lista de exercícios (gigi)

aluno:arthur melquiades dantas de sousa

link github:

data de entrega:08/11/23

1.Questão

passo 1 pega o azul e coloca no H3 passo 2 pega o verde e coloca no H2 passo 3 pegar o azul e coloca em cima do verde no H2 passo 4 pegar o amarelo e coloca no H3 passo 5 colocar o azul em cima do vermelho no H1 passo 6 pega o verde e coloca em cima do amarelo no H3 passo 7 pega o azul no H1 e coloca em cima do verde no H3 passo 8 você pegar o vermelho e coloca no H2 passo 9 pega o azul e coloca no H2 em cima do vermelho e pega o verde e coloca no H1 passo 10 você pega o azul de novo e coloca ele em cima do verde e pega o amarelo e coloca em cima do vermelho no H2 passo 11 pega o azul que estava no H1 e coloca no H3 passo 13 pega o verde e coloca no H2 junto com o vermelho e amarelo e passo 14 por último vc pega o azul e coloca em cima de todos no H2

2.Questão

1. O número mínimo de movimentos necessários para resolver a Torre de Hanói com 5 e 15
2. A relação matemática para encontrar o número mínimo de movimentos para um número n qualquer de discos é 2^n - 1.

3.Questão

Para equilibrar a garrafa, será necessário usar 2 copos.

4.Questão

A mulher de Carlos estava sentada entre Basílio e Armando.

5.Questão

Para rotular os potes corretamente, você precisaria tirar apenas um item de um dos potes rotulados como "Ambas". Se você tirar uma maçã ou uma laranja, você saberá qual é o conteúdo desse pote e, consequentemente, poderá corrigir a rotulagem dos outros dois potes.

6.Questão

Ted é um cavaleiro e Ben e Lil são patifes.

7.Questão

A)Algoritmos são conjuntos de instruções sequenciais que resolvem um problema específico. Suas partes são entrada, processamento e saída. A entrada é a informação fornecida ao algoritmo, o processamento é a manipulação dessa informação e a saída é o resultado produzido pelo algoritmo.

B) As formas de representar algoritmos são: textual (em linguagem natural), em fluxograma e em pseudocódigo.

Vantagens da representação textual:

- Fácil compreensão e familiaridade para pessoas que estão acostumadas com a linguagem natural.

- Flexibilidade na descrição dos passos do algoritmo.

Desvantagem da representação textual:

- Maior propensão a erros de sintaxe, especialmente em algoritmos mais complexos.

Vantagens do fluxograma:

- Visualização clara do fluxo de controle e das decisões tomadas pelo algoritmo.

- Facilidade em identificar loops e estruturas condicionais.

Desvantagem do fluxograma:

- Complexidade em algoritmos mais extensos, com muitos passos e decisões.

Vantagens do pseudocódigo:

- Combinação de linguagem natural com elementos estruturados.

- Facilidade em traduzir o pseudocódigo para uma linguagem de programação real.

Desvantagem do pseudocódigo:

- Não é uma linguagem de programação real, então não pode ser executada diretamente.

C)Texto:

1. Leia a temperatura em Celsius.

2. Converta a temperatura para Fahrenheit usando a fórmula F = (C \* 9/5) + 32.

3. Se F > 100, exiba "Você está com febre".

4. Caso contrário, exiba "Relaxa, tá tudo bem".

Fluxograma:

[Imagem de um fluxograma representando o algoritmo]

Pseudocódigo:

Leia temperatura Celsius

temperatura Fahrenheit = (temperatura Celsius \* 9/5) + 32

Se temperatura Fahrenheit > 100 então

Exiba "Você está com febre"

Senão

Exiba "Relaxa, tá tudo bem"

Fim Se

Observação: Como sou uma amiga e não um ambiente de programação, não consigo exibir imagens ou executar códigos em pseudocódigo.

8.Questão

Linguagem Natural: É a linguagem utilizada pelos seres humanos para se comunicarem, como o português, inglês, etc. Ela é compreendida e utilizada de forma natural pelos humanos, mas não é diretamente compreendida pelas máquinas.

Linguagem de Máquina: É a linguagem utilizada pelos computadores para executar instruções. Consiste em uma sequência de instruções binárias (0s e 1s) que são interpretadas diretamente pelo hardware do computador.

Linguagem de Programação: É uma linguagem criada para ser compreendida tanto por humanos quanto por máquinas. Ela permite que os programadores escrevam algoritmos e instruções de forma mais legível e estruturada, utilizando palavras e símbolos específicos. Exemplos incluem C++, Python, Java, entre outras.

Compilador: É um programa que traduz o código fonte escrito em uma linguagem de programação para código de máquina executável. Ele realiza essa tradução de uma vez só, gerando um arquivo executável independente do código fonte original.

Interpretador: É um programa que lê e executa o código fonte linha por linha, sem a necessidade de compilar previamente. Ele interpreta cada instrução conforme é encontrada, convertendo-a em tempo real para código executável.

9.Questão

Linguagens de baixo nível são linguagens de programação que estão mais próximas da linguagem de máquina e do hardware do computador. Elas são mais complexas e difíceis de serem entendidas pelos humanos, mas oferecem um alto nível de controle sobre o sistema. Exemplos incluem Assembly e linguagem de máquina.

Exemplo de código em linguagem Assembly:

MOV AX, 5

MOV BX, 10

ADD AX, BX

Linguagens de alto nível são linguagens de programação que estão mais distantes da linguagem de máquina e do hardware do computador. Elas são mais fáceis de serem entendidas pelos humanos, pois utilizam uma sintaxe mais próxima da linguagem natural. Essas linguagens são traduzidas para a linguagem de máquina por meio de compiladores ou interpretadores. Exemplos incluem Python, C++, Java.

Exemplo de código em Python:

python

a = 5

b = 10

c = a + b

10.Questão

java

public class Multiplicacao {

public static void main(String[] args) {

int x = 2;

int y = 6;

int resultado = x \* y;

System.out.println(resultado);

}

}

Algoritmo em C#:

csharp

using System;

class Program {

static void Main(string[] args) {

int x = 2;

int y = 6;

int resultado = x \* y;

Console.WriteLine(resultado);

}

}

A diferença entre as linguagens de programação está relacionada à sua sintaxe, semântica e propósito. Cada linguagem foi projetada para atender a diferentes necessidades e oferecer diferentes recursos e funcionalidades. Essas diferenças podem incluir a forma como as variáveis são declaradas, a sintaxe utilizada para realizar operações matemáticas, a maneira como os dados são manipulados, entre outros aspectos. As linguagens de programação são criadas com o objetivo de fornecer uma abstração mais amigável para os programadores, facilitando o desenvolvimento de software de forma eficiente e legível.

11.Questão

Os tipos de instruções primitivas comuns às linguagens de programação incluem:

1. Atribuição: Permite atribuir um valor a uma variável.

2. Expressões: São combinações de operadores e operandos que realizam cálculos ou operações.

3. Estruturas de controle: Incluem estruturas condicionais (if, else, switch) e estruturas de repetição (for, while, do-while) que controlam o fluxo de execução do programa.

4. Entrada/Saída: Permitem interagir com o usuário ou com o ambiente externo, lendo dados do teclado ou escrevendo dados em tela ou em arquivos.

No Python, todas essas instruções primitivas estão presentes. No entanto, em comparação com outras linguagens de programação, como C e Java, o Python não possui uma estrutura de controle switch. Em vez disso, utiliza-se uma combinação de declarações if-elif-else para realizar múltiplas verificações condicionais. Isso ocorre porque a filosofia do Python é manter a simplicidade e legibilidade do código, evitando a inclusão de construções complexas e redundantes.

12.Questão

Variáveis são espaços de armazenamento na memória que são usados para armazenar valores. Elas têm um nome único e podem conter diferentes tipos de dados.

No Python, os nomes permitidos para variáveis são:

1. customer\_list: É permitido, pois segue a convenção snake\_case, que é comumente usada em Python para nomes de variáveis, onde as palavras são separadas por underscores.

2. print: Não é permitido, pois "print" é uma palavra reservada em Python, utilizada para exibir informações na saída padrão. O uso de palavras reservadas como nomes de variáveis não é permitido.

3. num-people: Não é permitido, pois o caractere "-" não é permitido em nomes de variáveis em Python. Para separar palavras em um nome de variável, é recomendado o uso de underscores "\_".

4. $teste: Não é permitido, pois o caractere "$" não é permitido no início de um nome de variável em Python. Os nomes de variáveis em Python devem começar com uma letra ou um underscore "\_".

5. 2\_numero: Não é permitido, pois um nome de variável em Python não pode começar com um número. O primeiro caractere deve ser uma letra ou um underscore "\_".

Portanto, apenas o nome "customer\_list" seria permitido como uma variável em Python.

13.Questão

Em Python, uma constante é um valor que não pode ser alterado durante a execução do programa. Elas são usadas para representar valores fixos, como números ou strings, que não devem ser modificados.

Embora Python não tenha um tipo de dado específico para constantes, é uma convenção utilizar letras maiúsculas e underscores para nomear variáveis que devem ser tratadas como constantes. Isso ajuda a diferenciá-las das variáveis regulares e indica que seu valor não deve ser alterado.

Por exemplo, podemos declarar uma constante para armazenar o valor de Pi:

python

PI = 3.14159

Nesse caso, "PI" é nomeada em letras maiúsculas para indicar que é uma constante e seu valor deve permanecer o mesmo ao longo do programa.

Vale ressaltar que essa convenção é apenas uma prática recomendada e não há restrições técnicas em Python para impedir a modificação de uma variável declarada como constante.