

Especificações:

- 1) Atividade individual;**
- 2) Os pseudocódigos devem ser feitos e executados na ferramenta VisuAlg;
- 3) Ao terminar, tire um print ou copie e cole numa cópia desse documento;
- 4) Cole os prints de acordo com seu número de questão;
- 5) Mande o arquivo na atividade postada pela professora pela professora.

1) Desenvolva um algoritmo que calcule o valor final de uma refeição em um restaurante. O algoritmo deve seguir estes passos:

1. Receber o preço inicial da refeição e o percentual de gorjeta desejado pelo cliente (por exemplo, 15%).
2. Calcular o valor da gorjeta e o valor total da refeição, incluindo a gorjeta.
3. Aplicar um desconto adicional (por exemplo, 10%) se o preço inicial da refeição (sem incluir a gorjeta) exceder R\$ 100,00.
4. Exibir o valor da gorjeta, o valor do desconto (se aplicável) e o valor final da refeição.

Para essa tarefa:

- A gorjeta é 15% do preço inicial da refeição.
- O desconto é de 10% e só é aplicado se o preço inicial da refeição for superior a R\$ 100,00 (sem incluir a gorjeta)

algoritmo "calcula_refeicao"

var

preco_inicial, percentual_gorjeta, gorjeta, total_com_gorjeta, desconto,
preco_inicial_com_desconto: real

inicio

escreva("Digite o preço inicial da refeição: ")

leia(preco_inicial)

escreva("O percentual da gorjeta é de 15%")

gorjeta := preco_inicial * 15/100

```
total_com_gorjeta := preco_inicial + gorjeta
```

```
desconto := 0
```

```
se preco_inicial > 100 então
```

```
    desconto <- preco_inicial * 0.10
```

```
    preco_inicial_com_desconto <- preco_inicial - desconto
```

```
    total_com_gorjeta <- preco_inicial_com_desconto + gorjeta
```

```
fimse
```

```
escreva("Valor da Gorjeta: R$ ", gorjeta:0:2, "\n")
```

```
se desconto > 0 então
```

```
    escreva("Valor do Desconto: R$ ", desconto:0:2, "\n")
```

```
fimse
```

```
escreva("Valor Final da Refeição: R$ ", total_com_gorjeta:0:2, "\n")
```

```
fimalgoritmo
```

2) Um programa de recompensas por leitura concede pontos com base nas horas de leitura mensais, que podem ser trocados por prêmios. O sistema funciona da seguinte forma:

- Cada hora de leitura no mês vale pontos de acordo com a seguinte escala:
 - Até 5 horas de leitura no mês: ganha 3 pontos por hora
 - De 6 até 15 horas de leitura no mês: ganha 7 pontos por hora
 - Acima de 16 horas de leitura no mês: ganha 12 pontos por hora
- Além disso, se o cliente acumular mais de 50 pontos no mês, um bônus de 10% é adicionado ao total de pontos.
- Cada 10 pontos podem ser trocados por um cupom de desconto de R\$2,00.

Desenvolva um algoritmo que realize os seguintes passos:

1. Receba o total de horas de leitura realizadas no mês pelo cliente.
2. Calcule a quantidade de pontos acumulados com base nas faixas de horas de leitura.
3. Aplique um bônus de 10% se o total de pontos acumulados exceder 50 pontos.
4. Determine o número de cupons de desconto que o cliente pode obter com os pontos acumulados.
5. Exiba o total de pontos ganhos, o valor em cupons de desconto que o cliente pode obter e o valor total de cupons obtidos.

Para essa tarefa:

- Até 5 horas de leitura: 3 pontos por hora
- De 5 até 15 horas de leitura: 7 pontos por hora
- Acima de 15 horas de leitura: 12 pontos por hora
- Bônus de 10% se o total de pontos exceder 50 pontos
- Cada 10 pontos são trocados por um cupom de R\$2,00

algoritmo "calculo_refeicao"

var

total_horas, pontos:inteiro

inicio

escreval("Quantas horas de leitura você tem?")

leia(total_horas)

escreval("A tabela de pontos é essa a seguir")

leia(total_horas)

se total_horas <= 5 entao

 escreval("Você hanhou 3 pontos")

senao

se total_horas >= 5 até 15 entao

 escreval("Você ganhou 7 pontos")

se total_horas > 15 entao

 escreval("Você ganhou 12 pontos")

fimse

fimse

fimse

escreval("Quantos pontos você tem?")

leia(pontos)

se pontos >= 10 entao

escreval("Você pode trocar 10 pontos por um cumpom, aceita?")

leia(resposta)

se "sim" entao

 escreval("Você ganhou 2 reais")

senao

escreval("Você nao ganhou o cumpom")

fimse

fimse

fimalgoritmo

3) Aprimore o jogo de adivinhação conforme as seguintes regras:

1. O computador irá sortear um número aleatório entre 1 e 10.
2. O jogador terá até 4 tentativas para adivinhar o número sorteado.
3. Após cada tentativa, o computador fornecerá uma dica ao jogador:
 - Se o palpite for menor que o número sorteado, o computador dirá "Tente um número maior."
 - Se o palpite for maior que o número sorteado, o computador dirá "Tente um número menor."
 - Se o palpite estiver correto, o computador informará que o jogador venceu e o jogo terminará.

algoritmo "jogo_adivinhacao"

var

numero_sortado, palpite, tentativas, i : inteiro

inicio

// Inicializa o número de tentativas

tentativas <- 4

// Sorteia um número aleatório entre 1 e 10

numero_sortado <- randi(10)

// Inicia o loop para as tentativas

para i de 1 ate tentativas faca

```
// Solicita o palpite ao jogador
```

```
escreva("Tentativa ", i, ": Digite um número entre 1 e 10: ")
```

```
leia(palpite)
```

```
// Verifica se o palpite está correto
```

```
se palpite = numero_sorteado entao
```

```
    escreva("Parabéns! Você acertou o número ", numero_sorteado, "!")
```

```
senao
```

```
    se palpite < numero_sorteado entao
```

```
        escreva("Tente um número maior.")
```

```
    senao
```

```
        escreva("Tente um número menor.")
```

```
    fimse
```

```
fimse
```

```
se palpite <> numero_sorteado entao
```

```
    escreva("Você perdeu! O número era ", numero_sorteado, ".")
```

```
fimse
```

```
fimpara
```

```
fimalgoritmo
```

4. Se o jogador usar todas as tentativas sem acertar o número, o computador informará que o jogador perdeu e revelará o número sorteado.

Desenvolva um pseudocódigo para implementar este jogo.

Notas Adicionais:

- O número aleatório deve ser gerado de forma segura para garantir que esteja dentro do intervalo especificado (1 a 10).
- Certifique-se de que o programa trate entradas inválidas (por exemplo, números fora do intervalo ou entradas não numéricas) de maneira adequada.

algoritmo "jogo_adivinhacao"

var

numero_sortado, palpite, tentativas, i : inteiro

entrada_valida : booleano

inicio

// Inicializa o número de tentativas

tentativas <- 4

// Sorteia um número aleatório entre 1 e 10

numero_sortado <- aleatorio(1,10)

// Inicia o loop para as tentativas

para i de 1 ate tentativas faca

entrada_valida <- falso

// Repetir até obter uma entrada válida

enquanto nao entrada_valida faca

escreva("Tentativa ", i, ": Digite um número entre 1 e 10: ")


```
leia(palpite)
```

```
// Verifica se a entrada é um número válido
```

```
se (palpite >= 1 e palpite <= 10) entao
```

```
    entrada_valida <- verdadeiro
```

```
senao
```

```
    escreva("Entrada inválida. Por favor, digite um número entre 1 e 10.")
```

```
fimse
```

```
fimenquanto
```

```
// Verifica se o palpite está correto
```

```
se palpite = numero_sorteado entao
```

```
    escreva("Parabéns! Você acertou o número ", numero_sorteado, "!")
```

```
pare
```

```
senao
```

```
    se palpite < numero_sorteado entao
```

```
        escreva("Tente um número maior.")
```

```
    senao
```

```
        escreva("Tente um número menor.")
```

```
    fimse
```

```
fimse
```

```
fimp
```

```
// Verifica se o jogador não conseguiu adivinhar
```

```
se palpite <> numero_sorteado entao
```

```
    escreva("Você perdeu! O número era ", numero_sorteado, ".")
```

```
fimalgoritmo
```