# COSC2430: Programming and Data Structures

## Managing a budget using linked lists

### Introduction

In this homework, you will create a C++ program to store items in shopping cart, in order to get as many items as possible without exceeding a given budget.

Each item information include:

- Product name
- Cost

### Input and Output

The input is a single text file. Each file starts with a single number (budget), followed by a list of items and their cost.

### Example of input

15.00 ← Budget
Apples, 1.50
Broccoli, 2.00
Yogurt, 4.00
Chocolate, 3.80
Paper towels, 2.20
Milk, 1.70
Ice cream, 4.50

The total of the list above is 19.70, which is more than the allowed budget. The strategy used in this homework will be to **remove the most expensive item until we are below the required budget again.** In this case, we will remove first "Ice cream", then "Yogurt". Note that this strategy will maximize the number of items, but not necessarily the budget. If two or more items have the same maximum price, remove the last in alphabetical order. As output, we will print the final list **sorted by product name**, and the total number of items and cost.

### Example of output

Apples, 1.50
Broccoli, 2.00
Chocolate, 3.80
Milk, 1.70
Paper towels, 2.20

Number of items: 5
Cost: 11.20

The main C++ program will become the executable to be tested by the TAs. The result should be written on another text file (output file), provided on the command line. The input and output files are specified in the command line, not inside the C++ code.

The general call to the executable (max_items, in this example) is as follows:

```
max_items "A=<file>;C=<file>"
```

Call example with one input file and another output file.

```
max_items "A=a.txt;C=c.out"
```

### Handling wrong input

Your program should execute some basic checks on the input file to verify that all information is valid. Examples of not valid input files include:

- A file that does not include a budget
- Negative budget or costs
- Items with no names

If the file does not include a budget or the budget is negative, **the output file should be empty.**

If one of the items is incorrect (no name, negative cost...), **the item should not be added to the list.** Move to the next item.

### Example of wrong input

15.00
Apples, -1.50 → ignore
Broccoli, 2.00
Yogurt, 4.00
Chocolate, 3.80
Paper towels, 2.20
Milk, 1.70
4.50 → ignore

### Output

Broccoli, 2.00
Chocolate, 3.80
Milk, 1.70
Paper towels, 2.20
Yogurt, 4.00

Number of items: 5
Cost: 13.70

## Requirements

- **It is NOT allowed to use vector classes or other classes provided in the STL.**
- Your C++ code must be clear, indented and commented.
- Each item's cost and the total must be printed with 2 decimal points. The total number of items must be printed as an integer.
- Your program will be tested with GNU C++. Therefore, you are encouraged to work on Unix, or at least make sure that your code compiles and runs successfully on the server.
- You can use other C++ compilers, but the TAs cannot provide support or test your programs with other compilers.
- The output file must contain the result in the format specified, including one empty line between list and totals.

## Testing for grading:

- All the cpp files will be automatically compiled. Make sure you have only 1 file containing main(). If there is an error in compilation it will not be executed. **Programs that do not compile on the server will receive a score of 0.** You are responsible of the content of your submission folder.
- **The name of your submission folder must be hw1.** The folder must not be nested in another folder. Make sure that the TAs have read access to your folder.
- **Submitted files must be limited to headers and source files (.h and .cpp).**
- Your program should not crash, halt unexpectedly, take an unusual amount of time to run (more than 10 seconds) or produce unhandled exceptions.
- Your program will be tested with 10 test cases, going from easy to difficult.

## DEADLINE: February 22nd, 5:00PM – NO LATE SUBMISSIONS

## Grading

- A program not submitted by the deadline or that does not compile is zero points.
- A program that compiles but does not work is worth 10 points.
- A code with no comments will receive a 5 points penalty.
- **Bonus 10 points:** if your node structure and linked list are designed as templates, and you build a class "Item" to use with the template.

Plagiarism and cheating: C++ code is individual: it cannot be shared (other than small fragments used in class or in lab examples). Programs will be compared for plagiarism. If the TAs detect that a portion of your C++ code is highly similar to C++ on the Internet or other student the grade will be decreased at least 50%.