

## Sequential Structure

1. Write a program that displays the message “Hello, World!” on the screen.
2. Write a program that asks for a number and then shows the message: “The number entered was [number].”
3. Write a program that asks for two numbers and prints their sum.
4. Write a program that asks for four bimonthly grades and shows the average.
5. Write a program that converts meters to centimeters.
6. Write a program that asks for the radius of a circle, then calculates and shows its area.
7. Write a program that calculates the area of a square and then shows double this area to the user.
8. Write a program that asks how much you earn per hour and how many hours you worked in the month. Calculate and show your total monthly salary.
9. Write a program that asks for a temperature in degrees Fahrenheit, converts it, and shows it in degrees Celsius.
  - **Formula:**  $C = 5 * (F - 32) / 9$
10. Write a program that asks for a temperature in degrees Celsius, converts it, and shows it in degrees Fahrenheit.
11. Write a program that asks for 2 integers and 1 real number. Calculate and show:
  - a) the product of double the first plus half of the second;
  - b) the sum of triple the first plus the third;
  - c) the third raised to the cube.
12. Given a person’s height, build an algorithm that calculates their ideal weight:
  - **Formula:**  $(72.7 * \text{height}) - 58$
13. Given a person’s height and sex, build an algorithm that calculates the ideal weight using:
  - **Men:**  $(72.7 * h) - 58$
  - **Women:**  $(62.1 * h) - 44.7$Then ask for the person’s weight and say if they are **within**, **above**, or **below** the ideal weight.
14. João “Fisherman-Talk,” a good man, bought a microcomputer to control his daily income. Every time he brings in fish weighing more than the state regulation of São Paulo (50 kg), he must pay a fine of R\$4.00 per excess kilogram. Write a program that reads the variable peso (fish weight) and checks if there is an excess. If so, store the excess in excesso and the amount of the fine in multa. Otherwise, show both as **ZERO**.
15. Write a program that asks how much you earn per hour and how many hours you worked in the month. Calculate and show the total salary for the month, considering deductions of **11% Income Tax**, **8% INSS**, and **5% Union**. Output:
  - Gross Salary
  - Amount paid to INSS
  - Amount paid to Union
  - Net SalaryDisplay in the format:
16. + Gross Salary : R\$ x
17. - IR (11%) : R\$ y
18. - INSS (8%) : R\$ z
19. - Union (5%) : R\$ w
20. = Net Salary : R\$ n
  - Note:** Gross Salary – Deductions = Net Salary.
21. Paint shop. Ask the area in square meters to be painted. Coverage: **1 liter per 3 m<sup>2</sup>**. Paint is sold in **18L cans** at **R\$80.00**. Inform how many cans are needed (always round up) and the total price.

22. Paint shop (version 2). Ask the area ( $\text{m}^2$ ). Coverage: **1 liter per 6  $\text{m}^2$** . Paint is sold in **18L cans (R\$80.00)** and **3.6L gallons (R\$25.00)**. Add **10% slack** and always round quantities **up**. Inform:
- Buying **only 18L cans**;
  - Buying **only 3.6L gallons**;
  - **Mixed** (cans + gallons) to **minimize price**.
- 

### Decision Structure

1. Ask for two numbers and print the larger.
2. Ask for a value and show if it is **positive** or **negative**.
3. Read a character “F” or “M” and print: **F – Female, M – Male**, or **Invalid Sex**.
4. Ask for a 4-digit year and determine if it is **leap** or **not**.
5. Read a letter and determine whether it is a **vowel** or **consonant**.
6. Read three numbers and display them in **descending** order.
7. Read two partial grades. Compute the **average** and display:
  - “Approved” if average  $\geq 7$ ;
  - “Failed” if average  $< 7$ ;
  - “Approved with Distinction” if average = 10.
8. Read three numbers and show the **largest**.
9. Read three numbers and show the **largest and the smallest**.
10. Ask for the price of three products and inform which to buy (always choose the **cheapest**).
11. Ask which shift you study: **M** (morning), **V** (afternoon), **N** (night). Print **“Good morning!”**, **“Good afternoon!”**, **“Good evening!”**, or **“Invalid value!”**.
12. Salary adjustment by current salary:
  - up to R\$280.00 (inclusive): **+20%**
  - R\$280.00 to R\$700.00: **+15%**
  - R\$700.00 to R\$1500.00: **+10%**
  - above R\$1500.00: **+5%**After the increase, display: previous salary, applied percentage, increase amount, new salary.
13. Payroll calculation. Deductions: **IR** by bracket (below), **3% Union**. **FGTS** is **11%** of gross salary **paid by the company** (not deducted). Net = Gross – Deductions. Ask hourly rate and monthly hours.
  - Gross  $\leq 900$ : **IR exempt**
  - $\leq 1500$ : **5% IR**
  - $\leq 2500$ : **10% IR**
  - 2500: **20% IR**Show like the given example with FGTS, total deductions, and net salary.
14. Read a number 1–7 and print the **day of the week** (1=Sunday, 2=Monday, ...). Others: **invalid value**.
15. Given a **cost price** and an **origin code**, print price with origin. If code is none of the specified, classify as **imported**. Codes: **1 South, 2 North, 3 East, 4 West, 5 or 6 Northeast, 7 or 8 Midwest**.
16. Modify the average program to show:
  - 0 to 3: **Failed**
  - 3 to 6.9: **Exam**
  - 7 to 10: **Approved**

17. Read two partial grades; compute average and assign concept:

- 9.0 to 10.0 → A
- 7.5 to 9.0 → B
- 6.0 to 7.5 → C
- 4.0 to 6.0 → D
- 0 to <4.0 → E

Show grades, average, concept, and message **APPROVED** if A/B/C, else **FAILED**.

18. Read three sides; determine if they can form a triangle. If yes, classify as **equilateral**, **isosceles**, or **scalene**. (*Triangle inequality rules apply.*)

19. Quadratic roots ( $ax^2 + bx + c$ ). Ask a, b, c. Handle:

- $a = 0 \rightarrow$  not quadratic  $\rightarrow$  **end**;
- $\Delta < 0 \rightarrow$  no real roots  $\rightarrow$  **inform and end**;
- $\Delta = 0 \rightarrow$  one real root;
- $\Delta > 0 \rightarrow$  two real roots.

20. Ask for a year and say if it is **leap**.

21. Ask for a date **dd/mm/yyyy** and validate it.

22. Read an integer  $< 1000$  and print the quantity of **hundreds**, **tens**, **units**, handling plural/commas/conjunctions (e.g.,  $326 = 3$  hundreds, 2 tens and 6 units). Test with the suggested values.

23. Read ages of three students and:

- average  $< 25 \rightarrow$  "Young Group"
- $25-40 \rightarrow$  "Adult Group"
- $40 \rightarrow$  "Elderly Group"

Also (variant): read three partial grades and print Approved/Failed/Approved with Distinction like earlier.

24. ATM. Ask withdrawal amount and show the number of banknotes of **100**, **50**, **10**, **5**, **1**. Min = R\$10, Max = R\$600. Ignore note availability. Provide examples 256 and 399 as in the statement.

25. Read an integer and determine if it is **even or odd**. (*Use modulo.*)

26. Read a number and inform if it is **integer or decimal**. (*Use rounding.*)

27. Read two numbers; ask which operation to perform. Show the result and say if the result is:

a) even/odd; b) positive/negative; c) integer/decimal.

28. Crime questionnaire (5 yes/no questions). Classify: **Suspect** (2 "yes"), **Accomplice** (3–4), **Assassin** (5), else **Innocent**.

29. Fuel station discounts:

- **Alcohol (A)**: up to 20 L  $\rightarrow$  3%/L discount; above 20 L  $\rightarrow$  5%/L
- **Gasoline (G)**: up to 20 L  $\rightarrow$  4%/L; above 20 L  $\rightarrow$  6%/L

Prices: Gasoline R\$2.50/L; Alcohol R\$1.90/L. Read liters and type; compute total due.

30. Fruit shop price table and extra 10% discount if **> 8 kg** of fruits **or** total **> R\$25.00**. Read kg of strawberries and apples; output final price.

31. **Hipermercado Tabajara** meat promo:

- **File Duplo**: up to 5 kg  $\rightarrow$  R\$4.90/kg; above 5 kg  $\rightarrow$  R\$5.80/kg
- **Alcatra**: up to 5 kg  $\rightarrow$  R\$5.90/kg; above 5 kg  $\rightarrow$  R\$6.80/kg
- **Picanha**: up to 5 kg  $\rightarrow$  R\$6.90/kg; above 5 kg  $\rightarrow$  R\$7.80/kg

One meat type per customer; any quantity. If paid with **Tabajara Card**, extra **5% discount**. Print a receipt with meat type, quantity, total price, payment type, discount, and amount to pay.

---

## Repetition Structure

1. Ask for a grade (0–10). If invalid, show a message and keep asking until a valid value is entered.
2. Read username and password; **do not** accept a password equal to the username. Keep asking.
3. Validate:
  - Name: > 3 characters
  - Age: 0–150
  - Salary: > 0
  - Sex: 'f' or 'm'
  - Marital Status: 's', 'c', 'v', 'd'
4. Country A: population 80,000, growth 3%/year. Country B: 200,000, growth 1.5%/year. Calculate years for A to **reach or surpass** B.
5. Modify the previous program to let the user input populations and growth rates. Validate input and allow repeating.
6. Print numbers 1 to 20, one below the other. Then modify to print them **side by side**.
7. Print only **odd** numbers between 1 and 50.
8. Read two integers and generate all integers in the **inclusive interval** between them.
9. Modify the previous to show the **sum** at the end.
10. Multiplication table generator: for a chosen integer 1–10, print  $n \times 1$  to  $n \times 10$ .
11. Ask for base and exponent; compute **power** without using the language's power function.
12. Ask for 10 integers; show how many are **even** and how many **odd**.
13. Fibonacci series 1, 1, 2, 3, 5, 8, ... up to the **n-th term**.
14. Fibonacci series 0, 1, 1, 2, 3, ... until the value is **greater than 500**.
15. Factorial of a user-provided integer (e.g.,  $5! = 120$ ).
16. Show the series  $S = 1/1 + 2/3 + 3/5 + 4/7 + \dots + n/m$  and print the **sum**.
17.  $H = 1 + 1/2 + 1/3 + \dots + 1/N$ ; compute **H** with **N** terms.
18. **Do not do:**  $S = 1/N + 2/(N-1) + 3/(N-2) + \dots + (N-1)/2 + N/1$ .
19. Show and compute  $S = 1/1 + 3/2 + 5/3 + \dots + 99/50$ .
20. Show and compute  $S = 2(1)/50 + 2(2)/49 + \dots + 2(50)/1$ .
21. Show and compute  $S = (37*38)/1 + (36*37)/2 + \dots + (1*2)/37$ .
22. Show and compute  $S = 1/1 - 2/4 + 3/9 + 4/16 + \dots - 10/100$ .
23. Show and compute  $S = 1/3 - 3/5 + 5/7 - 7/9 + 9/11 - \dots$  with **N** terms.
24. Given a set of **N** numbers, determine the **min**, **max**, and **sum**.
25. Modify the previous to **accept only 0–65536**.
26. Modify the factorial program to allow multiple computations; limit to positive integers < **16**.
27. Ask for an integer and determine if it is **prime**.
28. Modify the prime program to, if not prime, show which numbers it is **divisible by**.
29. Show **all primes** between 1 and **N**, and the **number of divisions** performed.
30. Compute the **arithmetic mean** of **N** grades.
31. Ask the age of **n** people; compute the class average and classify as **young** (0–25), **adult** (26–60), or **elderly** (>60).
32. Election with three candidates. Ask total voters, collect votes, and show votes per candidate.
33. Compute the **average number of students per class**. Ask the number of classes and students in each. Classes cannot have **more than 40** students.
34. For a CD collection: ask the number of CDs and the price of each; compute **total invested** and **average per CD**.

35. Store “Quase Dois” price table: print the prices for **1 to 50 items** at **R\$1.99 each**.
36. Bakery price table: given the price of one bread (e.g., R\$0.18), print prices for **1 to 50 breads**.
37. Rudimentary cash register: read product prices until **0**. Show **total**, ask **cash given**, compute **change**. Then loop for the next purchase. Follow the example format.
38. Factorial with formatted output, e.g.,
39. Factorial of: 5
40.  $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$
41. Read 5 numbers and report the **largest**.
42. Read 5 numbers and report the **sum** and **average**.
43. Read an unspecified number of temperatures; report **min**, **max**, and **average**.
44. (Prime again) Ask an integer and determine if it is prime.
45. Generate a **list of primes** between 1 and a user-provided **N**.
46. Flexible multiplication table: read the number, the **start**, and **end** of the range. Validate  $\text{end} \geq \text{start}$ .
47. Gym survey: for each client, read code, height, weight. End when code = 0. At the end, show codes and values of the **tallest**, **shortest**, **heaviest**, **lightest**, plus **average height** and **average weight**.
48. Annual salary increases:
- Hired in 1995 with salary R\$1,000.00.
  - 1996: +1.5% over initial.
  - From 1997 on: the increase percentage **doubles** each year.
- Determine **current** salary. Then modify to accept the **initial salary**.
49. Read ten pairs: (student number, height in cm). Find the **tallest** and **shortest** students and show their numbers and heights.
50. Accident stats in five cities: read city code, number of passenger vehicles (1999), number of accidents with victims (1999). Report:
- highest and lowest accident indices and their cities;
  - average number of vehicles across the five cities;
  - average number of accidents for cities with fewer than **2,000** vehicles.
51. Debt installments table. Given a **debt value**, print a table with **debt value**, **interest**, **number of installments**, and **installment value**. Interest table:
- 1 installment: 0%
  - 3 installments: 10%
  - 6 installments: 15%
  - 9 installments: 20%
  - 12 installments: 25%
- Follow the sample output.
52. Read an indeterminate number of **positive** numbers and count how many fall into: **[0–25]**, **[26–50]**, **[51–75]**, **[76–100]**. Stop when a **negative** number is read.
53. Snack bar menu (code & price). Read item codes and quantities until the order ends. Show **price per item** and **grand total**.
- 100 Hot Dog R\$1.20
  - 101 Plain Bauru R\$1.30
  - 102 Bauru with egg R\$1.50
  - 103 Hamburger R\$1.20
  - 104 Cheeseburger R\$1.30

- 105 Soda R\$1.00
  - 54. Presidential election with 4 candidates. Codes: **1–4** candidates (create a mapping), **5** null, **6** blank. Read votes until **0**. Show:
    - total votes per candidate; total null; total blank;
    - percentage of null and blank over total.
  - 55. Quiz grading system (10 questions). Ask each answer and compare to the **answer key**:
  - 56. 01-A, 02-B, 03-C, 04-D, 05-E,
  - 57. 06-E, 07-D, 08-C, 09-B, 10-A
- Each correct = 1 point. After each student, ask if another will use the system. After all: show **highest** and **lowest** score, **number of students**, **class average**. (Optional: allow the teacher to input the answer key beforehand.)
58. Read a positive integer and display it **reversed** (e.g., 12376489 → 98467321).

## Lists

- Read a vector of **5 integers** and display them.
- Read a vector of **10 real numbers** and display them in **reverse order**.
- Read **4 grades**; display the grades and the average.
- Read a vector of **10 characters**; count how many **consonants** and print the consonants.
- Read **20 integers** into a vector. Store **even** numbers in even, **odd** in odd. Print all three vectors.
- Read four grades for **10 students**; store each student's **average**; print how many averages are  $\geq 7.0$ .
- Read a vector of **5 integers**; display the **sum**, **product**, and the numbers.
- Read **age and height** of **5 people** into separate vectors; print them **in reverse order**.
- Read vector **A** with **10 integers**; compute and show the **sum of squares** of its elements.
- Read two vectors with **10 elements** each; create a third vector with **20 elements** by **interleaving** them.
- Modify to **interleave 3 vectors** of 10 elements each.
- For **30 students**, read ages and heights. Determine how many **older than 13** have height **below the average height**.
- Read the **average monthly temperature** for the year; compute the **annual average** and print all months **above the annual average** with the month **name**.
- Crime questionnaire using a list (same classification as before).
- Read an undetermined number of **grades** until **-1** (do not store -1). Afterward:
  - a) show how many values were read;
  - b) list all values in input order (side by side);
  - c) list all values in reverse order (one per line);
  - d) sum of values;
  - e) average;
  - f) how many values **above** the average;
  - g) how many values **below seven**;
  - h) end with a message.
- Weekly salaries with commissions: each seller earns **\$200/week + 9% of gross sales**. For each seller, compute the salary and count how many fall in the ranges:
  - \$200–299, \$300–399, ..., \$900–999, \$1000+.

**Challenge:** compute the **index** in the list from salary **without** many nested ifs.
- Long jump competition: each athlete has **five jumps**. The result is the **average** of the five values. Read **name** and five distances, then print name, jumps, and average. End when **name** is blank. Follow the sample format.
- “Best player” poll (jersey numbers **1–23**). Read votes until **0**. Ignore invalid numbers with a warning. Afterward show:

- total votes; jersey numbers with votes and their counts; percentage per player; the **best player** with votes and percentage. Results must be ordered by jersey number. Use arrays. Compute percentage via a **function** that receives (playerVotes, totalVotes) and returns the percentage. Also save the full report to a **text file**.
- OS poll: Read values **1–6** (1=Windows XP, 2=Unix, 3=Linux, 4=Netware, 5=Mac OS, 6=Other) until **0**. Store counts in an array. Compute percentages and the **winner**. Follow the exact table format provided.
- “Bonus projection” (Organizations Tabajara): each employee receives **20%** of their December gross salary; **minimum bonus R\$100**. Read salaries until **0**. Afterward show: each salary with its bonus; total employees processed; **total bonus payout**; number who received the **minimum**; **highest** bonus. Follow the example formatting.
- Car consumption comparison: load a list with **5 car models** and another with their **km per liter**. Compute:
  - a) the **most economical** model;
  - b) for **1000 km**, how many liters each consumes and **how much it costs** if gasoline is **R\$2.25/L**. Follow the sample report.
- Mouse repair survey: for each mouse, read **ID** and **defect type**:
  1. needs ball; 2) needs cleaning; 3) needs cable/connector change; 4) broken/unusable.
 End when ID = 0. Print total and a table with **quantities and percentages** per defect.
- Disk usage report (ACME Inc.): Given a file usuarios.txt with usernames (15 chars) and bytes used, generate relatorio.txt with the formatted table showing usage in **MB** and **%**, plus **total** and **average**. Read the file **once** and store in memory as needed. Use a **function** to convert bytes→MB and another for **percentage**.

---

## Functions

- Print:
  - 1
  - 2 2
  - 3 3 3
  - ...
  - n n n ... n
 for a user-provided n. Use a function that receives n.
- Print:
  - 1 1
  - 1 2
  - 1 2 3
  - ...
  - 1 2 3 ... n
 for a user-provided n. Use a function.
- Function with **three arguments** that returns their **sum**.
- Function with **one argument** that returns 'P' if the argument is **positive**, 'N' if **zero or negative**.
- somalmposto(taxalmposto, custo): alters custo to include **sales tax (%)**.
- Convert **24-hour** time to **12-hour** notation. Use at least two functions (conversion and output). Represent AM/PM as 'A' and 'P'. Include a loop to repeat as desired.
- valorPagamento(valor, diasAtraso): for payments **without** delay, charge the **installment value**. With delay, charge **3% fine + 0.1% per day** interest. Keep asking until installment value **0**; then show the **daily report** (quantity and total paid).
- Function to **count digits** of an integer.
- **Reverse number**: function that returns the reverse of an integer (e.g., 127 → 721).
- **Craps game**: implement the dice game rules described (natural, craps, point, 7 loses).

- **Date with month name:** function receives DD/MM/YYYY and returns "D de <monthName> de YYYY". Optionally validate date and return NULL if invalid.
- **Shuffle word:** function receives a string and returns another with **characters shuffled** (all upper or all lower).
- **Draw a frame:** function that draws a rectangle using +, -, and |. Parameters: rows and columns, default min **1**, max **20**; clamp elegantly if out of range.
- **Magic square (3×3):** find and display all 3×3 magic squares using numbers 1–9 such that all rows, columns, and diagonals sum to the same value. Tip: generate combinations and check sums; a flat vector of 1..9 can be simpler than a 3×3 matrix.

---

## Strings

1. Read **two strings** and print each content with its **length**. Also say whether they have the **same length** and if their contents are **equal or different**.
2. Read a **name** and print it **backwards in uppercase**.
3. Read a **name** and print it **vertically** (one letter per line).
4. Modify to print the name in a **staircase** (cumulative) format.
5. Modify to print the staircase **inverted**.
6. Read a **birth date (dd/mm/yyyy)** and print it with the month **spelled out**.
7. Given a phrase, count:
  - a) number of **spaces**;
  - b) number of **vowels** (a, e, i, o, u).
8. **Palindrome:** read a text, ignore spaces and punctuation, and say whether it is a palindrome.
9. **CPF validation:** read a CPF in the format xxx.xxx.xxx-xx and validate both **check digits** and **formatting**.
10. **Number in words:** read an integer up to **99** and print it in words.
11. **Hangman:** read a list of words from a text file and pick one randomly. The player can make **6 mistakes** before losing. Provide prompts like the sample.
12. **Phone number normalize:** read a phone number. If it has **7 digits**, prepend '3'. Accept with or without dash. Show the normalization with and without formatting like the sample.
13. **Jumbled word game:** pick a random word from a text file, **shuffle** its letters, and give the user **six attempts** to guess. Show the final answer and whether the user won/lost.
14. **Leet speak generator:** research common mappings and transform a text into **leet**.

---

## Files

- Read a text file with IP addresses (one per line) and generate another file listing **valid** and **invalid** IPs as shown in the format provided.
- **ACME Inc. disk usage** (same as in Lists): from usuarios.txt, create relatorio.txt with MB and % usage, total, and average. Read input **once**; implement **bytes→MB** and **percentage** as separate functions.

---

## Classes

1. **Ball:** attributes **color**, **circumference**, **material**; methods **changeColor**, **showColor**.
2. **Square:** attribute **side**; methods to **set side**, **get side**, **area**.
3. **Rectangle:** attributes **A**, **B** (or length/width). Methods to **set/get sides**, **area**, **perimeter**. Program: ask room measures, create object, compute **tiles** and **baseboards** needed.
4. **Person:** attributes **name**, **age**, **weight**, **height**. Methods **ageUp**, **gainWeight**, **loseWeight**, **grow**. (*If age < 21, growing adds 0.5 cm per year.*)



5. **CheckingAccount**: attributes **account number**, **holder name**, **balance** (default 0). Methods **changeName**, **deposit**, **withdraw**.
6. **TV**: simulate a TV object; user can set **channel** and **volume**, both constrained to valid ranges.
7. **Virtual Pet** (Tamagotchi): attributes **name**, **hunger**, **health**, **age**; getters/setters. **Mood** is a computed value from hunger and health (no stored attribute).
8. **Monkey**: attributes **name**, **stomach**; methods **eat()**, **seeStomach()**, **digest()**. Create at least two monkeys; feed them and inspect after each meal. Try making a monkey **eat another**—is **cannibal** possible?
9. **Point & Rectangle**: class **Point(x, y)**; class **Rectangle(width, height)** with an **origin vertex** as a Point. Functions to **print a Point** and to **find the rectangle's center** returning a Point. Create a menu to **change values** and **print the center**.
10. **Fuel Pump**: class with attributes **fuelType**, **pricePerLiter**, **fuelAmount**. Methods:
  - **fillByValue(value)** → liters dispensed;
  - **fillByLiter(liters)** → amount to pay;
  - **changePrice**, **changeFuelType**, **changeFuelAmount**.
 Always **update remaining fuel** after filling.
11. **Car**: property **consumption (km/L)** and **fuel in tank** (initially 0). Methods **drive(distance)** reduces fuel; **getFuel()**, **addFuel(liters)**. Example usage provided.
12. **Investment Account**: like bank account but with **interestRate** attribute. Constructor sets **initial balance** and **rate**. Method **addInterest()** adds interest. Program: savings with **R\$1000.00** at **10%**, apply **addInterest()** five times and print final balance.
13. **Employee**: attributes **name (string)** and **salary (float)**; constructor and getters. Write a small test.
14. **Employee+**: add method **increaseSalary(percentage)**. Example provided.
15. **Virtual Pet++**: allow user to specify **food amount** and **play time**; these affect how quickly **hunger** and **boredom** decrease.
16. Add a **“hidden door”** to show the pet's exact attributes via an **unlisted menu option**. Tip: implement **\_\_str\_\_**.
17. **Pet Farm**: manage several pets in a list; each menu option applies to **all** pets (feed all, play with all, listen to all). Give each pet **random initial** hunger/boredom.

---

## Projects

1. **Disk Quota Control (ACME Inc.)**: From **usuarios.txt** (login + bytes used), generate **relatorio.txt** in the exact format provided, including **MB**, **%**, **total**, and **average**. Read file **once**; implement **bytes→MB** and **percentage** as functions. **Extras**:
  - Sort users by **% usage**;
  - Show only the **top n**;
  - Generate **HTML** output;
  - Create a program that **reads home folders** and generates the initial file.
2. **Apache Log Analyzer**: Show which **Google search strings** most bring users to the organization's site.
3. **Squid Log Analyzer — Blocked Sites**: Show which sites are **most blocked** in the organization.