**Respostas da 1ª lista: linguagem de programação.**

**1.**

**- Linguagem de alto nível**: são linguagens, cada vez mais, afastadas da L.M., através do uso de abstrações cada vez mais complexas. Estas abstrações se dão em função da adoção de tipos de dados, palavras reservadas, funções, procedimentos automáticos, etc. exemplos: COBOL, Pascal, Java, Visual Basic, etc.

**- Linguagem de baixo nível** é uma linguagem mais próxima da linguagem de Máquina (L.M.), ou seja, mais próxima do hardware. Exemplo: Assembler.

O compilador tem o papel de traduzir nos programas o código fonte de uma [linguagem de programação de alto nível](https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o_de_alto_n%C3%ADvel) para uma [linguagem de programação de baixo nível](https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o_de_baixo_n%C3%ADvel). E ele é de suma importancia para traduzir um programa de uma linguagem textual facilmente entendida por um ser humano para uma linguagem de máquina, específica para um processador e sistema operacional.

**5.**

**(A).** **Imperativas**: é um paradigma de programação que descreve a computação como ações (comandos) que mudam o estado (variáveis) de um programa. Muito parecido com o comportamento imperativo das linguagens naturais que expressam ordens, programas imperativos são uma sequência de comandos para o computador executar.

Ex. Ada, ALGOL, Basic, C, Fortran, etc.

**(B).** **Funcionais:** Programação funcional baseia-se no conceito de função, pode-se pensar na programação funcional como simplesmente avaliação de expressões e funções.

Ex. Haskell, ML, LISP, etc.

**(C).** **Lógicas:** LPs lógicas consistem na definição de relações lógicas que devem ser satisfeitas pela solução procurada, é um tipo de programação a qual utiliza uso de lógica matemática em sua essência.

Ex. PROLOG.

**(D).** **Marcação:** é um conjunto de códigos aplicados a um [texto](https://pt.wikipedia.org/wiki/Texto) ou a [dados](https://pt.wikipedia.org/wiki/Dado), com o fim de adicionar informações particulares sobre esse texto ou dado, ou sobre trechos específicos.

Ex. HTML.

**Híbrida:** Todas as linguagens são híbridas, quanto mais cruzamentos se processarem dentro de uma mesma linguagem, mais híbrida a linguagem será.

Ex. Linguagem verbal e não-verbal.

**7.** Paradigmas diferem nos conceitos e abstrações utilizadas para representar os elementos de um programa (como objetos, funções, variáveis, restrições, etc) e as etapas que compõem um cálculo (atribuição, avaliação, continuações, fluxos de dados,etc.). Mais claramente falando é a forma de estrutura que você vai usar para programar e executar o seu software.

**9.** É a terceira fase da [compilação](https://pt.wikipedia.org/wiki/Compiladores) onde se verificam os erros semânticos no [código fonte](https://pt.wikipedia.org/wiki/C%C3%B3digo_fonte) e coletam-se as informações necessárias para a próxima fase da compilação, que é a [geração](https://pt.wikipedia.org/wiki/Gerador_de_c%C3%B3digo) de [código objeto](https://pt.wikipedia.org/wiki/C%C3%B3digo_objeto). A análise semântica trata a entrada sintática e transforma-a numa representação mais simples e mais adaptada a geração de código.

**Semântica operacional:** Descreve o significado de um programa pela execução de suas instruções em uma máquina, seja ela real ou simulada, as mudanças no estado da máquina (memória, registradores, etc.) definem o significado da instrução.

**Semântica axiomática**: Baseada em lógica (cálculo de predicados), propósito original (Verificação formal de programas). Axiomas ou regras de inferência são definidas para cada tipo de instrução na linguagem (para permitir transformações de expressões para outras expressões).

**Semântica denotacional**: Define uma função que mapeia instâncias das entidades da linguagem em instâncias dos correspondentes objetos matemáticos.

10.

**Programa – A**

x = 122 \* y – 144 {x > 144}

122 \* y – 144 > 144

y > 2

**É a pré-condição mais fraca.**

**Programa – B**

y = 5 \* x - 5

x = y + 5

{x < 45}

y + 5 < 45

y < 40

**É a pré-condição mais fraca.**

**Programa – C**

if(x < 200)

y = y + 2;

else

y = y – 2;

{y > 2}

y – 2 > 2

y > 4, y > 0

**É a pré-condição mais fraca.**