

UNIDADE 1

1.

- Entrar na fila.
- Aguardar até chegar a minha vez.
- Inserir o cartão no caixa eletrônico.
- Selecionar a opção de pagamento de conta.
- Ler o código de barras.
- Digitar a senha.
- Imprimir o comprovante.

2.

(C) cidade

(I) endereço_nr tem um caractere especial inválido 'ç'

(I) média_peso% tem um caractere especial inválido '%'

(I) aluno-nota sinal de operação aritmética '-'

(I) media idade não pode conter espaço

(C) a3

(I) endereço.cep caracteres especiais inválidos 'ç' e '.'

(C) B5

(C) nome2

(I) 4cidade não pode começar com número

(C) cliente_nome

(I) 1234P não pode começar com número

3.

Algoritmo unilexe3**Var** num: **inteiro****Início** **Escreva**("Digite o valor:") **Leia**(num) **Escreva**("O número é:", num) **Escreva**("A raiz quadrada é:", sqrt(num)) **Escreva**("O quadrado é:", sqr(num))**Fim**

4.

exec5a:	exec5b:
5	30
5	5
5	5

UNIDADE 2

1.

Algoritmo uni2exel**Var**matricula: **inteiro**nome: **vetor**[1..64] **de caractere**salario, vendas, total: **real****Início****Escreva**("Informe a matrícula do funcionário:")**Leia**(matricula)**Escreva**("Informe o nome do funcionário:")**Leia**(nome)**Escreva**("Informe o salário fixo do funcionário:")**Leia**(salario)**Escreva**("Informe o valor das vendas do funcionário:")**Leia**(vendas)**Se** (vendas < 1500) **então**

total <- salario + (vendas * 0.02)

Senão

total <- salario + (vendas * 0.04)

Fim_se**Escreva**("Vendedor: ", nome)**Escreva**("Matrícula: ", matricula)**Escreva**("Salário fixo: ", salario)**Escreva**("Salário total: ", total)**Fim**

2.

```
Algoritmo uni2exe2
Var
    numero : inteiro
Início
    Escreva("Informe o número:")
    Leia(numero)
    Se (((numero MOD 3)=0) E ((numero MOD 7)=0))
então
        Escreva("O número ", numero, " é divisível
        por 3 e por 7")
    Senão
        Escreva("O número ", numero, " NÃO é di-
        visível por 3 e por 7")
    Fim_se
Fim
```

3.

Algoritmo uni2exe3**Var**num1, num2, num3, num4, num5, qtd : **inteiro****Início****Escreva**("Informe o primeiro número:")**Leia**(num1)**Escreva**("Informe o segundo número:")**Leia**(num2)**Escreva**("Informe o terceiro número:")**Leia**(num3)**Escreva**("Informe o quarto número:")**Leia**(num4)**Escreva**("Informe o quinto número:")**Leia**(num5)

qtd <- 0

Se (num1 < 0) **então**

qtd <- qtd + 1

Fim_se

```
Se (num2 < 0) então
    qtd <- qtd + 1
Fim_se
Se (num3 < 0) então
    qtd <- qtd + 1
Fim_se
Se (num4 < 0) então
    qtd <- qtd + 1
Fim_se
Se (num5 < 0) então
    qtd <- qtd + 1
Fim_se
Fim
```

4.

Algoritmo uni2exe4**Var** sexo: **caractere** altura, peso ideal: **real****Início** **Escreva**("Informe o sexo [M/F]:") **Leia**(sexo) **Escreva**("Informe a altura:") **Leia**(altura) **Se** (sexo = 'M') **então**

peso_ideal <- (72.7 * altura) - 58

Senão

peso_ideal <- (62.1 * altura) - 44.7

Fim_se **Escreva**("Seu peso ideal é: ", peso_ideal)**Fim**

5.

Algoritmo uni2exe5**Var** moto: **caractere** percurso, consumo: **real****Início** **Escreva**("Informe o tipo da moto [A/B/C]:") **Leia**(moto) **Escreva**("Informe o tamanho o percurso (em KM) :") **Leia**(percurso) **Caso** (moto) **Seja** 'A' **faça**

consumo <- percurso / 26

Seja 'B' **faça**

consumo <- percurso / 20

Seja 'C' **faça**

consumo <- percurso / 7

Fim_caso **Escreva**("O consumo estimado de combustível é: ",
 consumo)**Fim**

6.

Algoritmo uni2exe6**Var** saldo: **real****Início** **Escreva**("Informe o saldo médio do correntista:") **Leia**(saldo) **Se** (saldo < 500) **então** **Escreva**("Linha de crédito indisponível.") **Fim_se** **Se** ((saldo >= 500) **E** (saldo <= 1000)) **então** **Escreva**("Linha de crédito disponível no valor de:", saldo * 0.35) **Fim_se** **Se** ((saldo > 1000) **E** (saldo <= 3000)) **então** **Escreva**("Linha de crédito disponível no valor de:", saldo * 0.50)

```
Fim_se  
  
Se (saldo > 3000) então  
    Escreva("Linha de crédito disponível no  
    valor de:", saldo * 0.75)  
  
Fim_se  
  
Fim
```

UNIDADE 3

1.

```
Algoritmo uni3exel  
  
Var  
    qtd, contador, numero, maior, menor: inteiro  
    primeiro: lógico  
  
Início  
    contador <- 0  
    primeiro <- verdadeiro  
    Escreva("Informe quantos números deseja ler:")  
    Leia(qtd)  
    Enquanto (contador < qtd) faça  
        Leia(numero)  
        Se (primeiro = verdadeiro) então  
            maior <- numero
```

```
        menor <- numero
        primeiro <- falso

Fim_se

Se (numero > maior) então
        maior <- numero

Fim_se

        menor <- numero

Fim_se

        contador <- contador + 1

Fim_enquanto

Escreva("O maior número lido foi: ", maior)

Escreva("O menor número lido foi: ", menor)

Fim
```

2.

Algoritmo uni3exe2**Var**

```
    numero, contador, pares, impares: inteiro
    media: real
```

Início

```
pares <- 0
impares <- 0
media <- 0
```

```
contador <- 0

Escreva("Digite um número para efetuar o cálculo ou 0 para finalizar:")

Leia(numero)

Enquanto (numero <> 0) faça

    Se ((numero MOD 2) = 0) então

        pares <- pares + 1

    Senão

        impares <- impares + 1

    Fim_se

    media <- media + numero

    contador <- contador + 1

    Escreva("Digite um novo número para efetuar o cálculo ou 0 para finalizar:")

    Leia(numero)

Fim_enquanto

media <- media / contador

Escreva("Foram lidos ", contador, " números, dos quais ", pares, " eram pares.")

Escreva("Também foram encontrados ", impares, " números ímpares.")

Escreva("E a média aritmética dos números lidos é: ", media)
```

Fim

3.

```
Algoritmo uni3exe3  
Var  
    num: inteiro  
Início  
    Para num de 0 até 199 passo 1 faça  
        Se ((num MOD 5) = 0) então  
            Escreva(num)  
        Fim_se  
    Fim_para  
Fim
```

4.

```
Algoritmo uni3exe4  
Var  
    contador, numero, maior, menor: inteiro  
    media: real  
    primeiro: lógico  
Início  
    primeiro <- verdadeiro  
    media <- 0  
    contador <- 0  
  
    Escreva("Informe o valor:")
```

```
Leia (numero)

Enquanto (numero >= 0) faça
    Se (primeiro = verdadeiro) então
        maior <- numero
        menor <- numero
        primeiro <- falso
    Fim_se
    Se (numero > maior) então
        maior <- numero
    Fim_se
    Se (numero < menor) então
        menor <- numero
    Fim_se
    media <- media + numero
    contador <- contador + 1
    Escreva ("Informe o valor:")
    Leia (numero)
Fim_enquanto

media <- media / contador

Escreva ("O maior número lido foi: ", maior)
Escreva ("O menor número lido foi: ", menor)
Escreva ("A média dos números lidos é: ", media)

Fim
```

5.

```
Algoritmo uni3exe5  
Var  
    i, j: inteiro  
Início  
    Para i de 1 até 10 passo 1 faça  
        Escreva("Tabuada do ", i)  
        Para j de 1 até 10 passo 1 faça  
            Escreva(i, " * ", j, " = ", i*j)  
        Fim_para  
    Fim_para  
Fim
```

UNIDADE 4

1.

```
Algoritmo uni4exe1  
Var  
    nome: Vetor[1..20, 0..64] de caractere  
    num: inteiro  
Início  
    Para num de 1 até 20 passo 1 faça
```

```
        Escreva("Informe o ", num, " nome: ")
        Leia(nome[num])

Fim_para

Para num de 1 até 20 passo 1 faça

        Escreva(nome[num][1])

Fim_para

Fim
```

2.

Algoritmo uni4exe2**Var**nome: **Vetor**[1..20, 1..64] **de** caracterenotas: **Vetor**[1..20, 1..4] **de** realmedia: **Vetor**[1..20] **de** reali, j: **inteiro**media_turma: **real****Início**

media_turma <- 0

Para i **de** 1 **até** 20 **passo** 1 **faça** **Escreva**("Informe o ", i, " nome: ") **Leia**(nome[i])

media[i] <- 0


```
    Para j de 1 até 4 passo 1 faça
        Escreva("Informe a nota", j)
        Leia(notas[i, j])
        media[i] <- media[i] + notas[i, j]
    Fim_para
    media[i] <- media[i] / 4
    media_turma <- media_turma + media[i]
Fim_para
media_turma <- media_turma / 20
Para i de 1 até 20 passo 1 faça
    Escreva("Aluno", nome[i])
    Para j de 1 até 4 passo 1 faça
        Escreva("Nota", j, " é:", notas[i, j])
    Fim_para
    Escreva("A média do aluno", nome[i],
        "foi:", media[i])
Fim_para
Escreva("A média da turma é:", media_turma)
Fim
```

3.

```
Algoritmo uni4exe3

Var

    nome: Vetor[1..64] de caractere

Início

    Repita

        Escreva("Digite o nome, digite FIM para sair:")

        Leia(nome)

        Escreva(nome[1])

    Até_que (nome = "FIM")

Fim
```

4.

```
Algoritmo uni4exe4

Var

    num, maior, menor: inteiro

    numeros: Vetor[1..50] de inteiro

    primeiro: lógico

Início

    primeiro <- verdadeiro

    Escreva("Digite os 50 números")
```

```
Para num de 1 até 50 passo 1 faça  
    Leia(numeros[num])  
  
    Se (primeiro = verdadeiro) então  
        maior <- numeros[num]  
        menor <- numeros[num]  
        primeiro <- falso  
    Fim_se  
    Se (numeros[num] > maior) então  
        maior <- numeros[num]  
    Fim_se  
    Se (numeros[num] < menor) então  
        menor <- numeros[num]  
    Fim_se  
Fim_para  
Escreva("O maior número lido foi: ", maior)  
Escreva("O menor número lido foi: ", menor)  
  
Fim
```

5.

Algoritmo uni4exe5**Var** num: **inteiro** a, b, c : **Vetor**[1..20] **de inteiro****Início** **Escreva**("Digite os 20 números do Vetor A") **Para** num **de** 1 **até** 20 **passo** 1 **faça** **Escreva**("Digite o número ", num) **Leia**(a[num]) **Fim_para** **Escreva**("Digite os 20 números do Vetor B") **Para** num **de** 1 **até** 20 **passo** 1 **faça** **Escreva**("Digite o número ", num) **Leia**(b[num]) **Fim_para** **Escreva**("Calculando o vetor $C = A + B$ ") **Para** num **de** 1 **até** 20 **passo** 1 **faça**

c[num] <- a[num] + b[num]

Fim_para **Escreva**("Os valores do vetor $C = A + B$ são: ") **Para** num **de** 1 **até** 20 **passo** 1 **faça**

```
        Escreva (num, ") ", c[num])  
  
    Fim_para  
  
Fim
```

6.

```
Algoritmo uni4exe6  
  
    Tipo  
        cad_produto = registro  
            nome: Vetor[1..64] de caractere  
            p_compra: real  
            p_venda: real  
  
        Fim_registro  
  
    Var  
        num, lucro_maior, lucro_menor: inteiro  
        produto: Vetor[1..30] de cad_produto  
  
    Início  
        lucro_maior <- 0  
        lucro_menor <- 0  
  
        Escreva ("Cadastre os 30 produtos ")  
  
        Para num de 1 até 30 passo 1 faça  
            Escreva ("Digite o nome do produto ", num)  
            Leia (produto[num].nome)
```

```
Escreva("Digite o preço de compra do  
produto ", produto[num].nome)  
Leia(produto[num].p_compra)  
Escreva("Digite o preço de venda do  
produto ", produto[num].nome)  
Leia(produto[num].p_venda)  
Se (((produto[num].p_venda-produto[num].  
p_compra) / produto[num].p_compra)  
< 0.15) então  
    lucro_menor <- lucro_menor + 1  
Fim_se  
Se (((produto[num].p_venda-produto[num].  
p_compra) / produto[num].p_compra)  
> 0.30) então  
    lucro_maior <- lucro_maior + 1  
Fim_se  
Fim_para  
Escreva("Existem ", lucro_menor, " produtos  
com lucro menor do que 15%")  
Escreva("Existem ", lucro_maior, " produtos  
com lucro maior do que 30%")  
Fim
```

7.

Algoritmo uni4exe7**Var****i, j: inteiro****matriz: Vetor[1..5, 1..5] de inteiro****Início****Escreva**("Entre com os valores da matriz 5x5 ")**Para i de 1 até 5 passo 1 faça****Para j de 1 até 5 passo 1 faça****Escreva**("Digite o valor ", i, " x ", j)**Leia**(matriz[i, j])**Fim_para****Fim_para****Escreva**("Os valores da diagonal da matriz ", i, " x ", j, "são:")**Para i de 1 até 5 passo 1 faça****Escreva**("Posição ", i, " x ", i, " = "
matriz[i, i])**Fim_para****Fim**

8.

Algoritmo uni4exe8**Tipo**cad_aluno = **registro**nome: **Vetor**[1..64] **de** caracterenotas: **Vetor**[1..4] **de** realmedia: **real****Fim_registro****Var**i, j: **inteiro**alunos: **Vetor**[1..5] **de** cad_aluno

temp: cad_aluno

Início**Escreva**("Entre com os dados dos alunos")**Para** i **de** 1 **até** 5 **passo** 1 **faça****Escreva**("Digite o nome do aluno: ")**Leia**(alunos[i].nome)

alunos[i].media <- 0

Para j **de** 1 **até** 4 **passo** 1 **faça****Escreva**("Digite a nota ", j)**Leia**(alunos[i].notas[j])


```
        alunos[i].media <- alunos[i].media +
            alunos[i].notas[j]

Fim_para

    alunos[i].media <- alunos[i].media/4

Fim_para

Para i de 1 até 5 passo 1 faça

    Para j de i+1 até 5 passo 1 faça

        Se (alunos[i].media > alunos[j].
            media) então

            temp <- alunos[i]

            alunos[i] <- alunos[j]

            alunos[j] <- temp

        Fim_se

    Fim_para

Fim_para

Escreva("A classificação dos alunos por média
    crescente é:")

Para i de 1 até 5 passo 1 faça

    Escreva(alunos[i].nome, "média: ",
        alunos[i].media)

Fim_para

Fim
```

9.

Algoritmo uni4exe9**Var****i, j: inteiro****a, b, soma, dif : Vetor[1..5, 1..5] de inteiro****Início****Escreva**("Entre com os valores da primeira matriz")**Para i de 1 até 5 passo 1 faça****Para j de 1 até 5 passo 1 faça****Escreva**("Digite o valor ", i, " x ", j)**Leia**(a[i, j])**Fim_para****Fim_para****Escreva**("Entre com os valores da segunda matriz")**Para i de 1 até 5 passo 1 faça****Para j de 1 até 5 passo 1 faça****Escreva**("Digite o valor ", i, " x ", j)**Leia**(b[i, j])

soma[i,j] <- a[i, j] + b[i, j]

dif[i,j] <- a[i, j] - b[i, j]

Fim_para

```
Fim_para

Escreva("A matriz resultante da soma das ma
trizes:")

Para i de 1 até 5 passo 1 faça
    Para j de 1 até 5 passo 1 faça
        Escreva("Posição ", i, " x ", j,
soma[i, j])
    Fim_para
Fim_para

Escreva("A matriz resultante da diferença das
matrizes:")

Para i de 1 até 5 passo 1 faça
    Para j de 1 até 5 passo 1 faça
        Escreva("Posição ", i, " x ", j,
dif[i, j])
    Fim_para
Fim_para

Fim
```

10.

Algoritmo uni4exe10**Var****i, j: inteiro****contas: Vetor[1..12, 1..4] de real****mensal: Vetor[1..12] de real****total: real****Início****Escreva**("Entre com os valores das contas a pagar")**total** <- 0**Para i de 1 até 12 passo 1 faça****mensal[i]** <- 0**Para j de 1 até 4 passo 1 faça****Escreva**("Digite o valor do mês ",
i, ", semana ", j)**Leia**(contas[i, j])**mensal[i]** <- **mensal[i]** + **contas[i, j]****Fim_para****total** <- **total** + **mensal[i]****Fim_para****Escreva**("O total de contas a pagar, por mês é:")

Para i de 1 até 12 passo 1 faça

Escreva("Total a pagar no mês ", i,
" é : ", mensal[i])

Fim_para

Escreva("O total de contas a pagar no ano
é:", total)

Fim

UNIDADE 5

1.

Algoritmo cadastros

Tipo

CadProduto = **registro**

codigo: **inteiro**

descricao: **Vetor**[1..50] **de caractere**

unidade: **Vetor**[1..5] **de caractere**

preco: **real**

Fim_registro

Procedimento mostrarMenu()

Início

Escreva("1 - Cadastrar registro")

```
Escreva("2 - Pesquisar um produto pelo  
código")
```

```
Escreva("3 - Classificar por ordem de  
descrição")
```

```
Escreva("4 - Classificar por ordem de  
código")
```

```
Escreva("5 - Apresentar todos os  
registros")
```

```
Escreva("6 - Sair do programa de  
cadastro")
```

Fim_procedimento

Procedimento produtosCadastrar()

Início

Se (qtdProd < 20) **então**

```
Escreva("Digite o código: ")
```

```
Leia(produtos[qtdProd+1].codigo)
```

```
Escreva("Digite a descrição: ")
```

```
Leia(produtos[qtdProd+1].descricao)
```

```
Escreva("Digite a unidade: ")
```

```
Leia(produtos[qtdProd+1].unidade)
```

```
Escreva("Digite o preço: ")
```

```
        Leia (produtos[qtdProd+1].preco)

        qtdProd <- qtdProd + 1

        Escreva ("Produto cadastrado com
        sucesso!")

    Senão

        Escreva ("Limite máximo atingido:
        20 produtos!")

    Fim_se

    opt <- 0

Fim_procedimento

Função produtosBuscar(cod: inteiro): inteiro

Var

    i, achou: inteiro

Início

    i <- 1

    achou <- 0

    Enquanto (i <= qtdProd) faça

        Se (produtos[i].codigo == cod) então

            Escreva ("Produto: ", cod)

            Escreva ("Descrição: ",
            produtos[i].descricao)
```

```
        Escreva ("Unidade: ",
               produtos[i].unidade)

        Escreva ("Preço:   ",   produtos
               [i].preco)

        Retorne i

    Fim_se

    i <- i + 1

Fim_enquanto

Escreva ("Produto não encontrado...")

Retorne -1

Fim_função

Procedimento produtosOrdenarDescr()

Var

    i, j, k: inteiro

    aux: CadProduto

Início

    Para i de 1 até qtdProd passo 1 faça

        Para j de i+1 até qtdProd passo
        1 faça

            k <- 1

            Enquanto ((produtos[i].des-
            cricao[k] = produtos[j].des-
            cricao[k]) e (k <= 50)) faça
```



```
        k <- k + 1

    Fim_enquanto

    Se ((produtos[i].descricao[k]
> produtos[j].descricao[k]) e
(k <= 50)) então

        aux <- produtos[i]
        produtos[i] <- produtos[j]
        produtos[j] <- aux

    Fim_se

Fim_para

Fim_para

Fim_procedimento

Procedimento produtosOrdenarCod()

Var

    i, j: inteiro

    aux: CadProduto

Início

    Para i de 1 até qtdProd passo 1 faça

        Para j de i+1 até qtdProd passo 1
        faça

            Se (produtos[i].codigo >
produtos[j].codigo)
```

```
        então
            aux <- produtos[i]
            produtos[i] <- produtos[j]
            produtos[j] <- aux
        Fim_se
    Fim_para
Fim_procedimento

Procedimento produtosMostrar()
Var
    i: inteiro
Início
    Para i de 1 até qtdProd passo 1 faça
        Escreva("Produto: ", produtos[i].
            codigo)
        Escreva("Descrição: ", produtos[i].
            descricao)
        Escreva("Unidade: ", produtos[i].
            unidade)
        Escreva("Preço: ", produtos[i].
            preco)
```

```
        Fim_para
    Fim_procedimento

Var
    produtos: Vetor[1..20] de CadProduto
    opt, qtdProd, i: inteiro

Início
    qtdProd <- 0

    Repita
        mostrarMenu()

        Leia(opt)

        Caso (opt)
            Seja 1 faça
                Para i de 1 até 20 passo 1 faça
                    produtosCadastrar()

                Fim_para

            Seja 2 faça
                produtosBuscar()

            Seja 3 faça
                produtosOrdenarDescr()

            Seja 4 faça
                produtosOrdenarCod()

            Seja 5 faça
                produtosMostrar()
```

```
Fim_caso  
Até_que (opt = 6)  
Fim
```

2.

```
Algoritmo cadastroPessoas  
  
Tipo  
    CadPessoa = registro  
        nome: Vetor[1..50] de caractere  
        idade: inteiro  
        altura: real  
        peso: real  
  
Fim_registro  
  
Procedimento mostrarMenu()  
  
Início  
    Escreva("1 - Cadastrar pessoa")  
    Escreva("2 - Classificar por ordem de  
    idade")  
    Escreva("3 - Calcular média de peso")  
    Escreva("4 - Mostrar quem está acima do  
    limite de 50")  
    Escreva("5 - Sair do programa de cadas-  
    tro")
```

Fim_procedimento

Procedimento pessoaCadastrar()

Início

Se (qtdPessoas < 10) **então**

Escreva("Insira o nome: ")

Leia(pessoas[qtdPessoas+1].nome)

Escreva("Insira a idade: ")

Leia(pessoas[qtdPessoas+1].idade)

Escreva("Insira a altura: ")

Leia(pessoas[qtdPessoas+1].altura)

Escreva("Insira o peso: ")

Leia(pessoas[qtdPessoas+1].peso)

qtdPessoas <- qtdPessoas + 1

Senão

Escreva("Limite máximo de 10
pessoas atingido")

Fim_se

Fim_procedimento

Procedimento pessoasOrdenarIdade()

Var

i, j: inteiro

```
    aux: CadPessoa

Início

    Para i de 1 até qtdPessoas passo 1 faça
        Para j de i+1 até qtdPessoas passo
            1 faça
                Se (pessoas[i].idade >
                    pessoas[j].idade) então
                    aux <- pessoas[i]
                    pessoas[i] <- pessoas[j]
                    pessoas[j] <- aux

                Fim_se

            Fim_para

        Fim_para

    Fim_procedimento

Função mediaPesos(): real

Var

    i: inteiro
    media: real

Início

    media <- 0

    Para i de 1 até qtdPessoas passo 1 faça
```

```
        media <- media + pessoas[i].peso

    Fim_para

    Retorne media / qtdPessoas

Fim_função

Procedimento    nomePessoasPesoMaiorQue (peso:
real)

Var

    i: inteiro

Início

    Para i de 1 até qtdPessoas passo 1 faça
        Se (pessoas[i].peso > peso) então
            Escreva(i, " - Nome: ",
                pessoas[i].nome)

        Fim_se

    Fim_para

Fim_procedimento

Var

    pessoas: Vetor[1..10] de CadPessoa
    opt, qtdPessoas: inteiro
```

Início

```
qtdPessoas <- 0
```

Repita

```
mostrarMenu()
```

```
Leia (opt)
```

```
Caso (opt)
```

```
    Seja 1 faça
```

```
        Para i de 1 até 10 passo 1
```

```
        faça pessoaCadastrar()
```

```
        Fim_para
```

```
    Seja 2 faça
```

```
        pessoasOrdenarIdade()
```

```
    Seja 3 faça
```

```
        Escreva ("Média dos pesos: ",
```

```
        mediaPesos())
```

```
    Seja 4 faça
```

```
        nomePessoasPesoMaiorQue(50)
```

```
    Fim_caso
```

```
Até_que (opt = 5)
```

Fim

3.

```
Função testaCaractere(c: caractere): inteiro  
Início  
    Se ((c = 'A') ou (c = 'a') ou (c = 'E') ou  
        (c = 'e')  
  
        ou (c = 'I') ou (c = 'i') ou (c = 'O') ou  
        c = 'o')  
  
        ou (c = 'U') ou (c = 'u')) então  
        Retorne 0  
  
    Senão  
        Retorne 1  
  
    Fim_se  
Fim_função
```

4.

```
Algoritmo somatorio  
  
    Função somador(limite: inteiro): inteiro  
        Var soma, i: inteiro  
  
        Início  
            soma <- 0  
  
            Para i de 1 até limite passo 1 faça  
                Se ((i MOD 2) = 0) então  
                    soma <- soma + i
```

```
        Fim_se
    Fim_para
    Retorne soma
Fim_função
Var
    num: inteiro
Início
    Escreva("Gostaria de somar os inteiros pares
    até qual limite?")
    Leia(num)
    Escreva("O somatório é: ", somador(num))
Fim
```

5.

```
Procedimento parOuImpar(num: inteiro)
Início
    Se ((num MOD 2) = 0) então
        Escreva("O número ", num, " é par.")
    Senão
        Escreva("O número ", num, " é ímpar.")
    Fim_se
Fim_procedimento
```

6.

Algoritmo valorAbsoluto

Função abs(num: **real**): **real**

Início

Se (num < 0) **então**

Retorne num * (-1)

Senão

Retorne num

Fim_se

Fim

Var

x: **real**

Início

Escreva("Digite um número real qualquer:")

Leia(x)

Escreva("O valor absoluto é: ", abs(x))

Fim

7.

Algoritmo conversao

Função converteParaReais(c: **real**, q: **real**):
real

Início

Retorne c * q

Fim_função

Var

```
cot, qtde: real
```

Início

```
Escreva("Qual é a cotação atual do Euro?")
```

```
Leia(cot)
```

```
Escreva("Quantos Euros você deseja converter?")
```

```
Leia(qtde)
```

```
Escreva("O valor em reais é: ",  
converteParaReais(cot, qtde))
```

Fim

8.

Algoritmo conversao

```
Função converteParaFahrenheit(c: real): real
```

Início

```
Retorne (c * 1.8) - 32
```

Fim_função

Var

```
celcius: real
```

Início

```
Escreva("Qual a temperatura em °C?")
```

```
Leia(celcius)
```

```
Escreva("Convertendo para °F: ",  
converteParaFahrenheit(celcius))
```

Fim