

### **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)

Área dos algoritmos ( Edição do código fonte ) -> Nome do arquivo: [re1.ALG]

```
1 Algoritmo "semnome"
2 // Disciplina : [Linguagem e Lógica de Programação]
3 // Professor : Antonio Carlos Nicolodi
4 // Descrição : Aqui você descreve o que o programa faz! (função)
5 // Autor(a) : Nome do(a) aluno(a)
6 // Data atual : 28/08/2024
7 Var
8 // Seção de Declarações das variáveis
9 g, v, gorjeta, desconto, total, dt: real
10
11 Inicio
12 // Seção de Comandos, procedimento, funções, operadores, etc...
13 g <- 0.15
14 escreval("Qual o valor total da sua refeição?")
15 leia(v)
16 se v > 100 entao
17 desconto <- v * 0.1
18 dt <- v - desconto
19 gorjeta <- v * g
20 total <- gorjeta + dt
21 escreval("O valor da gorjeta é de", gorjeta, " , o valor do desconto é de", des
22 senao
23 gorjeta <- v * g
24 total <- gorjeta + v
25 escreval("O valor da gorjeta é de", gorjeta, " , e o valor final é de", total)
26 fimse
27
28 Fimalgoritmo
```

```
Área dos algoritmos ( Edição do código fonte ) -> Nome do arquivo: [re1.ALG] -
1
2  ção]
3
4  ama faz! (função)
5
6
7
8
9
10
11
12 adores, etc...
13
14
15
16
17
18
19
20
21 o valor do desconto é de", desconto, " e o valor final é de", total)
22
23
24
25 e o valor final é de", total)
26
27
28
```

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

Área dos algoritmos ( Edição do código fonte ) -> Nome do arquivo: [re2.ALG]

```
7 Var
8 // Seção de Declarações das variáveis
9 p, cq, cv, cqt: real
10 h: inteiro
11
12 Inicio
13 // Seção de Comandos, procedimento, funções, operadores, etc...
14 escreval("Por quantas horas você leu esse mês?")
15 leia(h)
16 se h <= 5 entao
17 p <- h * 3
18   se p > 50 entao
19     p <- p + (p * 0.1)
20   fimse
21 cq <- (p / 10)
22 cqt <- (p % 10) - cq
23 cv <- cqt * 2
24 escreval("Você ganhou", p, " pontos, ", cq, " cupons que dão ", cv, "R$ de des.",
25 fimse
26 se (h > 5) e (h <= 15) entao
27 p <- 7 * h
28   se p > 50 entao
29     p <- p + (p * 0.1)
30   fimse
31 cq <- (p / 10)
32 cqt <- (p % 10) - cq
33 cv <- cqt * 2
34 escreval("Você ganhou", p, " pontos, ", cq, " cupons que dão ", cv, "R$ de desc.",
35 fimse
36 se h > 15 entao
37 p <- 12 * h
38   se p > 50 entao
39     p <- p + (p * 0.1)
40   fimse
41 cq <- (p / 10)
42 cqt <- (p % 10) - cq
43 cv <- cqt * 2
44 escreval("Você ganhou", p, " pontos, ", cq, " cupons que dão ", cv, "R$ de desc.",
45 fimse
46
47
48 Fimalgoritmo
```

Área dos algoritmos ( Edição do código fonte ) -> Nome do arquivo: [re2.ALG] -

```
7
8 is
9
10
11
12
13 funções, operadores, etc...
14 u esse mês?)
15
16
17
18
19
20
21
22
23
24 , ", cqt, " cupons que dão ", cv, "R$ de desconto")
25
26
27
28
29
30
31
32
33
34 , ", cq, " cupons que dão ", cv, "R$ de desconto")
35
36
37
38
39
40
41
42
43
44 , ", cq, " cupons que dão ", cv, "R$ de desconto")
45
```

### 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.

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.

Área dos algoritmos ( Edição do código fonte ) -> Nome do arquivo: [re3.ALG] -

```
1 Algoritmo "semnome"
2 // Disciplina : [Linguagem e Lógica de Programação]
3 // Professor : Antonio Carlos Nicolodi
4 // Descrição : Aqui você descreve o que o programa faz! (função)
5 // Autor(a) : Nome do(a) aluno(a)
6 // Data atual : 28/08/2024
7 Var
8 // Seção de Declarações das variáveis
9 c, r, s: inteiro
10
11 Inicio
12 // Seção de Comandos, procedimento, funções, operadores, etc...
13 s <- randi(10)
14 escreval("Você tem 4 tentativas para acertar o numero sorteado entre 1 e 10")
15 c <- 4
16 repita
17 c <- c - 1
18 escreval("Digite um numero")
19 leia(r)
20 se (r < 1) ou (r > 10) entao
21 escreval("Opção invalida digite outro numero")
22 leia(r)
23 fimse
24 se r > s entao
25 escreval("Tente um número menor")
26 fimse
27 se r < s entao
28 escreval("Tente um número maior")
29 fimse
30 se r = s entao
31 escreval("Você venceu, o jogo terminou")
32 fimse
33 ate (c = 0) ou (r = s)
34 escreval("O numero sorteado foi", s)
35 Fimalgoritmo
```



