

Funções:

1. Escreva um algoritmo que solicita ao usuário dois valores reais via teclado e depois exibe a soma dos números fornecidos.

Dica: Utilize o operador + para efetuar a soma dos valores.

```
Início {  
    // Declaração das variáveis  
  
    Real: n1;  
  
    Real: n2;  
  
    Real: soma;  
  
  
    // Leitura dos valores  
  
    Leia ("Valor 1", n1);  
    Leia ("Valor 2", n2);  
  
  
    // Soma dos valores  
  
    Soma = n1 + n2;  
  
  
    // Exibição da soma  
  
    Escreva (soma);  
  
}
```

2. Escreva um algoritmo que solicita ao usuário três valores inteiros via teclado e depois exibe os números fornecidos.

```
Início {  
    // Declaração das variáveis  
    Inteiro: n1;  
    Inteiro: n2;  
    Inteiro: n3;  
    // Leitura dos valores  
    Leia ("Valor 1", n1);  
    Leia ("Valor 2", n2);  
    Leia ("Valor 3", n3);  
  
    // Exibição dos valores  
    Escreva (n1, " ", n2, " ", n3);  
}
```

3. A partir do seguinte trecho de algoritmo, informe o valor armazenado na variável “MEDIA” ao final de sua execução.

Resposta 7,23

4. Analise o algoritmo abaixo desenvolvido no *Visualg* e identifique o(s) problema(s).

| | |
|--|---|
| <pre>Algoritmo “cálculo do ano de nascimento” // O algoritmo deverá informar o ano de nascimento a partir da idade de uma pessoa Algoritmo “cálculo do ano de nascimento” Var Nome, idade, calculo: inteiro Início Escreve (“Digite o seu nome: “) Leia (nome) Escreva (“Digite a sua idade”) Leia (calculo) Calculo <- (2016 + idade) Escreva (“O(A) “,nome,” nasceu no ano de “,calculo) finalgoritmo</pre> | <p>A variável nome deve ser do tipo “caractere”</p> <p>O comando de saída no visualg é o “Escreva” ou “Escreval”</p> <p>A variável utilizada para ler a idade do usuário deve ser a “idade” e não a “calculo”</p> <p>No cálculo, para descobrir o ano de nascimento, deveremos utilizar o sinal de subtração “-” ao invés do sinal de adição.</p> |
|--|---|

5. Faça um algoritmo que calcule a média do aluno a partir de três notas e escreva na tela “aprovado” se ela é maior ou igual a 7,0 e “reprovado” se a média for menor que 7,0.

Algoritmo “cálculo média do aluno”

Var

N1, N2, N3, MEDIA: real

Início

Escreva (“Digite a primeira nota do aluno: “)

Leia (N1)

N1 <- 7,6

N2 <- 8,2

N3 <- 5,9

MEDIA <- (N1 + N2 + N3) / 3

Se (MEDIA >= 7) então

 Escreva (“O aluno foi aprovado”)

senao

 Escreva (“O aluno foi reprovado”)

fimse

fimalgoritmo

6. Escreva um algoritmo que guarda alguns contatos em uma agenda telefônica (cada contato é formado por um nome e um telefone). Depois, o usuário deverá fornecer um nome, que será utilizado para pesquisar na agenda. Se o nome for encontrado, o nome e o telefone do contato devem ser exibidos. Caso contrário, uma mensagem indicando que o registro não foi encontrado deve ser exibida.

```
// Cria o contato
tipo contato {
    caractere: nome;
    inteiro: telefone;
}
início {
    // Arquivo para armazenar os registros
    arquivo: agenda;
    // Vetor com os contatos a serem cadastrados
    contatos: contatos[5];
    // Variáveis de controle
    contato: aux;
    inteiro: i;
    caractere: nome;

    // Cadastro de alguns contatos
    contatos[0].nome = "Pedro";
    contatos[0].telefone = 11114632;
    contatos[1].nome = "Paulo";
    contatos[1].telefone = 11118427;
    contatos[2].nome = "Maria";
    contatos[2].telefone = 11117523;
    contatos[3].nome = "Ana";

    // Abre o arquivo
    Abra (agenda);
    // Armazena todos os registros na agenda
    Para i de 0 até 4 passo 1 {
        guarde (agenda, contatos[i]);
        avance (agenda);
    }
    // Fecha o arquivo
    Feche (agenda);
    // Solicita um nome para pesquisa
    Leia ("Nome", nome);
    // Abre a agenda
    Abra (agenda);
    // Itera sobre todos os elementos do arquivo,
    até que o nome desejado seja encontrado
    // ou os registros terminem
    repita {
        copie agenda, aux);
        avance (agenda);
    } enquanto (aux.nome <> nome e não da
    (agenda));

    se (aux.nome == nome) {
```

```
contatos[3].telefone = 11112309;           // Encontrou o registro, mostra os dados
contatos[4].nome = "João";                 escreva(aux.nome, " -> ", aux.telefone);
contatos[4].telefone = 11116652;           } senão {
                                           // O registro não foi encontrado
                                           Escreva ("Nome ", nome, " não existe");
                                           }

                                           // Fecha o arquivo
                                           Feche (agenda);
                                           }
```

7. Escreva um algoritmo que cria dois arquivos cujos elementos são livros (um livro é composto por nome, autor, assunto e número de páginas). Em cada um dos arquivos, insira alguns livros de exemplo. Depois, o algoritmo deve criar um terceiro arquivo, também de livros, com todos os registros presentes nos dois arquivos anteriormente criados. Após criar esse novo arquivo, imprima os dados de todos os livros cadastrados nele.

```
// cria o tipo livro
tipo livro {
    caractere: título;
    caractere: autor;
    caractere: assunto;
    inteiro: num_pag;
}

início {
    // arquivos ('bib' são arquivos que conterão os
registros de 'bib1' e 'bib2')
    arquivo: bib;
    arquivo: bib1;
    arquivo: bib2;

    livro: l;

    // grava alguns registros em 'bib1'
abra(bib1);
l.titulo = "lógica de programação";
l.autor = "Cristiani Dias";
l.assunto = " lógica de programação ";
l.num_pag = 125;
guarde (bib1, l);
avance(bib1);

l.titulo = "aprenda programação ";
l.autor = "Daniel Buarque";
l.assunto = " lógica de programação ";
l.num_pag = 200;
guarde (bib1, l);
avance(bib1);

l.titulo = "como programar";
l.autor = "Edilaine da Rocha";
l.assunto = " lógica de programação ";
l.num_pag = 147;

guarde (bib1, l);
feche (bib1);
```

```
// grava alguns registros em 'bib2'
abra (bib2);
l.titulo = "fundamentos de java";
l.autor = "Rômulo do Arantes";
l.assunto = "java";
l.num_pag = 245;
guarde (bib2, l);
avance (bib2);
l.titulo = "como programar em java";
l.autor = "Marcelo Ribas";
l.assunto = "java";
l.num_pag = 297;
guarde (bib2, l);
feche(bib2);

// abre 'bib'
abra(bib);

// copia os registros de 'bib1' para 'bib'
abra(bib1);
repita {
    copie (bib1, l);
    guarde (bib, l);
    avance(bib1);
    avance(bib);
} enquanto (nao da(bib1));
feche(bib1);

// copia os registros de 'bib2' para 'bib'
abra(bib2);
repita {
    copie (bib2, l);
    guarde (bib, l);
    avance (bib2);
    avance (bib);
} enquanto (nao da(bib2));
feche (bib2);

// posiciona no primeiro registro de 'bib'
posicione (bib, 1);

// imprime as informações dos livros cadastros
em 'bib'
repita {
    copie (bib, l);
    escreva (l.titulo, " - ", l.autor, " - ", l.assunto,
" - ", l.num_pag);
    avance (bib);
} enquanto (nao da (bib));
feche (bib);
}
```

8. Escreva um algoritmo que insere filmes em um catálogo. Um filme é composto por um nome e um ano. Todas as vezes em que o algoritmo for executado, essas informações devem ser definidas pelo usuário. Se o catálogo já tiver filmes cadastrados, o novo filme deve ser inserido no final.

```
// Declara o tipo filme
tipo filme {
    caractere: nome;
    inteiro: ano;
}

início {
    // Cria o arquivo e uma variável auxiliar
    arquivo: filmes;
    filme: aux;

    // Lê o nome e o ano do filme
    leia("Nome", aux.nome);
    leia("Ano", aux.ano);

    // Abre o arquivo
    abra(filmes);

    // Vai até o primeiro registro vazio no fim do
    arquivo
    repita {
        avance(filmes);
    } enquanto (nao da(filmes));

    // Grava os dados no arquivo
    guarde (filmes, aux);

    // Volta para o primeiro registro do arquivo
    posicione (filmes, 1);

    // Itera sobre todos os registros existente,
    imprimindo o nome e ano do filme
    repita {
        copie(filmes, aux);
        escreva(aux.nome, " - ", aux.ano);
        avance(filmes);
    } enquanto (nao da (filmes));

    // Fecha o arquivo
    feche(filmes);
}
```

9. Em um hotel, os hóspedes são classificados de acordo com a sua faixa etária. A classificação é feita da seguinte forma:

| Categoria | Faixa etária |
|------------------|---------------------|
| Recanto | 0 a 3 anos |
| Infantil | 4 a 11 anos |
| Juvenil | 12 a 17 anos |
| Adulto | 18 a 50 anos |
| Melhor idade | A partir de 51 anos |

Escreva um algoritmo que classifique um hóspede com base em uma idade fornecida. O algoritmo também deve verificar se a idade digitada é inválida (uma idade inválida é uma idade negativa ou maior de 130 anos).

```

início {
    // Define as variáveis
    inteiro: idade;
    caractere: categoria;

    // Solicita a idade
    leia("Idade do hóspede", idade);

    // Verifica a faixa etária. Se a idade for negativa ou superior a 130 anos, coloca em uma categoria
    "Inválida".
    se (idade < 0 ou idade > 130) {
        categoria = " Inválida ";
    } senão se (idade >= 0 e idade <= 3) {
        categoria = "Recanto";
    } senão se (idade >= 4 e idade <= 11) {
        categoria = "Infantil";
    } senão se (idade >= 12 e idade <= 17) {
        categoria = "Juvenil";
    } senão se (idade >= 18 e idade <= 50) {
        categoria = "Adulto";
    } senão {
        categoria = "Melhor Idade";
    }

    // Mostra o resultado. Se a categoria for " Inválida ", significa que a idade digitada foi incorreta.
    se (categoria <> " Inválida ") {
        escreva ("Categoria: ", categoria);
    } senão {
        Escreva ("A idade fornecida É Inválida ");
    }
}

```

10. Os números de Fibonacci são uma sequência numérica. Ela inicia com 0 e 1 e cada número seguinte da sequência é a soma dos dois números anteriores. Por exemplo: os 8 primeiros números da sequência são: 0, 1, 1, 2, 3, 5, 8, 13. Escreva um algoritmo que imprime os números da sequência de Fibonacci até um determinado valor (fornecido pelo usuário).

```
início {  
    // Declaração das variáveis  
    inteiro: max;  
    inteiro: i;  
    inteiro: a;  
  
    // Lê o valor máximo  
    leia("Valor máximo ", max);  
  
    // a inicia com valor 1  
    a = 1;  
  
    // Repete enquanto o valor máximo não for atingido  
    enquanto (i < max) {  
        escreva(i);  
        i = i + a;  
        a = i - a;  
    }  
}
```

11. Crie um algoritmo que lê a nota de cinco alunos e depois exibe as notas fornecidas, bem como a média entre elas. Utilize um vetor para armazenar as notas.


```
início {  
  
    // Define as variáveis  
    real: notas[5];  
  
    real: media;  
  
    inteiro: i;  
  
  
    // Inicia o valor da média como 0  
    media = 0;  
  
  
    // Lê todas as notas, e vai acumulando os valores na média  
    para i de 0 até 4 passo 1 {  
        leia("Nota", notas[i]);  
        media = media + notas[i];  
    }  
  
  
    // Mostra as notas digitadas  
    para i de 0 até 4 passo 1 {  
        escreva("Nota: ", notas[i]);  
    }  
  
  
    // Divide a soma das notas por 5 e atribuição própria variável  
    media = media / 5;  
  
  
    // Escreve a média final  
    escreva("média: ", media);  
}
```

Constantes, variáveis:

12. Escreva um algoritmo que solicita ao usuário duas informações no formato texto e depois exibe os dois textos fornecidos de forma concatenada.

Dica: Utilize o comando **concatene** para efetuar a concatenação dos valores. O comando **concatene** junta textos e produz outro como resultado. Por exemplo, para juntar os textos *para* e *quedas*, você pode fazer:

Caractere: texto;

Texto = concatene ("para", "quedas");

Você fornece os textos entre aspas e separados por vírgula (ou variáveis), e a variável texto terá o texto concatenado (neste caso, paraquedas).

```
Início {  
    // Declaração das variáveis  
    Caractere: t1;  
    Caractere: t2;  
    Caractere: result.;  
  
    // Leitura dos valores  
    Leia ("para", t1);  
    Leia ("quedas", t2);  
  
    // Concatenação dos valores  
    Result = concatene (t1, t2);  
  
    // Exibição do resultado  
    Escreva (result);  
}
```

Operadores:

13. Determine se o resultado obtido na avaliação das expressões lógicas é verdadeiro ou falso, sabendo que A, B e C contêm, respectivamente, 2, 7, 3 e que existe uma variável lógica L cujo valor é falso (F):

- a) $7 = 2 * 3$ - FALSO
- b) $7 > 2$ ou $7 = 2 * 2$ - FALSO
- c) Falso = $2 > 3$ - VERDADEIRO
- d) $3 + 7 = (2 + 2 + 2 + 2 + 2)$
- e) Não Falso = $7 * 3 < 2 * 3$ - VERDADEIRO

14. Suponha que as variáveis A, B e C são do tipo inteiro, com os valores iguais a 5, 8 e 10. Quais são os resultados das expressões aritméticas a seguir?

- 1. $2 + 5 - (10 / 2) + 8 * 2 = 7 - 5 + 16 = 18$
- 2. $[(5 + 8 + 10) * 2] + 4 / 5 = 10$
- 3. $(5 + 10) * 8 - (8 + 5) * 10 = -10$
- 4. $5 + 5 + 5 + 5 / 10 = 2$
- 5. $[(8 / (10 - 8)) + (5 * 5)] + [((10 / 2) + (8 - 8)) * 3] - 44 = 0$

15. Encontre os erros dos seguintes comandos de atribuição:

Lógico: A

Inteiro: B, C

Real: D, E

Caractere: F, G

| | |
|----------------------------|---|
| a) $A <- B$ | Resposta: Uma variável do tipo lógico não recebe variável do tipo inteiro. |
| b) $F <- G$ | Resposta: Uma variável do tipo caractere não recebe variável do tipo lógico. |
| c) $C + 1 <- D$ | Resposta: Uma variável não pode realizar operação antes da atribuição. A operação deve ser executada à direita. |
| d) $D \text{ e } E <- 3,5$ | Resposta: Duas variáveis, assim como uma operação lógica não podem estar a esquerda da atribuição. Apenas uma variável pode receber um dado ou valor. |
| e) $C <- E$ | Resposta: Uma variável do tipo inteiro não pode receber uma variável do tipo real. |

16. Uma equação do segundo grau é definida da seguinte forma:

$$ax^2 + bx + c = 0$$

O cálculo das raízes de uma equação do segundo grau é dado pela fórmula de Bhaskara, definida da seguinte forma:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Escreva um algoritmo que encontre as raízes de uma equação do segundo grau a partir dos valores a, b e c fornecidos pelo usuário.

Dica: Ao executar o algoritmo, fique atento aos valores fornecidos para a, b e c. Lembre-se de que o resultado da raiz quadrada de um número negativo não pertence ao conjunto dos números reais, fazendo com que a resposta mostrada seja **[Número Imaginário]**.

```
Início {  
    // Variáveis a, b e c  
    Real: a;  
    Real: b;  
    Real: c;  
  
    // Raízes  
    Real: x1;  
    Real: x2;  
  
    // Leitura dos valores  
    Leia("A", a);  
    Leia("B", b);  
    Leia("C", c);  
  
    // Raiz 1  
     $x1 = (-b + \text{rad}(\text{pot}(b, 2) - 4 * a * c)) / (2 * a);$   
  
    // Raiz 2  
     $x2 = (-b - \text{rad}(\text{pot}(b, 2) - 4 * a * c)) / (2 * a);$   
  
    // Exibição do resultado  
    Escreva ("Raiz 1: ", x1);  
    Escreva ("Raiz 2: ", x2);  
}
```