

PROVA AV1 - Linguagens Formais e Automatos - Seg

izaell.official@uni9.edu.br [Mudar de conta](#)



* Indica uma pergunta obrigatória

Perguntas de Multipla escolha

*

Expressões regulares (regex) são padrões de texto utilizados para buscar e validar padrões em strings. No contexto de programação em Java, expressões regulares são comumente aplicadas para validar e extrair informações de emails.

Considerando o uso de expressões regulares em Java para validar um email, escolha a alternativa que corresponde a uma regex correta para validar um email no formato padrão.

- ☐ "[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}"
- ☐ "\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b"
- ☐ "^((https?:/)?([A-Za-z]{3}\.))?[A-Za-z0-9]+\.[A-Za-z]{3,4}(\.[A-Za-z]{2})?\$"
- ☒ "[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}"
- ☐ "\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\b"



Analise as definições e coloque na ordem correspondente ao conceito.

*

I - É um caractere ou símbolo do alfabeto que é consumido pelo autômato durante o processo de computação.

Função de Transição - Define como o autômato muda de estado com base no símbolo de entrada atual e no estado atual.

III-É um ou mais estados em que o autômato termina sua computação, indicando que a entrada foi aceita de acordo com as regras definidas.

IV-Representa uma condição ou configuração específica do autômato em um determinado momento durante a computação.

V-Refere-se à mudança de estado que ocorre quando o autômato processa um símbolo de entrada.

- ☐ a) Símbolo de Entrada, Função de Transição, Estado de Aceitação, Estado e Transição
- ☐ e) Símbolo de Entrada, Função de Transição, Transição, Estado e Estado de Aceitação
- ☐ c) Estado, Símbolo de Entrada, Função de Transição , Estado de Aceitação e Transição
- ☒ d) Símbolo de Entrada, Função de Transição, Estado, Estado de Aceitação e Transição
- ☐ b) Estado de Aceitação, Símbolo de Entrada, Função de Transição , Estado e Transição

Leia o comando abaixo:

*

input : 'abababaa_abaaabab_ababaabb_abababaa_abaaabab'

blank: ''

start state: Ponto:

table:

Ponto:

[a,b,'_'] : R

[' '] : {R: Final}

Final:

Analise as asserções abaixo, escolha as alternativas corretas.

I-um input inválido seria '010 '

II-um input inválido seria 'aba_'

III-o estado inicial desse automato é Ponto

IV- o estado final desse automato é End

V- ao ler '' vai para o End

☐ b) I e V

☐ e) somente I

☒ a) I, III

☐ d) I e II

☐ c) I e IV



Defina a ordem correta das asserções:

*

I- Linguagens especializadas, como SQL, voltadas para tarefas específicas, como consultas em bancos de dados e voltadas para a geração automática de código.

II- Linguagens especializadas em descrever base de conhecimento, informação e dados.

III- Linguagens especializadas que utiliza mnemônicos para representar instruções de máquina

IV- Linguagens de programação de alto nível, mais próximas da linguagem humana.

V- Linguagem de máquina, compreendida diretamente pelo hardware do computador.

- ☐ Quarto nível, Quinto nível, Segundo nível, Terceiro nível e Primeiro nível.
- ☒ Quinto nível, Quarto nível, Terceiro nível, Segundo nível e Primeiro nível.
- ☐ Primeiro nível, Quarto nível, Quinto nível, Segundo nível e Terceiro nível.
- ☐ Terceiro nível, Quinto nível, Segundo nível, Segundo nível e Primeiro nível.
- ☐ Segundo nível, Terceiro nível, Quarto nível e Quinta nível.

Analise o código abaixo, trata-se da :

*

```
section .data
    hello db 'Hello, World!', 0
```

```
section .text
    global _start
```

```
_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, hello
    mov edx, 13
    int 0x80
```

```
    mov eax, 1
    xor ebx, ebx
    int 0x80
```

O código apresentado é uma implementação em Assembly x86, que exibe a mensagem "Hello, World!" no console. É importante compreender os elementos utilizados nesse código e sua função dentro do contexto da linguagem Assembly.

Com base na análise do código fornecido, selecione a alternativa que descreve corretamente a finalidade do código Assembly apresentado.

- ☒ Uma rotina que imprime a mensagem "Hello, World!" no console.
- ☐ Um script que executa uma série de cálculos matemáticos complexos.
- ☐ Um programa que lê dados de um arquivo de texto e os exibe no console.
- ☐ Uma função que realiza operações aritméticas simples em números inteiros.
- ☐ Um código que solicita entrada do usuário e imprime na tela a mensagem correspondente.

Analise o texto abaixo:

*

"AbcDatalog é uma implementação de código aberto da linguagem de programação lógica Datalog escrita em Java. Ele fornece implementações prontas para uso de algoritmos comuns de avaliação do Datalog, bem como alguns mecanismos experimentais de avaliação multithread. Ele suporta recursos de linguagem além do Datalog principal, como (des)unificação explícita de termos e negação estratificada. Além disso, o AbcDatalog foi projetado para ser facilmente extensível com novos mecanismos de avaliação e novos recursos de linguagem. Esperamos que seja útil tanto para a pesquisa quanto para a pedagogia." Fonte: <https://harvardpl.github.io/AbcDatalog/>

Considerando o texto apresentado, assinale a(s) assertiva(s) verdadeira.

Asserção I - linguagem de quinto nível

Asserção II - trata-se de programação lógica declarativa

Asserção III - trata-se de POO por conta da terceira geração

Asserção IV - trata-se da linguagem de quarto nível

- ☐ I e II
- ☐ IV somente
- ☒ II somente
- ☐ I e III
- ☐ III somente



Ao ler a máquina de turing escrito em C abaixo, responda sobre este código : *

```
#include <stdio.h>

int main() {
    int estado = 0;
    char fita[10];
    int cabecote = 0;

    printf("Insira a fita (0s e 1s): ");
    scanf("%s", fita);

    while (1) {
        switch (estado) {
            case 0:
                if (fita[cabecote] == '0') { estado = 1; cabecote++;
                } else if (fita[cabecote] == '1') { estado = 2; cabecote++;
                } else { estado = -1; } // Estado de rejeição
                break;
            case 1:
                if (fita[cabecote] == '0') { cabecote++;}
                else if (fita[cabecote] == '1') { estado = -1; } // Estado de rejeição
                else { estado = 0; }
                break;
            case 2:
                if (fita[cabecote] == '1') { cabecote++; }
                else if (fita[cabecote] == '0') {
                    estado = -1; } // Estado de rejeição
                else { estado = 0; }
                break;
        }

        if (fita[cabecote] == '\0') { break; }
    }

    if (estado == 0)
        printf("A fita foi aceita pela máquina de Turing.\n");
    else
        printf("A fita foi rejeitada pela máquina de Turing.\n");

    return 0;
}
```



}

Asserção I - A linguagem trata-se da segunda geração

Asserção II - A linguagem é C com palavras reservadas

Asserção III - Este programa trata-se da linguagem de quinta geração

Asserção IV - Este programa trata-se de um bash

Asserção V - Este programa trata-se da linguagem de terceira geração

- ☐ I e III
- ☐ I, II e III
- ☐ I e IV
- ☒ I, II e V
- ☐ I e II

A evolução das linguagens de programação ao longo do tempo tem sido marcada por diferentes gerações, cada uma trazendo novas características e abordagens para o desenvolvimento de software. No contexto da quarta geração de linguagens de programação, destacam-se atributos específicos que diferenciam esse modelo dos anteriores. *

Considerando o contexto apresentado, assinale a alternativa que descreve uma característica fundamental do modelo de quarta geração de linguagens de programação.

- ☐ A restrição à utilização de bibliotecas de terceiros.
- ☐ A dependência de compiladores para gerar código de máquina.
- ☐ A priorização da eficiência de execução em detrimento da legibilidade do código.
- ☐ A ênfase na programação orientada a objetos.
- ☒ A capacidade de criar pipelines para automatizar processos complexos.

Leia o código de terceiro nível abaixo:

*

```
class Aluno{
    Aluno(nomex,notax){ this.nome=nomex; this.nota=notax;}
    String nome;
    float nota;
}
```

Defina qual dos itens abaixo corresponde a vantagem da linguagem acima.

- ☐ São voltadas para a geração automática de código, reduzindo significativamente o tempo necessário para desenvolver sistemas complexos.
- ☐ Mais legível que linguagem de máquina, usa mnemônicos e rótulos que facilitam a compreensão das instruções.
- ☒ São mais próximas da linguagem humana, facilitando o desenvolvimento e a manutenção de software.
- ☐ Cada instrução é executada diretamente pelo hardware, tornando os programas muito rápidos.
- ☐ Embora ainda dependa da arquitetura do processador, é mais portátil que a linguagem de máquina.

[Voltar](#)

[Próxima](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este formulário foi criado em Uninove. [Denunciar abuso](#)

Google Formulários

