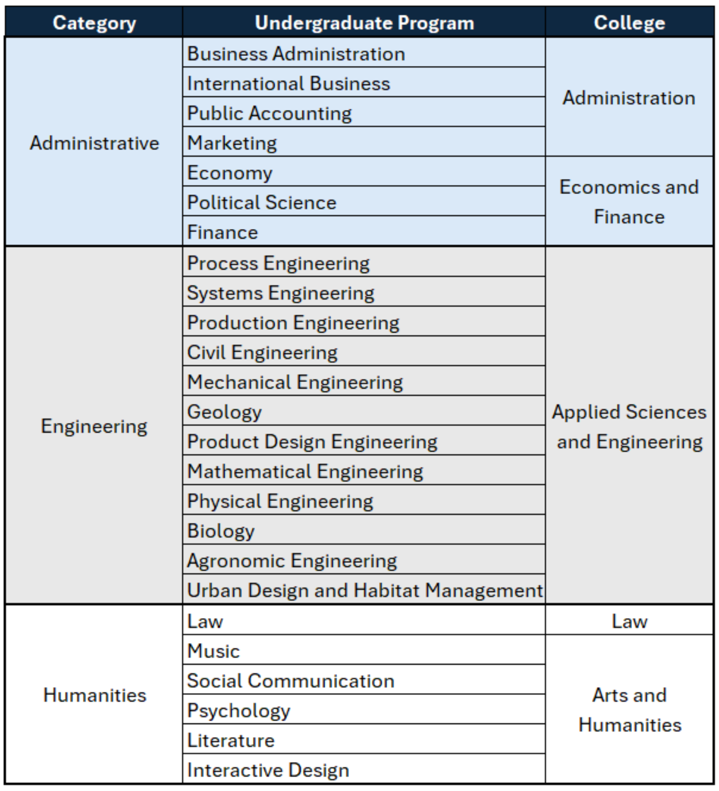
Universidad EAFIT
Systems Engineering
ST0244 Programming Languages
Diego Iván Cruz Ordiéres
6th May 2024
Practice 2 (15%)

Student’s ID card

The employees of the Admissions and Registry office at the Universidad EAFIT need to modify the way in which the enrollment code that is registered in the student's ID card is given to the student who will enter the undergraduate programs as of semester 2024-2, in such a way that it identifies:

- ✓ The period of admission.
- ✓ The type of program the student enters, considering three categories (Administrative, Humanities and Engineering) without giving a specific number to these.
- ✓ A number that reflects the order in which students are admitted to the program they enroll in.
- ✓ The ID code can only be a positive integer numeric value and must contain a maximum of 8 digits.
- ✓ Finally, they would like to know if each ID code is an odd or even numeric value.

Table 1. Undergraduate Academic Programs at Universidad EAFIT



Category	Undergraduate Program	College
Administrative	Business Administration	Administration
	International Business	
	Public Accounting	
	Marketing	
	Economy	Economics and Finance
	Political Science	
	Finance	
Engineering	Process Engineering	Applied Sciences and Engineering
	Systems Engineering	
	Production Engineering	
	Civil Engineering	
	Mechanical Engineering	
	Geology	
	Product Design Engineering	
	Mathematical Engineering	
	Physical Engineering	
	Biology	
	Agronomic Engineering	
	Urban Design and Habitat Management	
Humanities	Law	Law
	Music	Arts and Humanities
	Social Communication	
	Psychology	
	Literature	
	Interactive Design	

Fuente: https://bit.ly/3wdZPKz

Several proposals were received, but one of them caused the admiration of the staff in charge of the process and they decided to adopt it. The proposal consists of composing the student's code with the following elements:

- The first three (3) digits correspond to the income period. Example: 241 for 2024-1, 242 for 2024-2, 251 for 2025-1 and so on
- The next two (2) digits correspond to the category of the academic program and are assigned as follows: random values between 01 and 99 are assigned to define the category (if the value has only one digit, a zero is prefixed to the left). The program category is defined based on the classification scheme proposed by the

Greek mathematician Nicomachus (Polymeris, 2024) for positive integers. The detailed explanation of this method is made after defining the requirements.

- The last three (3) digits correspond to the consecutive number obtained by the student upon admission to the set of programs that are in the category to which the student's academic program belongs, containing values from 001 to 999 (if the value contains less than three digits, the three digits must be completed with zeros to the left). This consecutive number is defined by the Admissions and Registry office.

In example: The 24276002 code consists of three sections:

$$\underbrace{2\,4\,2}_{\text{Academic period}}\overbrace{7\,6}^{\text{Academic program category}}\underbrace{0\,0\,2}_{\text{Consecutive admission number}}$$

Nicomanchus classification scheme:
It consists of classifying a positive integer value as abundant, perfect, or deficient according to its "aliquot sum". The aliquot sum consists of adding the integer divisors of a positive integer without repeating and without including the number itself.

<u>Examples:</u>

- ✓ Aliquot sum of 6: 1 + 2 + 3 = 6. Since the sum of the factors greater than or equal to one (1) and less than (6) results in the same value, six (6) is classified as a perfect number.
- ✓ Aliquot sum of 10: 1 + 2 + 5 = 8. Since the sum of factors is less than the number in question, 10 is classified as a poor number.
- ✓ Aliquot sum of 12: 1 + 2 + 3 + 4 + 6 = 16. The result of the sum of the factors is greater than the number in question; therefore, 12 is classified as an abundant number.

In the striking proposal for the officials, the abundant numbers correspond to the academic programs that are in the administrative category, the perfect numbers correspond to the programs that are in the engineering category, and the deficient numbers correspond to the academic programs that are in the humanities category.

Explanatory note:
Programs are not determined as perfect, abundant, or deficient, it is simply a process that is performed to establish a classification process.

Activity
The university hires you to develop a computer program that can deliver the characteristics of a code entered as an input parameter. The application must be developed using the Haskell programming language. The program must identify the characteristics of each ID card that was produced between semesters 2024-2 and 2026-2. There are four (4) characteristics that the program must output, each characteristic is identified by a word or set of characters without spaces, the spaces are used to separate the characteristics from each other.

To illustrate the expected result, the following examples are included:

Input	Expected Output
24276002	2024-2 Humanities num2 even
25128112	2025-1 Engineering num112 even
25206025	2025-2 Engineering num25 odd
26124236	2026-1 Administratives num236 even
26299115	2026-2 Humanities num115 odd

Delivery

The solution proposed by the team should be implemented in a script using the Haskell programming language using the online ide OnlineGDB. The team must be formed by a minimum and a maximum of two members of the course.

Delivery of the solution is done in two ways:

I. Through the site set up in the OnlineGDB Classroom by clicking on this link. Each member of the team must enter the link and deliver the proposed solution.

II. Include a text document with the names of the two team members and the answers to the following points:

1. Describe in detail the process you carried out to solve the exercise.
2. Write a form in which the currying process is manifested in the proposed solution.
3. For each function developed, indicate whether you identify it as polymorphic, of higher order, or, if you do not consider it to be classified in one or both criteria, justifying each defined determination.
4. Store the developed source file(s) in a GitHub repository, invite the teacher as a collaborator and place a link to access the developed repository at this point in the written document.
5. For the presentation of the developed solution, the team must elaborate a video in which each member appears detailing step by step the process that was carried out, all team members must appear in the video and participate. The video must be uploaded to the Internet and a link must be generated so that it can be accessed for the corresponding review of the delivery. The duration of the video should not exceed seven (7) minutes. This link must be included in this point of the document.

Valuation:

- Solution of the exercise using the Haskell programming language. (50%)
- Support video. (50%)

Considerations:

- The practice can be done in teams of up to two (2) people.
- There is a section in “Evaluaciones / Buzón” In “Interactva” platform for the delivery of the practice. The delivery is done both through the space enabled for delivery and through a link to a repository on GitHub. For this reason, the developed exercise must be uploaded to GitHub and the link to access the proposed solution must be shared through the delivered document.
- The delivered solution must be based on **functional programming**.

- In the GitHub repository, a "Readme" file must be included containing the full names of the members, the version of the compiler they selected, the IDE they used for the development of the practice and the link to the support video.

References

Polymeris, L. (2024). *Perfect numbers*. Exercism: https://exercism.org/tracks/haskell/exercises/perfect-numbers