

## QUESTIONS

About Compilers:

1) A compiler is a computer program (or a group of programs) that, from source code written in a machine language, creates a semantically equivalent program but written in another high-level language, object code. Source: ["Compiler" (2022).](<https://paperpile.com/c/nzWb1H/DR4A>)

a) Wrong (X)

b) Right ( )

2) Classically, a compiler translates a program from a textual language easily understood by humans into a machine language specific to a processor and operating system. Source: ["Compiler" (2022).](<https://paperpile.com/c/nzWb1H/m4GU>)

a) Wrong ( )

b) Right (X)

3) A compiler translates a program from a machine language into a textual language easily understood by humans. Source: ["Compiler" (2022).](<https://paperpile.com/c/nzWb1H/VYOM>)

a) Wrong (X)

b) Right ( )

4) Regarding programming languages and compilers, judge the following items. A computational algorithm written in a programming language can be fully executed without generating any output. Source: [Concursos 2022).](<https://paperpile.com/c/nzWb1H/V6sm>)

a) Wrong (X)

b) Right ( )

5) A compiled language is a programming language where the source code is executed directly by the operating system or processor after being translated through a process called compilation, using a computer program called a compiler, into a low-level language such as Assembly or machine code. Source: ["Compiled Language" (2022).](<https://paperpile.com/c/nzWb1H/Aedg>)

a) Wrong ( )

b) Right (X)

6) An interpreted language is a programming language where the source code is executed directly by the operating system or processor after passing through an interpreter, generating an executable file. Source: ["Interpreted Language" (2022).](<https://paperpile.com/c/nzWb1H/kMa1>)

a) Wrong (X)

b) Right ( )

7) An interpreted language is a programming language where the source code is executed by a computer program called an interpreter, which is then executed by the operating system or processor. Examples of interpreted languages include ASP, JavaScript, PHP, and Python. Source: ["Interpreted Language" (2022).](<https://paperpile.com/c/nzWb1H/sKqt>)

a) Wrong ( )

b) Right (X)

8) A compiled language generates an executable file that can be run directly by the operating system. Some languages, like interpreted languages, do not generate executable code. Examples of compiled languages include C, C++, COBOL, Delphi, and Visual Basic. Source: ["Compiled Language" (2022).](<https://paperpile.com/c/nzWb1H/Aedg>)

a) Wrong ( )

b) Right (X)

9) Regarding the characteristics of major programming languages, judge the following item:

> A high-level programming language consists of symbols, words, and rules close to human language, making it easy to write, debug, and maintain. Source: ["Exam Questions" (2022).](<https://paperpile.com/c/nzWb1H/AnsA>)

a) Wrong ( )

b) Right (X)

10. In computer programming, a reserved word is a word that cannot be used as an identifier in certain programming languages because it is reserved for the language's grammar. For example, in Java, a programmer cannot name a variable "For" because it is a keyword reserved for loop construction. It is a "keyword" and, due to its restricted use, also a "reserved word." (In languages like C or Java, the terms are synonymous.) Source: ["Reserved Word" (2022).](<https://paperpile.com/c/nzWb1H/sIJw>)

a) Wrong ( )

b) Right (X)

11. \*Case-sensitive\* is an anglicism referring to a category of typographic analysis in computing. In Portuguese, it means "sensitive to letter case" or "sensitive to uppercase and lowercase." Software is called \*case-sensitive\* or has "\*case-sensitivity\*" if it can analyze a string, evaluate the presence of uppercase and lowercase letters, and behave differently based on that. Source: ["Case-sensitive" (2022).](<https://paperpile.com/c/nzWb1H/hx6H>)

a) Wrong ( )

b) Right (X)

12. Software is called \*non-case-sensitive\* or lacks "\*case-sensitivity\*" if it cannot analyze a string, evaluate the presence of uppercase and lowercase letters, and behave differently based on that. Source: ["Case-sensitive" (2022).](<https://paperpile.com/c/nzWb1H/aBDX>)

a) Wrong (X)

b) Right ( )

## About Libraries

13. In computer science, a library is a collection of subprograms used in software development. Libraries contain helper code and data that provide services to independent programs, enabling code and data sharing in a modular way. Source: ["Library (computing)" (2022).](<https://paperpile.com/c/nzWb1H/bsM2>)

a) Wrong ( )

b) Right (X)

14. Some executables are both standalone programs and libraries, but most libraries are not executables. Source:

a) Wrong ( )

b) Right (X)

15. Most modern operating systems provide libraries that implement the majority of system services, turning into utilities the services that modern applications expect from the OS. Thus, most code used in modern applications is provided by these libraries. Source: ["Library (computing)" (2022).](<https://paperpile.com/c/nzWb1H/bsM2>)

a) Wrong ( )

b) Right (X)

16. Most modern operating systems do not provide libraries to implement the majority of system services. Therefore, most code used in modern applications is not provided by these libraries. Source: ["Library (computing)" (2022).](<https://paperpile.com/c/nzWb1H/bsM2>)

a) Wrong (X)

b) Right ( )

17. When a program is developed using a programming language, it can use a set of pre-written functions by other programmers that already solve specific problems so that programmers don't have to "reinvent the wheel" in every program. What is this set of functions called? Source: ["Library (computing)" (2022).](<https://paperpile.com/c/nzWb1H/bsM2>)

a) Variable ( )

b) String ( )

c) Library (X)

d) Char ( )

e) Integer ( )

## About Variables

18. Choose the Right option about variables:

> In programming, a variable is an object (a position, often in memory) capable of retaining and representing a value or expression. Source: ["Variable (programming)" (2022).](<https://paperpile.com/c/nzWb1H/ZBBR>)

a) Wrong ( )

b) Right (X)

19. Choose the Right option about variables:

> Integer-type variable: stores numbers without decimal places, only the integer part, which can be positive or negative (e.g., 1, 2, 67, 999). Source: ["Variable (programming)" (2022).](<https://paperpile.com/c/nzWb1H/ZBBr>)

a) Wrong ( )

b) Right (X)

20. Integer-type variable: stores characters (e.g., A, B, C, a, b) and symbols like (@, \\$, %). Source: ["Variable (programming)" (2022).](<https://paperpile.com/c/nzWb1H/ZBBr>)

a) Wrong (X)

b) Right ( )

21. Choose the Right option about variables:

> Real-type variable: stores numbers with decimal places, which can be positive, negative, or integers (e.g., 1, 2.3, 5.7, 78). To use decimals, a period must be used instead of a comma. Source: ["Variable (programming)" (2022).](<https://paperpile.com/c/nzWb1H/ZBBr>)

a) Wrong ( )

b) Right (X)

22. Choose the Right option about variables:

> \*Char\* (Character)-type variable: stores a single character, which can be a letter, number, or symbol (depending on the language), following the ASCII standard (e.g., A, B, 7, J). Numbers stored as characters are incompatible with integers or reals. Source: ["Variable (programming)" (2022).](<https://paperpile.com/c/nzWb1H/ZBBr>)

a) Wrong ( )

b) Right (X)

23. Choose the Right option about variables:

> \*String\*-type variable: stores a sequence of characters, such as words, phrases, or any multi-character text (e.g., "Hello world"). Source: ["Variable (programming)" (2022).](<https://paperpile.com/c/nzWb1H/ZBBr>)

a) Wrong ( )

b) Right (X)

24. A variable is a reserved memory space to store a specific type of data, such as a number, character, or symbol. Source: ["Variable (programming)" (2022).](<https://paperpile.com/c/nzWb1H/ZBBr>)

a) Wrong ( )

b) Right (X)

25. A variable has a name to reference its content. Source: ["Variable (programming)" (2022).](<https://paperpile.com/c/nzWb1H/ZBBr>)

a) Wrong ( )

b) Right (X)

26. Choose the Right option about variables. Source: ["Variable (programming)" (2022).](<https://paperpile.com/c/nzWb1H/ZBBr>)

a) ( ) The value of a variable is the set of control list(s) it accesses.

b) ( ) The value of a variable is the content of its alphanumeric character(s).

c) ( ) A variable's identifier is the content of the sequencing cell(s) associated with it.

d) \ (X) The value of a variable is the content of the memory cell(s) associated with it. (Right)

e) ( ) The value of a control variable is the identifier of a loop structure.

About Arrays and Matrices

27. An array (one-dimensional array) is a variable that stores multiple variables of the same type. Source: [De Oliveira Alves (2013a)](<https://paperpile.com/c/nzWb1H/dCxl>)

a) Wrong ( )

b) Right (X)

28. Arrays are single-dimension matrices. Source: [De Oliveira Alves (2013a)](<https://paperpile.com/c/nzWb1H/dCxl>)

a) Wrong ( )

b) Right (X)

29. Arrays are multi-dimensional arrays. Source: [De Oliveira Alves (2013a)](<https://paperpile.com/c/nzWb1H/dCxl>)

a) Wrong (X)

b) Right ( )

30. A matrix is a collection of variables of the same type, accessible under a single name and stored contiguously in memory. Each variable in an array is individualized using indices. Source: [De Oliveira Alves (2013a)](<https://paperpile.com/c/nzWb1H/dCxl>)

a) Wrong ( )

b) Right (X)

About Arithmetic Operators

31. Arithmetic operators are those studied in school: basic functions like addition (+), subtraction (-), multiplication (\*), division (/), and modulus or remainder (%). Source: [Eduardo (2012) and De Oliveira Alves (2013a)](<https://paperpile.com/c/nzWb1H/dCxl+T147>)

a) Wrong ( )

b) Right (X)

32. Arithmetic operators are represented by: (if, then, else)? Source: [Eduardo (2012) and De Oliveira Alves (2013a)](<https://paperpile.com/c/nzWb1H/dCxl+T147>)

a) Wrong (X)

b) Right ( )

#### About Relational Operators

33. Relational operators compare two values and return a boolean (\*true\* or \*false\*). The operators are Equal to (==), Not equal to (!=), Greater than (>), Less than (<), Greater than or equal to (>=), Less than or equal to (<=). Source: [Corob-msft (2022).](<https://paperpile.com/c/nzWb1H/Ab7g>)

a) Wrong (X)

b) Right ( )

34. Relational operators compare two values and return a boolean (\*true\* or \*false\*). Does the following example illustrate the use of these operators? Source: [Corob-msft (2022).](<https://paperpile.com/c/nzWb1H/Ab7g>)

```
```c
```

```
int a = 10, b = 25, c = 50, d = 100;
```

```
...
```

Does the operation `(a == d)` result in TRUE?

a) Wrong (X)

b) Right ( )

35. Does the operation `(a == d)` result in FALSE?

a) Wrong ( )

b) Right (X)

#### Logical Operators

36. The logical AND operator (`&&`) returns true only if all input conditions are true. For example, `(3 > 2)` is a logical expression with a true value, while `(4 < 1)` is false. Analyze the AND (`&&`) in the following C code: Source: [Eduardo (2019)](<https://paperpile.com/c/nzWb1H/yXkf>)

`((3 < 2) && (2 == 2))`: Result is FALSE?

a) Wrong ( )

b) Right (X)

37. Does `((3 < 2) && (2 == 2))` result in TRUE?

a) Wrong (X)

b) Right ( )

38. The logical OR operator ('||') returns false only if all input conditions are false. For example, '(3 > 2)' is true, while '(4 < 1)' is false. Analyze the OR ('||') in the following C code: Source: [Eduardo (2019)](<https://paperpile.com/c/nzWb1H/yXkf>)

'((2 > 1) || (3 < 7))': Result is TRUE?

a) Wrong ( )

b) Right (X)

39. Does '((2 > 1) || (3 < 7))' result in FALSE?

a) Wrong (X)

b) Right ( )

40. The logical NOT operator ('!') inverts the input condition: true becomes false, and vice versa. For example, '(3 > 2)' is true. Does '!(3 > 2)' result in TRUE?

a) Wrong (X)

b) Right ( )

41. Does '!(3 > 2)' result in FALSE?

a) Wrong ( )

b) Right (X)

#### Conditional Structures

42. Decision commands like 'IF', 'THEN', and 'ELSE' are not considered flow control commands because they use arithmetic operators. Source: [Neiva (2022).](<https://paperpile.com/c/nzWb1H/mmnH>)

a) Wrong (X)

b) Right ( )

43. In computer science, indentation (derived from the English word \*indentation\*) refers to the use of whitespace in source code to highlight or define the structure of an algorithm. In most programming languages, indentation improves code readability. However, in languages like Python, Occam, and Haskell, indentation is mandatory to define code block hierarchy. Source: [Neiva (2022).](<https://paperpile.com/c/nzWb1H/mmnH>)

a) Wrong ( )

b) Right (X)

44. While blocks in C, Java, and PHP are delimited by braces, and in Pascal and Fortran by keywords like 'then' and 'endif', Python uses spaces or tabs for visual indentation without explicit block symbols. Source: ["Python/Basic Concepts/Indentation" (2022).](<https://paperpile.com/c/nzWb1H/XK9w>)

a) Wrong ( )

b) Right (X)

45. Indentation does not make code more readable, does not aid maintenance, and does not help find coding errors. Source: [Neitzke (2022).](https://paperpile.com/c/nzWb1H/9HBy)

a) Wrong (X)

b) Right ( )

46. Python requires standardized indentation. In other languages like C/C++ or JavaScript, indentation is optional and used only for visual clarity. Source: ["Python/Basic Concepts/Indentation" (2022).](https://paperpile.com/c/nzWb1H/OXEF)

a) Wrong ( )

b) Right (X)

47. Python does not require standardized indentation. In languages like C/C++ or JavaScript, indentation is necessary due to block delimiters. Source: ["Python/Basic Concepts/Indentation" (2022).](https://paperpile.com/c/nzWb1H/OXEF)

a) Wrong (X)

b) Right ( )

48. A selection structure (conditional expression) is a control flow structure in programming languages that performs different computations based on whether a condition is true or false. Keywords like `if`, `else if`, and `else` are used. Source: ["Selection Structure" (2022).](https://paperpile.com/c/nzWb1H/DkzX)

a) Wrong ( )

b) Right (X)

49. When the interpreter encounters the identifier "If," it expects a boolean condition (e.g., `x > 0`). If true, the "Then" block executes; if false, the "Else" block executes. The "Else" part is optional. Source: ["Selection Structure" (2022).](https://paperpile.com/c/nzWb1H/DkzX)

a) Wrong ( )

b) Right (X)

50. Simple and compound selection. The "if-then (else)" structure is common in programming languages. Its basic structure is:

> If (condition) Then

> (code block)

> Else

> (code block)

> End If

Source: ["Selection Structure" (2022).](https://paperpile.com/c/nzWb1H/DkzX)

a) Wrong ( )



b) Right (X)

51. The "Else If" structure is formed by composing selection structures:

> If (condition) Then

> (code block)

> Else

> If (condition) Then

> (code block)

> End If

> End If

Source: ["Selection Structure" (2022).](<https://paperpile.com/c/nzWb1H/DkzX>)

a) Wrong ( )

b) Right (X)

52. Multiple-choice selection. This structure compares a value to constants and redirects code flow based on the match. Its basic structure is:

> Switch X

> Case V1:

> (code block)

> Case V2:

> (code block)

> Default:

> (code block)

> End Switch

Source: ["Selection Structure" (2022).](<https://paperpile.com/c/nzWb1H/DkzX>)

a) Wrong ( )

b) Right (X)

### Loop Structures

53. Loop structures allow executing the same command(s) multiple times based on a condition or counter. Source: ["Selection Structure" and Zanelato (2018)](<https://paperpile.com/c/nzWb1H/DkzX+qBn1>)

a) Wrong ( )

b) Right (X)

54. Loop structures allow executing the same command(s) only once based on a condition or counter. Source: ["Selection Structure" and Zanelato (2018)](<https://paperpile.com/c/nzWb1H/DkzX+qBn1>)

a) Wrong (X)

b) Right ( )

55. Loops are used to repeat actions for all elements in a data list or until a stop condition is met. Source: ["Selection Structure" and Zanelato (2018)](<https://paperpile.com/c/nzWb1H/DkzX+qBn1>)

a) Wrong ( )

b) Right (X)

56. While Loop: Tests a logical condition at the loop's start. If true, the loop's commands execute. When false, the loop ends. Source: ["Selection Structure" and Zanelato (2018)](<https://paperpile.com/c/nzWb1H/DkzX+qBn1>)

a) Wrong ( )

b) Right (X)

57. [The following illustration shows which loop structure?] Source: Original authorship.



a) ( ) For

b) ( ) Do While

c) \X\ While

58. [The following illustration shows which loop structure?] Source: Original authorship.



a) ( ) For

b) ( ) While

c) \X\ Do While

59. The `Do While` loop functions similarly to `While`, but the code block executes at least once before checking the condition. Source: ["C/C++ Programming - Loop Commands" (2022).](<https://paperpile.com/c/nzWb1H/UhaX>)

a) Wrong ( )

b) Right (X)

About Functions

60. Functions (subroutines) are widely used in programming to avoid code duplication and improve readability. For example, calling a square root function ten times instead of rewriting it. Source: [De Oliveira Alves (2013b)](<https://paperpile.com/c/nzWb1H/vHyo>)

a) Wrong ( )

b) Right (X)

61. Example C Program:

```
```c
int main() {
    return 0;
}
```
```

The `main` function returns an integer. Source: Original authorship.

a) Wrong ( )

b) Right (X)

About Procedures

62. Procedures differ from functions by not returning a result. Functions produce a concrete result usable by other algorithms. Source: [De Oliveira Alves (2013b)](<https://paperpile.com/c/nzWb1H/vHyo>)

a) Wrong ( )

b) Right (X)

About Records

63. A record is a heterogeneous data structure composed of variables (fields) that can hold different data types, including composite types (arrays, matrices, records). Source: ["Struct - Records in C Language" (2017)](<https://paperpile.com/c/nzWb1H/AsMG>)

a) Wrong ( )

b) Right (X)

64. A record is a non-heterogeneous data structure composed of variables (fields) that hold identical data types. Source: ["Struct - Records in C Language" (2017)](<https://paperpile.com/c/nzWb1H/AsMG>)

a) Wrong (X)

b) Right ( )

65. In C, records are called structures using the `struct` keyword. Example:

```
```c
struct student {
    int code;
    char name[200];
    float grade;
}
```

```
};
```

```
struct student special_student, regular_student, listener_student;
```

```
...
```

Source: ["Struct - Records in C Language" (2017)](<https://paperpile.com/c/nzWb1H/AsMG>)

a) Wrong ( )

b) Right (X)

#### About Object-Oriented Programming

66. A class is a user-defined type containing the "recipe" for objects. It associates data and methods, controlling access to them. Source: ["Caelum Technology School Online Courses" (2022).](<https://paperpile.com/c/nzWb1H/rtLx>)

a) Wrong ( )

b) Right (X)

67. Objects are characterized by attributes and methods. Attributes are properties; methods are actions. Objects are instantiated from classes. Source: ["Caelum Technology School Online Courses" (2022).](<https://paperpile.com/c/nzWb1H/rtLx>)

a) Wrong ( )

b) Right (X)

68. Attributes are an object's characteristics (e.g., color, weight, name for a dog). Source: ["Caelum Technology School Online Courses" (2022).](<https://paperpile.com/c/nzWb1H/rtLx>)

a) Wrong ( )

b) Right (X)

69. Methods are actions an object can perform (e.g., bark, run, jump for a dog). Source: ["Caelum Technology School Online Courses" (2022).](<https://paperpile.com/c/nzWb1H/rtLx>)

a) Wrong ( )

b) Right (X)

70. Abstraction focuses on essential aspects of a context, ignoring less important details. In OOP, a class abstracts entities in the software domain. Source: ["Caelum Technology School Online Courses" (2022).](<https://paperpile.com/c/nzWb1H/rtLx>)

a) Wrong ( )

b) Right (X)

71. Generalizing a class via inheritance centralizes common characteristics. Derived classes specialize the base class. Source: ["Caelum Technology School Online Courses" (2022).](<https://paperpile.com/c/nzWb1H/rtLx>)

a) Wrong ( )

b) Right (X)

72. Polymorphism allows methods or objects to take multiple forms. It is enabled through inheritance and polymorphic rules. Source: ["Caelum Technology School Online Courses" (2022).](https://paperpile.com/c/nzWb1H/rtLx)

a) Wrong ( )

b) Right (X)

73. Encapsulation hides code complexity. For example, a driver doesn't need to understand a car's engine internals. Data access is restricted to authorized methods. Source: ["Caelum Technology School Online Courses" (2022).](https://paperpile.com/c/nzWb1H/rtLx)

a) Wrong ( )

b) Right (X)

## REFERENCES

[Library (computing). Available at:](https://pt.wikipedia.org/wiki/Biblioteca\_(computa%C3%A7%C3%A3o)). Accessed: Jan 20, 2022.

[Caelum Technology School Online Courses. Available at:](https://www.caelum.com.br/apostila-java-orientacao-objetos/orientacao-a-objetos-basica). Accessed: Jan 20, 2022.

[Case-sensitive. Available at:](https://pt.wikipedia.org/wiki/Case-sensitive). Accessed: Jan 20, 2022a.

[Case-sensitive. Available at:](https://pt.wikipedia.org/wiki/Case-sensitive). Accessed: Jan 20, 2022b.

[Compiler. Available at:](https://pt.wikipedia.org/wiki/Compilador). Accessed: Jan 20, 2022a.

[Compiler. Available at:](https://pt.wikipedia.org/wiki/Compilador). Accessed: Jan 20, 2022b.

[Compiler. Available at:](https://pt.wikipedia.org/wiki/Compilador). Accessed: Jan 20, 2022c.

[CONCURSOS, T. Question 617938 CEBRASPE (CESPE) - Technical Intelligence Officer. Available at:](https://www.tecconcursos.com.br/questoes/617938). Accessed: Jan 20, 2022.

[COROB-MSFT. C Relational and Equality Operators. Available at:](https://docs.microsoft.com/pt-br/cpp/c-language/c-relational-and-equality-operators). Accessed: Jan 20, 2022.

[EDUARDO. Arithmetic Operators in C. Available at:](http://linguagemc.com.br/operadores-aritmeticos-em-linguagem-c/). Accessed: Jan 20, 2022.

[EDUARDO. Logical Operators in C. Available at:](http://linguagemc.com.br/operadores-logicos-em-c/). Accessed: Jan 20, 2022.

[Selection Structure. Available at:](https://pt.wikipedia.org/wiki/Estrutura\_de\_sele%C3%A7%C3%A3o). Accessed: Jan 20, 2022.

[Compiled Language. Available at:]([https://pt.wikipedia.org/wiki/Linguagem\\_compilada](https://pt.wikipedia.org/wiki/Linguagem_compilada)). Accessed: Jan 20, 2022.

[Interpreted Language. Available at:]([https://pt.wikipedia.org/wiki/Linguagem\\_interpretada](https://pt.wikipedia.org/wiki/Linguagem_interpretada)). Accessed: Jan 20, 2022a.

[Interpreted Language. Available at:]([https://pt.wikipedia.org/wiki/Linguagem\\_interpretada](https://pt.wikipedia.org/wiki/Linguagem_interpretada)). Accessed: Jan 20, 2022b.

[NEITZKE, N. A. Good Programming Practices: Organizing Code and Packages. Available at:](<https://www.devmedia.com.br/boas-praticas-de-programacao/21137>). Accessed: Jan 20, 2022.

[NEIVA. C Language: Conditional Structures. Available at:](<https://br.ccm.net/faq/10200-linguagem-c-estruturas-condicionais>). Accessed: Jan 20, 2022.

[DE OLIVEIRA ALVES, G. F. What Are Arrays and Matrices. Available at:](<https://dicasdeprogramacao.com.br/o-que-sao-vetores-e-matrizes-arrays>). Accessed: Jan 20, 2022a.

[DE OLIVEIRA ALVES, G. F. What Are Functions and Procedures. Available at:](<https://dicasdeprogramacao.com.br/o-que-sao-funcoes-e-procedimentos>). Accessed: Jan 20, 2022b.

[Reserved Word. Available at:]([https://pt.wikipedia.org/wiki/Palavra\\_reservada](https://pt.wikipedia.org/wiki/Palavra_reservada)). Accessed: Jan 20, 2022.

[C/C++ Programming - Loop Commands. Available at:](<https://www.inf.pucrs.br/~pinho/Laprol/ComandosDeRepeticao/Repeticao.html>). Accessed: Jan 20, 2022.

[Python/Basic Concepts/Indentation. Available at:]([https://pt.wikibooks.org/wiki/Python/Conceitos\\_b%C3%A1sicos/Indenta%C3%A7%C3%A3o](https://pt.wikibooks.org/wiki/Python/Conceitos_b%C3%A1sicos/Indenta%C3%A7%C3%A3o)). Accessed: Jan 20, 2022a.

[Python/Basic Concepts/Indentation. Available at:]([https://pt.wikibooks.org/wiki/Python/Conceitos\\_b%C3%A1sicos/Indenta%C3%A7%C3%A3o](https://pt.wikibooks.org/wiki/Python/Conceitos_b%C3%A1sicos/Indenta%C3%A7%C3%A3o)). Accessed: Jan 20, 2022b.

[Exam Questions. Available at:](<https://www.qconcursos.com/questoes-de-concursos/questoes/026552a2-59>). Accessed: Jan 20, 2022.

[Struct - Records in C Language. Available at:](<https://www.embarcados.com.br/struct-registros-em-linguagem-c/>). Accessed: Jan 20, 2022.

[Variable (programming). Available at:]([https://pt.wikipedia.org/wiki/Vari%C3%A1vel\\_\(programa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Vari%C3%A1vel_(programa%C3%A7%C3%A3o))). Accessed: Jan 20, 2022.

[ZANELATO, J. Programming Logic - Loop Structures. Available at:](<https://podprogramar.com.br/logica-de-programacao-estruturas-de-repeticao/>). Accessed: Jan 20, 2022.

Note: The original images (e.g., `media/image1.png`) and hyperlinks are retained as placeholders. Ensure images are properly embedded in the final document.

