CS 345 — Fall 2001	**Programming Assignment #2**
Programming Languages	**Due December 7, 2001**

General Notes

Read these directions carefully.

Turn in a file called "README" that contains at least your full name and your login name.

You may use either Java or C++ to do this assignment. If you use Java, the class with a main() method must be in "University.java". If you use C++, include in your README file the command used to compile your code to the executable and make sure that your executable is called "university". Also, make absolutely sure that your code compiles and runs on the sun systems on campus, because if we cannot compile or run your program, your grade will suffer significantly.

You will need to turn in both a written and an electronic copy of your source code (do not turn in binaries). To turn in the electronic copy, put your source code and README file in a directory called "proj2" and then type "/p/bin/turnin -submit reeber cs345-lin proj2". Turn in the written copy in the usual way. Both are due by 5pm.

Feel free to discuss algorithms with each other and to look up algorithms in books. However, do not show each other code or look at anyone else's code. Copying code or down-loading code from the Internet is cheating and will lead to failure of the course.

One third of your grade on this assignment will be based on code readability, and two-thirds will be based on how well your program works.

If you wish, you may use code or algorithms from the Programming Assignment 1 solution, but, in that case, reference the solution in your README file.

Problem Description

The following problems expand on the Programming Assignment 1. For each problem, expand your university simulation to handle the expansion. Also, on your paper copy, for each problem write the classes that were added and the classes that were modified by that problem.

1. Add a new type of Academic project, called a secret (government) project, to your simulation. In the input, the number of secret projects a Professor creates when he or she is working without pay will appear after the number of industrial projects that Professor creates. Thus, if the following input is used:

```
2 2 1 1 1 1 1
Lin C 2 0 1 1
Shwartz P 0 0 1 0
```

Lin will generate one secret project and Shwartz will create zero.

Also, the following rules apply concerning secret projects (as well as any general project-rules from before):

- When a Professor is generating project, he or she generates crazy projects, then fundamental projects, then industrial projects, then secret projects.
- A Professor working on a secret project generates no fundamental or industrial projects.
- A Professor working on a secret Math project generates a secret CS project.

2. Add projects that take multiple days. The new input, will look as follows:

$n\ f\ a\ b\ c\ A_m\ I_m\ aa\ bb\ cc$

$Name_0\ Department_0\ Crazy_0\ Fundamental_0\ Industrial_0\ Secret_0$

$Name_1\ Department_1\ Crazy_1\ Fundamental_1\ Industrial_1\ Secret_1$

...

$Name_{f-1}\ Department_{f-1}\ Crazy_{f-1}\ Fundamental_{f-1}\ Industrial_{f-1}\ Secret_{f-1}$

Here, aa, bb, and cc are non-negative integers. The number of days a project will take to complete is equal to

$$D(i) = ((aa * i + bb) mod\ cc) + 1$$

Assume a Professor *Prof* works on a project *proj*, with id i. *Prof* will receive the pay for *proj* on the day*proj* is picked, and on all other days that *Prof* works on *proj*. However, *Prof* will not work on any other project or generate any projects until *proj* is complete.

Note that D(i) = 1 is the old case, where a professor waits until the end of that day before generating projects. All the old modifiers and rules still hold when the Professor generates projects again.

3. The faculty at our university will no longer consist only of Professors. Now we will include a new type of faculty, called Deans. The new input stream will be as follows:

$n\ f\ a\ b\ c\ A_m\ I_m\ aa\ bb\ cc$

$Name_0\ Type_0\ Department_0\ Crazy_0\ Fundamental_0\ Industrial_0\ Secret_0$

$Name_1\ Type_1\ Department_1\ Crazy_1\ Fundamental_1\ Industrial_1\ Secret_1$

$Name_2\ Type_2\ Department_2\ Crazy_2\ Fundamental_2\ Industrial_2\ Secret_2$

...

$Name_{f-1}\ Type_{f-1}\ Department_{f-1}\ Crazy_{f-1}\ Fundamental_{f-1}\ Industrial_{f-1}\ Secret_{f-1}$

Here, $Type_i$ will be 'D' if the person named $Name_i$ is a Dean, and 'P' if the person is a professor.

When it comes to picking projects and working on projects Deans act exactly like Professors. A Dean will work on and receive money for the project than he or she picks. Rather than generating projects, however, a Dean will increase the value of other people's projects by hyping them up. The following rules apply for Deans:

- Each faculty member will get a turn to increase or generate projects each day. Just as in the previous project, on day d, the faculty with id d will go in sequence.
- If a Dean is working on a project that takes m days, then that Dean will not increase any projects until he or she has worked for m days. Note that a Dean increases a batch of projects exactly when, if he were a Professor, he would generate a batch of projects.
- When a Dean is not working on a project, he or she does not increase the value of any projects.
- If a Dean d is working on a project created by Professor *prof*, then d will only increase projects created by *prof*.
- The Dean $Name_i$ will consider increasing no more than $Crazy_i$ crazy projects, $Fundamental_i$ fundamental projects, $Industrial_i$ industrial, and $Secret_i$ secret projects.
- A Dean considers the most recent projects possible. For example, assume that on day 3, Dean d finish working on a project created by Professor *prof*. Furthermore assume that d can consider a max of 2

crazy projects. Finally, assume *prof* creates crazy projects 0, 4, and 7 before *d* increases any projects. Then, by this rule, *d* considers projects 4 and 7.

- Let project *proj* be under consideration by Dean *d*. If *proj* has been picked to work on by any faculty member, then *d* will not increase the pay of *p*.

  Note that a Dean can only **consider** $Crazy_i$ crazy projects. Even if none of these projects are actually increased, no other projects will be increased or considered. Hence, if in our previous example project 4 has already been worked on (or is currently being worked on), then only project 7 will be increased.

- Each act of increasing a project will receive a unique increase id. These increase ids use the same identifier system as project ids. For example, if project ids 0 through 2 have been generated and a project is now increased, the act of increasing gets the id 3. If our next move is to generate a project, that project gets id 4.

- For any increase id i, the amount the pay for a project is increased is $A_m * P(i)$ if the project is an academic project, and $I_m * P(i)$ if the project is an industrial project.

- When a Dean hypes multiple projects on a single day, the Crazy projects are hyped first, then the Fundamental projects, then the Industrial projects, and then the Secret projects. If multiple projects of the same type are being hyped, then the project are hyped in order from lowest project id to highest project id.

  Note that this rule is a little odd. A Dean considers only the *most* recent crazy projects generated by a given Professor. However, if multiple crazy projects are increased, then the *least* recent project is increased first.

  Due to a mistake by your TA, the following alternate rule can be used: When a Dean hypes multiple projects on a single day, that deans hypes projects in order from lowest project id to highest project id. Note that the example in the following section will use the original rule (not this alternate one).

- Note that the modifiers that affect generating projects (like a person working on a crazy project generates no industrial projects) do not apply increasing projects.

- A Dean who is working on a secret project does not increase the value of any project that is not a secret project.

- Faculty use the full payment for a project, including any increases made by Deans, to decide which project to pick.

For output, print the following:

$Name_1$ $DollarsMade_1$ $CreatedDollars_1$

$Name_2$ $DollarsMade_2$ $CreatedDollars_2$

...

$Name_f$ $DollarsMade_f$ $CreatedDollars_f$

Here, $DollarsMade_i$ is the sum of the payment of all the projects that the person with $Name_i$ picked. For a Professor, $CreatedDollars_i$ is the sum of the payments originally assigned to projects that he or she created and were eventually picked. For a Dean, $CreatedDollars_i$ is the sum of the increase in payments due to that Dean on projects that were eventually picked.

4. Which classes were you able to reuse, and which classes were not modified by any of the above problems?

Example

Imagine that in a file called in3.txt, we store the following:

```
6 5 13 11 17 2 5 1 0 2
Snyder P P 0 3 0 0
Tien D C 0 0 10 10
Lin P C 2 0 2 1
Marshal D C 2 3 2 2
Wiles P M 1 0 1 0
```

Now we type in the following (assuming a Java implementation) and get the following output:

```
> java University < in3.txt
Snyder 178 153
Tien 412 438
Lin 226 417
Marshal 393 313
Wiles 295 183
```

To understand this output, first note that the Pay function is P(id) = [((13*id+11) mod 17) + 1]. This will get multiplied by 2 for academic projects and 5 for industrial projects. Also, the number of Days function is Day(id) = [((1*id+0) mod 2) + 1].

```
Day 0:
Person# Name     Picked   Working-On  Dollars-Payed
0       Snyder   ----
1       Tien     ----
2       Lin      ----
3       Marshal  ----
4       Wiles    ----

Generated or Increased Projects (the first three fields are blank if we
are generating a project, the Pay includes the current increase):
Increasor IncreaseID PayIncrease ProjectId CreatorName     Field    Type        Pay    Days
---       ---        ---         0         Snyder          P        Fundamental 24     1
---       ---        ---         1         Snyder          P        Fundamental 16     2
---       ---        ---         2         Snyder          P        Fundamental 8      1
---       ---        ---         3         Lin             C        Crazy       34     2
---       ---        ---         4         Lin             C        Crazy       26     1
---       ---        ---         5         Lin             C        Industrial  45     2
---       ---        ---         6         Lin             C        Industrial  25     1
---       ---        ---         7         Lin             C        Secret      2      2
---       ---        ---         8         Wiles           M        Crazy       28     1
---       ---        ---         9         Wiles           M        Industrial  50     2
```

The first day looks normal. Note that Marshal and Tien do nothing, since they are not working on a Project yet. Also, you should follow project 7, which will be increased many times.

```
Day 1 (Days-Left is the number of Days left on the project
once today day is done):
```

Person#	Name	Picked	Working-On	Dollars-Payed	Days-Left
1	Tien	9	9	50	1
2	Lin	5	5	45	1
3	Marshal	3	3	34	1
4	Wiles	8	8	28	0
0	Snyder	0	0	24	0

```
Generated or Increased Projects (the first three fields are blank if we
are generating a project, the Pay includes the current increase):
```

Increasor	IncreaseID	PayIncrease	ProjectId	CreatorName	Field	Type	Pay	Days
---	---	---	10	Wiles	M	Crazy	12	1
---	---	---	11	Snyder	P	Fund	4	2
---	---	---	12	Snyder	P	Fund	30	1
---	---	---	13	Snyder	P	Fund	22	2
---	---	---	14	Snyder	P	Ind	35	1

Day 1 is also pretty uninteresting. Note, however, that the faculty members working on projects that take more than one day did not generate or increase any projects.

```
Day 2:
```

Person#	Name	Picked	Working-On	Dollars-Payed	Days-Left
2	Lin	---	5	45	0
3	Marshal	---	3	34	0
4	Wiles	14	14	35	0
0	Snyder	12	12	30	0
1	Tien	---	9	50	0

```
Generated or Increased Projects (the first three fields are blank if we
are generating a project, the Pay includes the current increase):
```

Increasor	IncreaseID	PayIncrease	ProjectId	CreatorName	Field	Type	Pay	Days
---	---	---	15	Lin	C	Ind	15	1
---	---	---	16	Lin	C	Ind	80	2
---	---	---	17	Lin	C	Secret	24	1
Marshal	18	16	4	Lin	C	Crazy	42	2
Marshal	19	20	15	Lin	C	Ind	35	1
Marshal	20	85	16	Lin	C	Ind	165	2
Marshal	21	26	7	Lin	C	Secret	28	1
Marshal	22	18	17	Lin	C	Secret	42	2
---	---	---	23	Wiles	C	Ind	25	1
---	---	---	24	Wiles	M	Ind	5	2
---	---	---	25	Snyder	P	Fund	28	1
---	---	---	26	Snyder	P	Fund	20	2
---	---	---	27	Snyder	P	Fund	12	1
---	---	---	28	Snyder	M	Ind	10	2
Tien	29	75	23	Tien	C	Ind	100	1
Tien	30	55	24	Tien	M	Ind	60	2

In Day 2, the Dean finally increase a projects. Note the order that they increase the projects in, and how the increase IDs are generated.

At this point the ammount of dollars created and made are shown below:

Faculty#	Name	Dollars-Made	Dollars-Created
0	Snyder	24+30=54	24+35+30=89
1	Tien	50+50=100	0
2	Lin	45+45=90	45+34+45+34=158
3	Marshal	34+34=68	0
4	Wiles	28+35=63	50+28+50=128

Moving on.

Day 3:

Person#	Name	Picked	Working-On	Dollars-Payed	DaysLeft
3	Marshal	16	16	165	0
4	Wiles	23	23	100	1
0	Snyder	24	24	60	0
1	Tien	4	4	42	0
2	Lin	17	17	42	1

Generated or Increased Projects (the first three fields are blank if we are generating a project, the Pay includes the current increase):

Increasor	IncreaseID	PayIncrease	ProjectId	CreatorName	Field	Type	Pay	Days
Marshal	31	35	15	Lin	C	Ind	70	1
Marshal	32	6	7	Lin	C	Secret	34	2
---	---	---	33	Snyder	P	Fund	32	1
---	---	---	34	Snyder	P	Fund	24	2
---	---	---	35	Snyder	P	Fund	16	1
---	---	---	36	Snyder	C	Ind	20	2
Tien	37	85	6	Lin	C	Ind	110	1
Tien	38	65	15	Lin	C	Ind	135	2
Tien	39	18	7	Lin	C	Secret	52	1

Tien and Marshal both increased project number 7.

At this point the ammount of dollars created and made are shown below:

Faculty#	Name	Dollars-Made	Dollars-Created
0	Snyder	54+60=114	89
1	Tien	100+42=142	0+75+55=130
2	Lin	90+42=132	158+80+26+24=288
3	Marshal	68+165=233	0+85+16+18=119
4	Wiles	63+100=163	128+25+5=158

Moving on.

Day 4:

Person#	Name	Picked	Working-On	Dollars-Payed	DaysLeft
4	Wiles	---	23	100	0

0	Snyder	33	33	32	1
1	Tien	15	15	135	1
2	Lin	---	17	42	0
3	Marshal	16	6	110	0

Generated or Increased Projects (the first three fields are blank if we are generating a project, the Pay includes the current increase):

Increasor	IncreaseID	PayIncrease	ProjectId	CreatorName	Field	Type	Pay	Days
---	---	---	40	Wiles	M	Ind	24	1
---	---	---	41	Lin	C	Crazy	2	2
---	---	---	42	Lin	C	Crazy	28	1
---	---	---	43	Lin	C	Secret	20	2
Marshal	44	12	41	Lin	C	Crazy	14	2
Marshal	45	4	42	Lin	C	Crazy	32	1
Marshal	46	30	43	Lin	C	Secret	50	2

Day 5:

Person#	Name	Picked	Working-On	Dollars-Payed	DaysLeft
0	Snyder	---	33	32	0
1	Tien	---	15	135	1
2	Lin	7	7	52	1
3	Marshal	43	43	50	1
4	Wiles	---	42	32	0