# Cowsay Project

## Programming with BASH

Students: Zefizef Sami, Vasileios Skarleas

**Introduction:**

On the UE course of INF203 under the general idea of learning the basic commands of BASH and having an introduction at the C language programming, we are forced to develop two different projects using BASH (Shell scripting) for the first phase as well as C for the rest of the phases and following the different programming techniques as well as skills that we developed during this semester. Below are written the indications that we needed to follow, the procedure that we followed in order to resolve any problems, as well as a summary of the problems / difficulties that we met. Our goal through this project is to explore smart applications of functions in these two essential programming languages while we are staying focused on the indications given by the project’s paper.

Status:

* Total hours of development: 32 hours
* Total lines of codes: ---

**Indications:**

Phase 1 (Bash)

By using the cowsay library of Bash (to be installed) we need to propose several scripts that perform the following actions mentioned below. In the scripts below, the goal is to use dynamic listing of content, directed to cowsay printing function. We are required to create seven scripts. More specifically:

1. cow kindergarten
2. cow primaryschool
3. cow highschool
4. cow college
5. cow university
6. smart cow
7. crazy cow

cow kindergarten: Pronounces the numbers from 1 to 10.

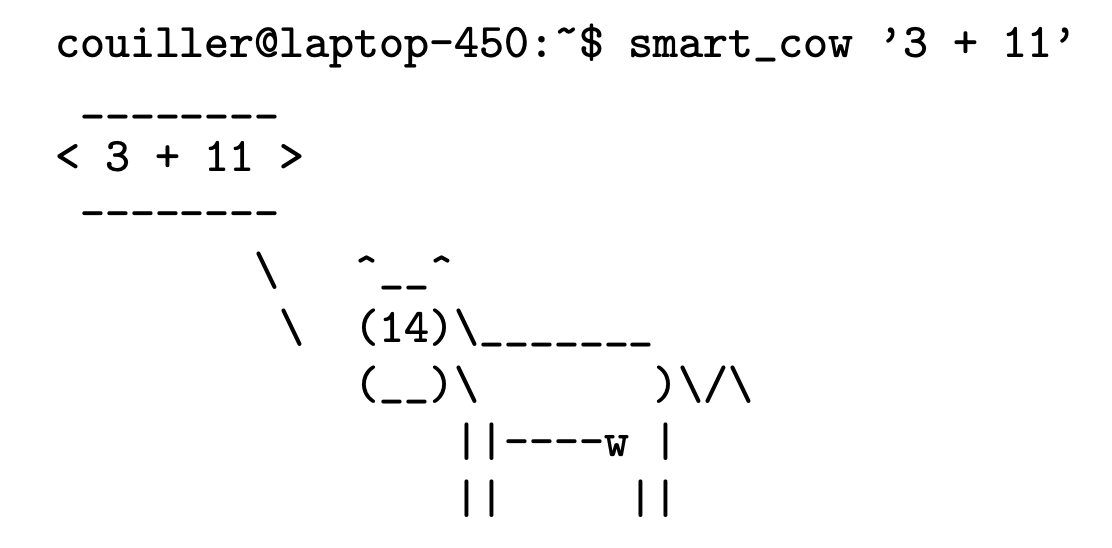
cow primaryschool: Pronounces the numbers from 1 to a number that is specified as argument from the user (ex: ./cow\_primaryschool 15).

cow highschool: Pronounces the number from 1 all the suites of number until reaching the limit x with the condition to pronounce the .

cow college: Pronounces the fibonanchi numbers until a limit set by the user (ex: ./cow\_college 8).

cow university: Pronounces the suite of first numbers until a certain limit set by the user (ex: ./cow\_university 6).

smart cow: By adding a mathematical expression (\*,+,-,/) with only two terms, the program does the operation and gives an output like the following:



crazy cow: On the project is mentioned to leave our imagination free and create a script. We decided to create a feedback program for a website. The cow poses several questions to the user after he is automatically redirected on the website, saves the responses (0 to 5) and according to level of satisfaction, it may ask for further feedback. Last but not least, before it terminates the feedback survey, the program prints a dynamic message according to user’s overall review.

Phase 2 (C)

---

**Procedure:**

As team, we decided to work to the corresponding workflow below:

* Organization, transparency and communication are our standards for a good team work.
* According to the mantra mentioned above, we used the softwares mentioned below:
  1. GitHub: Includes our private repository with all the code that has been submitted by the team members. It tracks the live versions that every team member is watching every time on his personal desktop workstation. Also, GitHub provides a great way to visualize any changes and to restore previous versions of the code.
  2. VSCode: Instead of the direct terminal idle, we decided to work on VSCode since it is a universal development app, with some great extensions to integrate onto the workflow and to visualize better the different commands. In addition to that, it offers a direct connection to our GitHub repository.
  3. W3Schools: Used for classes documentation that we integrated for part c on our separate version (explanation at paragraph Difficulties/Project – section 11).
* From the first view of the project, we knew that the phase C would be the most complicated one and there was a possibility that we couldn’t be able to make it so far. So, our goal was set to complete as better as possible the rest of the tasks.
* For every step on the different two phases mentioned above, we decided to follow the “exams technique”. This means, that we will follow the exact steps by including the required elements mentioned on the specific task (function, returned value, required arguments, etc.) as part of project’s understanding. The rest of the code will get completed according to the personal point of view through the understanding process of the task. This will allow us to have the biggest understanding that is possible, for this demanding project.
* The phase 1 as proposed needs to be completed before the final course of Bash (week before the vacations). We finalized the code for this phase after our return from the vacation.

**Difficulties/Problems:**

In this section we are presenting the different challenges that we met and the workflows around them in order to be solved:

1. The first difficulty the we needed to solve was how the smart cow program will understand the expression that it needs to perform. To solve this issue we divided the cases that we will do a filtration with the cut command. Firstly, we save the operator in a variable and then we compare the operator’s variable with the four math expressions.
2. Something that is interested is that during our TPs when we needed to do a multiplication we couldn’t do that by using $(expr 5 \* 4) for instance. In order to go around this issue, for all the multiplications required to the phase 1, we used this syntax $(( 5 \* 4 )).
3. ---

**Project’s code:**

We are attaching our project’s code divided into the different sections of the project mentioned on this document’s introduction.

Version 1: