1E3 Practical 4
5th February 2014

Objectives: To practice While Loops – including sentinel controlled loop and nested loops.

Task#1:
Marks: 2
Summary: a program to determine a monthly repayment plan.
Details: write a C++ program to determine how many months it will take to pay for a stereo system that cost €1,500 under the following credit plan: no down payment, an interest rate of 12% per year (1.0 % per month) and monthly payments of €50. Your program should also print the total amount of interest paid over the life of the loan.

Notes/Hints/Additional Details:
- The monthly payment is used to pay the month's interest on the current debt and whatever is left is used to reduce the remaining debt. So of the first month's €50, €15 is interest (1.0% of €1,500) and the remaining €35 reduces the debt to €1465.00.
- You will need a variable to count the number of loop iterations (repetitions) and hence the number of months. Another variable will accumulate the total interest paid over the life of the loan. You will need other variables too. Be sure they all get appropriate initial values!
- Though it is not required it is recommended that you print out relevant details at every loop iteration – the month number, the remaining balance, the monthly interest, the total interest so far, etc.

Task#2 (Sentinel Controlled Loops):
Marks: 2
Summary: write a program to take weather data as input and then calculate some weather statistics and print them to the screen.
Details: write C++ a program to read a weather station name (one word) followed by a series of temperature readings, terminated by the value -999 and to print station name, and the maximum, minimum and average temperature. (Don't worry about alignment.) Your program should correctly compute max and min for any range of temerpature values.

Task#3 (Advanced Task – Nested Loop):
Mark: 1
Summary: this is an extension to the weather statistics program. Part of this extension involves reading weather data from a file, insterad of asking the user to input them.
Details: extend your program to handle many such temperature records, until "XXX" appears in place of a weather station. An example of the input is as follows:

```
DUBAIRPORT      10.5 6.5 5.0 -3.5 -4.0 7.5 3.0 -999
CORKAIRPORT     12.5 13.0 4.0 7.5 8.0 7.0 14.0 10.5 11.5 -999
BELMULLET       6.5 1.0 3.5 2.0 2.5 1.0 0.0 0.0 -2.5 -1.0 2.0 -1.0 -999
XXX
```

If you had these values in a text file called data.txt (use Smultron to create it), instead of typing the input, you could feed the values to your program by running it as
```
./practical4 < data.txt
```

NB:
Take the time to do the assignment carefully. It is not a race.
Do your own error-checking – don't wait for demosntrators to notice errors in calculations!