANTS, SVVRL, and KMM group, respectively. It is known that $100 \leq N_i \leq 10000$, $i \in \{I, A, S, L\}$.
- In each of the second to fifth line, there are 21 integers D^i_{11}, D^i_{12}, D^i_{13}, D^i_{21}, ..., and D^i_{73}, where D^i_{ds} is the number of group-i agents required for boarding in shift $s \in \{1, 2, 3\}$ of day $d \in \{1, 2, ..., 7\}$. The second, third, fourth, and fifth lines are for the IEDO, ANTS, SVVRL, and KMM group, respectively. It is known that $0 \leq D^i_{ds} \leq 1000$.
- In the sixth line, there are four integers F^I, F^{II}, P^I, and C, which are the weekly wage for a full-time agent in the IEDO group, the weekly wage for a full-time agent in the other three groups, the per shift wage for a part-time agent in the IEDO group, and the per week compensation for a full-time agent having nonconsecutive days off. It is known that $100 \leq F^I \leq 1000$, $100 \leq F^{II} \leq 1000$, $10 \leq P^I \leq 500$, and $0 \leq C \leq 500$.

In each line, two consecutive integers are separated by a white space.

Given the input, you should try to find a hiring plan and an agent schedule to minimize the *total additional wage* paid to hire *new agents* plus *all compensations* while satisfying all demands. In short, you want to minimize the total cost of your schedule (excluding the sunk wage for existing full-time agents). The points you earn for each instance is $2(\frac{z^*}{z})$, where z is the total cost of your schedule and z^* is that of a cost-minimizing schedule, as long as your plan is feasible. If your plan is infeasible, you will get 0.

To output your plan, follow the rules below:

- In the first line, output five integers f^I, f^A, f^S, f^K and p, the number of new full-time agents for the IEDO group, ANTS group, SVVRL group, KMM group, and part-time agents hired, respectively.
- In each of the second to fifth line, output 21 integers x^g_{12}, x^g_{13}, ..., x^g_{17}, x^g_{23}, x^g_{24}, ..., and x^g_{67}, where x^g_{ij} means the number of group-g full-time agents, $g \in \{I, A, S, K\}$, whose days off are days i and j, $i = 1, ..., 6$, $j = i + 1, ..., 7$, respectively. The second, third, fourth, and fifth line are for group IEDO, ANTS, SVVRL, and KMM, respectively.
- In each of the sixth to ninth line, output 21 integers y^g_{11}, y^g_{12}, y^g_{13}, y^g_{21}, ..., and y^g_{73}, where y^g_{ds} is the number of group-g full-time agents, $g \in \{I, A, S, K\}$, scheduled to do the IEDO task in shift

$s \in \{1, 2, 3\}$ of day $d \in \{1, 2, ..., 7\}$. The sixth, seventh, eighth, and ninth line are for group IEDO, ANTS, SVVRL, and KMM, respectively.

- In the tenth line, output 21 integers p^I_{11}, p^I_{12}, p^I_{13}, p^I_{21}, ..., and p^I_{73}, where p^I_{ds} is the number of part-time agents scheduled to do the IEDO task in shift $s \in \{1, 2, 3\}$ of day $d \in \{1, 2, ..., 7\}$.

In each line, two consecutive integers are separated by a white space.

As an example, consider the following input:

```
200 170 150 180
160 100 70 120 80 40 130 140 30 100 110 20 50 60 80 180 160 40 100 120 50
80 40 50 60 40 70 100 50 80 70 50 100 60 60 80 110 100 150 90 80 90
70 80 80 70 70 50 110 40 80 80 50 90 100 80 40 70 90 80 90 70 60
70 90 60 100 80 80 120 110 140 130 80 90 120 120 100 100 110 110 90 80 70
180 280 50 80
```

For the example above, an optimal schedule can be expressed in the following way:

```
105 190 160 237 190
30 0 0 0 0 35 105 0 0 0 0 0 0 0 0 135 0 0 0 0 0
0 0 0 0 0 100 120 0 0 0 0 0 0 0 0 140 0 0 0 0 0
50 0 0 0 0 30 70 0 0 0 0 10 0 0 0 80 0 0 10 0 60
117 0 0 0 0 80 40 0 0 0 0 6 0 0 0 77 0 0 0 0 97
70 100 70 50 80 40 119 51 30 40 110 20 30 60 80 180 125 0 100 120 50
90 0 0 70 0 0 10 0 0 0 0 0 20 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 34 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 89 0 26 0 0 0 0 0 0 35 40 0 0 0
```

This means to hire 105 new full-time agent for the IEDO group, $190 + 160 + 237 = 587$ new full-time agent for the other three groups, and 190 part-time agents (one for a shift). As no full-time agent has nonconsecutive days off, the total cost is $\$180 \times 105 + \$280 \times 587 + \$50 \times 190 = \$192760$. Your output needs not to give an optimal schedule. PDOGS will read your input and check feasibility. If it is feasible, PDOGS will compute the total wage z of your schedule. You then obtain $2(\frac{192760}{z})$ points.

4 Submission rules and grading

Students should form teams to work on this project. Each team should have ***at most four students***.

4.1 Online submissions to PDOGS

Each team should submit their C++ programs to PDOGS. PDOGS will input testing data and check the output to give grades. For each of Problems 1 and 2, there are 15 instances. They together give at most 60 points to a team. Each team should also write (I mean, type) a report. In the report, please describe your algorithm (in words, not in codes), design of your system, and work division among team members. Your report cannot be longer than ***six pages*** (i.e., three double-sided sheets). It counts for 20 points and may be written in English or Chinese. The submission deadline is ***2:00 am, May 9, 2016***.

4.2 Demonstration of your system

Each team should also demonstrate your system to the TAs at the lecture time on ***May 9, 2016***. The key is to design a user-friendly output format. Each team may design its own way to present the proposed schedule generated by their system. The demonstration counts for 20 points. The demonstration schedule will be announced when the number of teams is fixed.