

## LOAN PAYMENTS

Write a program that prints out loan payment schedules and balances given loan and salary parameters. The parameters are: a loan amount (in dollars), monthly interest rate (in percent), monthly salary (in dollars), salary proportion allocated for loan payments (as a percentage of salary), and a maximum term length (in number of months). The program will take in several “loan plans” described by these parameters and then output a monthly schedule of balances and payments for each plan. For each month within a plan, there is a beginning balance, an interest applied to the balance, a monthly payment, and a resulting ending balance. The schedule will display these fields starting with month 1 and ending with the last month in the term or the month that causes the ending balance to turn to zero, whichever occurs earlier.

The input file will be named “Amounts.txt” and the first line in that file will be header information which your program should ignore. The file will continue with a list of loan plan data, one loan plan per line. Each line will contain **tab-separated** (`\t`) values, in the following order:

- name of the plan (a string value), for example: Plan One
- loan amount (a double value), for example: 5000.00
- monthly interest rate (a double value, followed by a % character), for example: 1.2%
- monthly salary (a double value), for example: 3600.00
- salary proportion (an int value, followed by a % character), for example: 25%
- maximum term length (an int value), for example: 12

Monthly interest is computed by multiplying the beginning balance for that month by the monthly interest rate. Monthly payments are fixed for each month and this fixed amount is computed by multiplying the salary amount by the specified proportion. There is one possible exception: the last monthly payment may be less than this fixed amount if the beginning balance plus interest for that month is less than the amount. Ending balance for the month is simply the beginning balance plus the interest minus the payment, and is the beginning balance for the next month.

The output file should be named “Payments.txt” and should contain, for each plan, the name of the plan, the monthly payment schedule, and a statement indicating the number of months needed to pay off the loan (the month when the ending balance becomes zero). In the case where the balance does not reach zero but the maximum term length has been reached, a message to that effect should be printed. Refer to the next section for format details in the output file. Note that the names and month numbers are left justified, while all other values are printed right justified.

In case the input file does not exist, output a “File not found” message in the output file.

Normally, when dealing with values that involve monetary amounts, it is more appropriate to use integer components of the amounts (one for dollars and one for cents). For this assignment, however, you may use double values for balance, interest, and payment computations. You must be careful, however, when comparing such values to zero, because of precision loss during computations. For example, instead of the test `(balance == 0.00)` which will almost always be false, use `(balance <= 0.0001)`. Similarly, instead of the test `(balance > 0)`, use `(balance > -0.0001)`. These alternatives are reasonable since precision up to the second decimal place is sufficient for our purposes.

### Sample input and corresponding output:

Plan Name	Loan Amount	Interest	Salary	Proportion	Term Limit
Plan One	5000.00	1.2%	3600.00	25%	12
Big Loan	50000.00	0.5%	5000.00	20%	12
Small Loan	500.00	1%	3000.00	15%	3
Low Salary	5555.00	2%	1000.00	10%	5

```
Programmer: <put your name here>
CS 1044 Summer I 2004 Project 3
```

```
Plan Name: Plan One
```

Month	Initial Balance	Interest	Payment	Ending Balance
1	5000.00	60.00	900.00	4160.00
2	4160.00	49.92	900.00	3309.92
3	3309.92	39.72	900.00	2449.64
4	2449.64	29.40	900.00	1579.03
5	1579.03	18.95	900.00	697.98
6	697.98	8.38	706.36	0.00

```
Loan will be paid off in 6 months.
```

```
Plan Name: Big Loan
```

Month	Initial Balance	Interest	Payment	Ending Balance
1	50000.00	250.00	1000.00	49250.00
2	49250.00	246.25	1000.00	48496.25
3	48496.25	242.48	1000.00	47738.73
4	47738.73	238.69	1000.00	46977.42
5	46977.42	234.89	1000.00	46212.31
6	46212.31	231.06	1000.00	45443.37
7	45443.37	227.22	1000.00	44670.59
8	44670.59	223.35	1000.00	43893.94
9	43893.94	219.47	1000.00	43113.41
10	43113.41	215.57	1000.00	42328.98
11	42328.98	211.64	1000.00	41540.63
12	41540.63	207.70	1000.00	40748.33

```
Salary parameters are insufficient to pay off loan in 12 months.
```

```
Plan Name: Small Loan
```

Month	Initial Balance	Interest	Payment	Ending Balance
1	500.00	5.00	450.00	55.00
2	55.00	0.55	55.55	0.00

```
Loan will be paid off in 2 months.
```

```
Plan Name: Low Salary
```

Month	Initial Balance	Interest	Payment	Ending Balance
1	5555.00	111.10	100.00	5566.10
2	5566.10	111.32	100.00	5577.42
3	5577.42	111.55	100.00	5588.97
4	5588.97	111.78	100.00	5600.75
5	5600.75	112.01	100.00	5612.76

```
Salary parameters are insufficient to pay off loan in 5 months.
```

### Submitting your program:

This program is an exercise in loops and file I/O. It should follow the usual documentation and submission standards: your program must include a header comment, pledge statement, and helpful inline comments explaining key portions of your code. Refer to "Elements of Programming Style" available in the course website for some guidelines. As usual, submissions will be handled and auto-graded through the curator system, but will be hand-graded for compliance and documentation by the GTA.