

Lógica de Programação

CSTSI CEFET-RS

Como resolver o problema abaixo?

PROBLEMA:

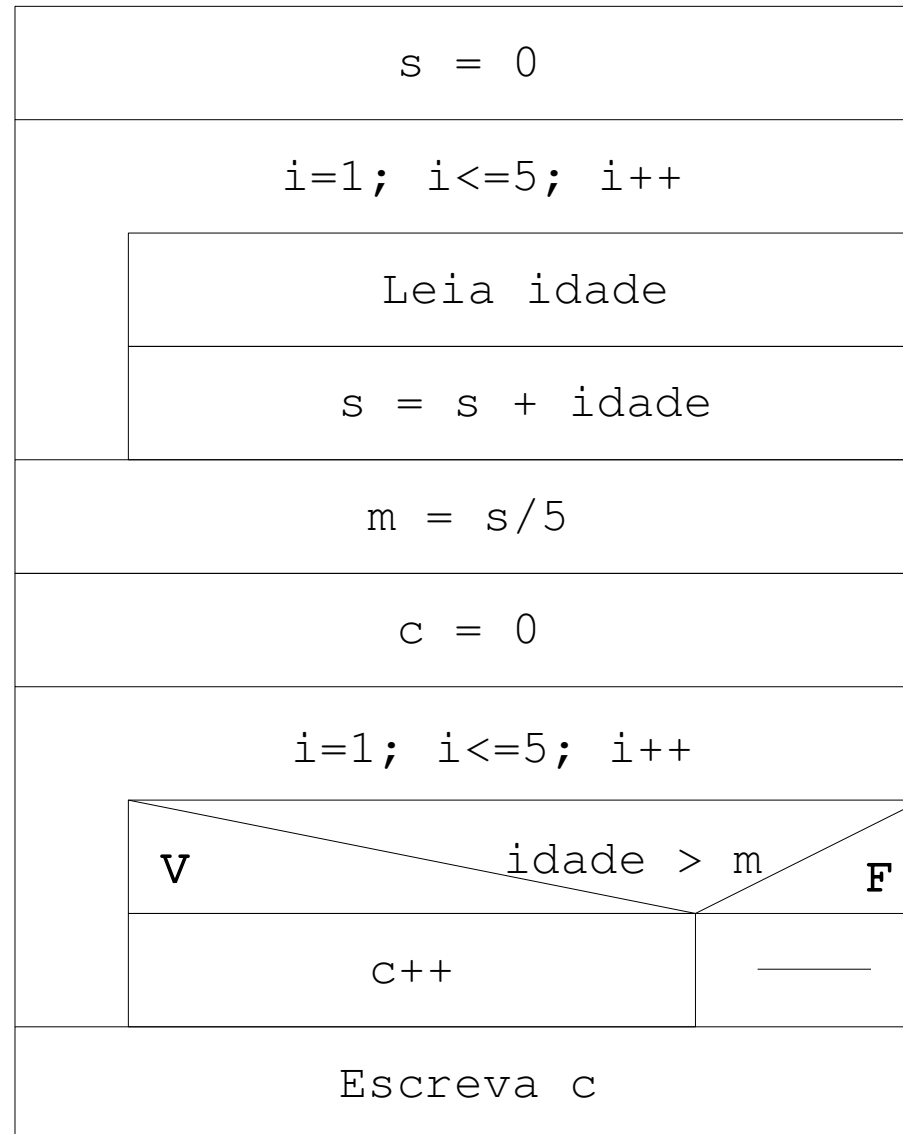
a) Escreva um algoritmo para ler a idade de cada componente de um grupo de 5 alunos. Contar e escrever quantos alunos possuem idade acima da média de idade do grupo.

[Entrada]	[Saída]
21	
10	
9	
17	
18	3

A média de idade do grupo é 15

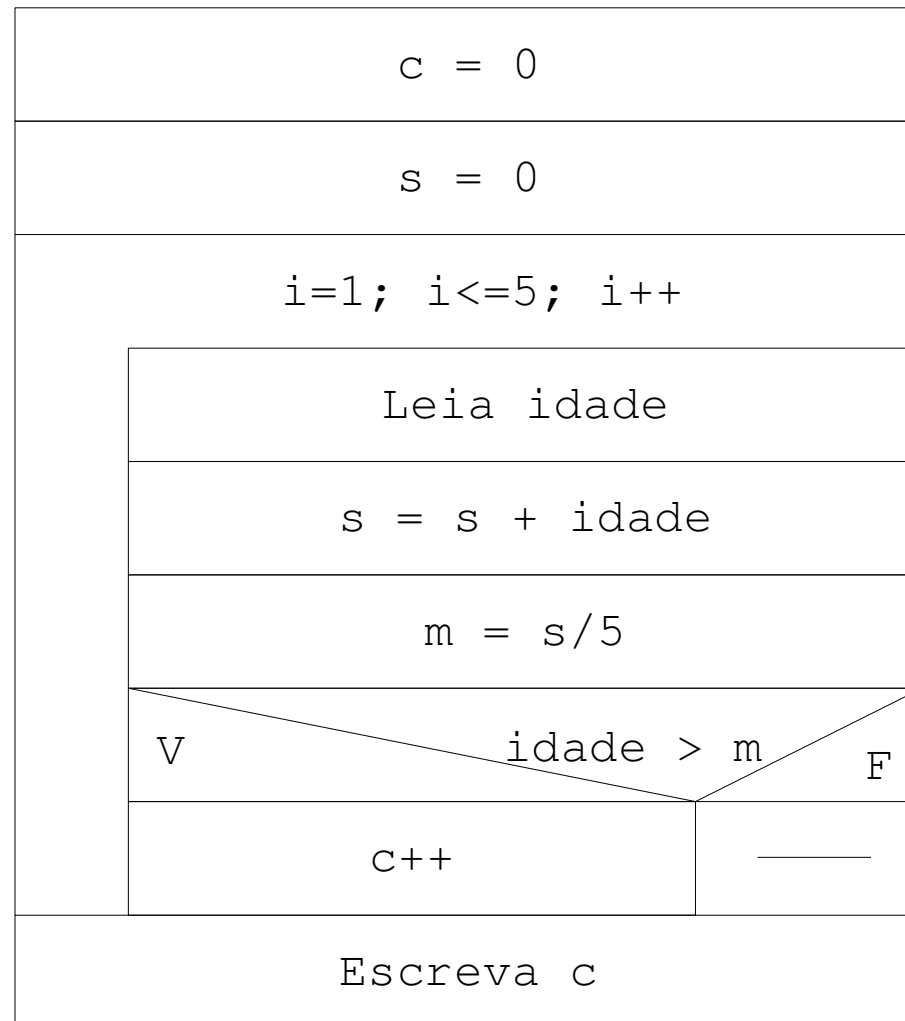
Solução incorreta 1

Por que está incorreta?



Solução incorreta 2

Por que está incorreta?



Solução 3

Correta, mas inadequada

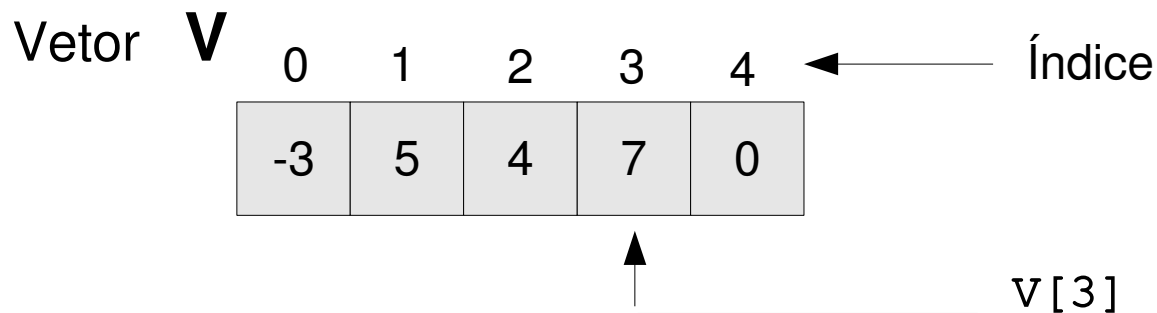
ct = 0	
Leia a,b,c,d,e	
m = (a+b+c+d+e)/5	
a > m	
ct++	_____
b > m	
ct++	_____
c > m	
ct++	_____
d > m	
ct++	_____
e > m	
ct++	_____
Escreva ct	

Variáveis indexadas

É um conjunto de variáveis do **mesmo tipo** que compartilham um **mesmo nome**. Índices são associados a este nome com a finalidade de permitir a individualização dos elementos do conjunto.

Quando possuí apenas um índice (uma dimensão) chamamos de matriz unidimensional ou vetor.

Ex: Um vetor V de 5 elementos



Variáveis indexadas

Em “C” um vetor deve ser declarado com a especificação da quantidade de elementos

<code>v[0] = -3</code>
<code>v[1] = 5</code>
<code>v[2] = 4</code>
<code>v[3] = 7</code>
<code>v[4] = 0</code>
<code>Escreva v[3]</code>
<code>a = 1</code>
<code>Escreva v[a]</code>

Podemos colocar uma variável como índice

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
    int a, v[5];
```

Qde de
elementos

```
    v[0] = -3;
    v[1] = 5;
    v[2] = 4;
    v[3] = 7;
    v[4] = 0;
    printf("%d\n", v[3]);
    a=1;
    printf("%d\n", v[a]);
    system("pause");
    return 0;
}
```

Variáveis indexadas

O índice deve ser um **inteiro** não negativo.
Pode-se colocar uma **constante**, **variável** ou **expressão** no lugar de um índice.

<code>v[0] = -3</code>
<code>v[1] = 5</code>
<code>v[2] = 4</code>
<code>v[3] = 7</code>
<code>v[4] = 0</code>
<code>a = 1</code>
<code>Escreva v[a+1]</code>
<code>x = v[a]+1</code>
<code>Escreva x</code>

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a,x,v[5];

    v[0] = -3;
    v[1] = 5;
    v[2] = 4;
    v[3] = 7;
    v[4] = 0;
    a = 1;
    printf("%d\n",v[a+1]);
    x = v[a]+1;
    printf("%d\n",x);
    system("pause");
    return 0;
}
```


Variáveis indexadas

O elemento de um vetor também pode ser um índice, desde que seja um **inteiro** com um valor **válido** para o índice.

<code>v[0] = -3</code>
<code>v[1] = 5</code>
<code>v[2] = 4</code>
<code>v[3] = 7</code>
<code>v[4] = 0</code>
<code>Escreva v[v[4]]</code>
<code>Escreva v[v[4]+1]</code>
<code>Escreva v[v[v[2]]]</code>

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int v[5];

    v[0] = -3;
    v[1] = 5;
    v[2] = 4;
    v[3] = 7;
    v[4] = 0;
    printf("%d\n", v[v[4]]);
    printf("%d\n", v[v[4]+1]);
    printf("%d\n", v[v[v[2]]]);
    system("pause");
    return 0;
}
```

Variáveis indexadas

<code>v[0] = -3</code>
<code>v[1] = 5</code>
<code>v[2] = 4</code>
<code>v[3] = 7</code>
<code>v[4] = 0</code>
<code>i=0; i<=4; i++</code>
<code>Escreva v[i]</code>

```
#include <stdio.h>
#include <stdlib.h>

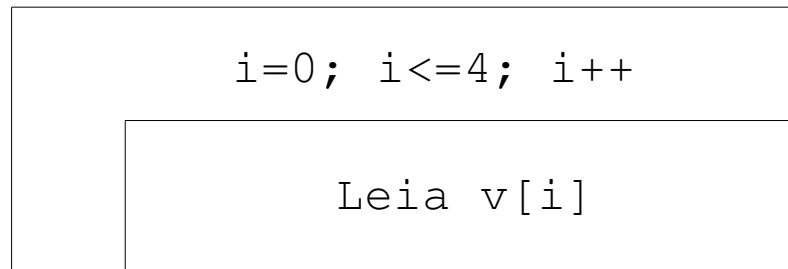
int main()
{
    int v[5], i;

    v[0] = -3;
    v[1] = 5;
    v[2] = 4;
    v[3] = 7;
    v[4] = 0;
    for (i=0; i<=4; i++)
        printf("%d\n", v[i]);
    system("pause");
    return 0;
}
```

Leitura e escrita de um vetor

A leitura do vetor é executada de forma que cada elemento seja lido (ou escrito) individualmente.

Exemplo de leitura de um vetor de 5 elementos



```
...  
for (i=0; i<=4; i++) {  
    printf("Informe o elemento %d:", i);  
    scanf("%d", &v[i]);  
}  
...
```

Observações sobre variáveis indexadas

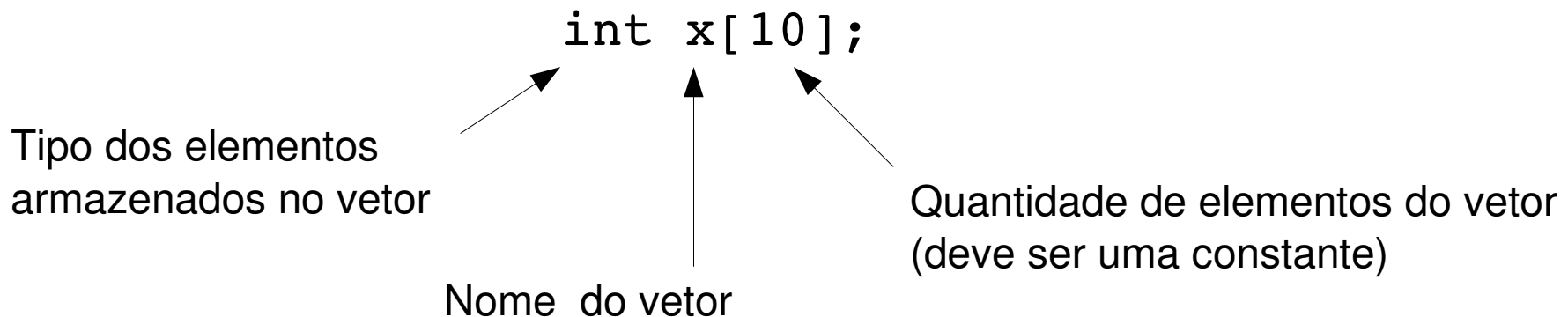
- O índice deve ser um **inteiro** (constante, variável ou expressão).
- O **primeiro** elemento do vetor possui **índice 0**.
- Qualquer operação definida para uma variável simples é válida para um componente isolado de um vetor.

```
...  
for (i=0; i<=4; i++)  
do  
{  
    printf("Informe um número positivo:");  
    scanf("%d",&num[i]);  
}  
while (num[i]<=0);  
...
```

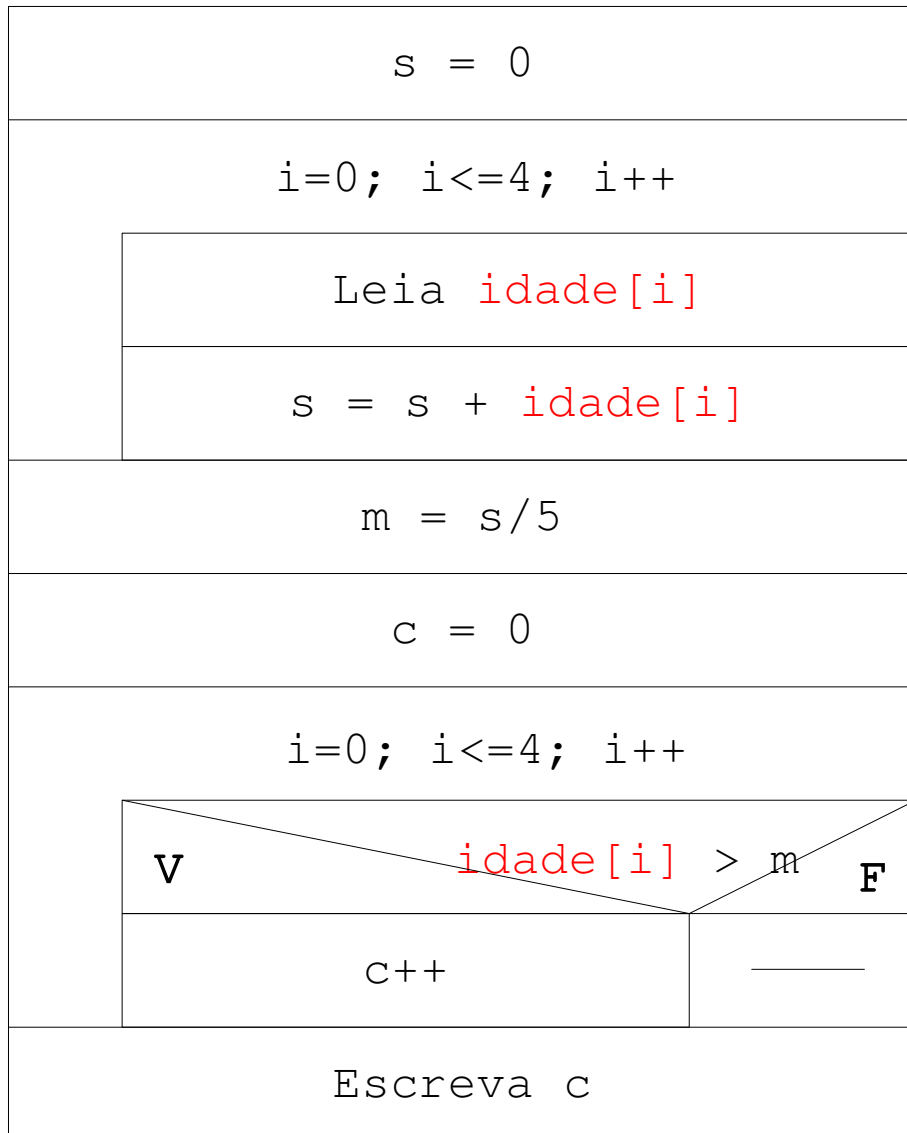
- Não é possível operar diretamente com o conjunto completo.

Observações sobre variáveis indexadas

- Ao acessar os elementos de um vetor não utilizar índices fora da faixa.
- Um vetor pode ser declarado para armazenar valores de qualquer tipo. Ex: `float vet[100];`
- Todos os elementos do vetor são do mesmo tipo.
- A constante definida na declaração indica a quantidade máxima de elementos de um vetor.



Resolvendo o problema proposto



```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int idade[5], i, s, c;
    float m;

    s = 0;
    for (i=0; i<=4; i++) {
        printf("Informe a idade %d:", i);
        scanf("%d", &idade[i]);
        s = s + idade[i];
    }

    m = (float) s/5;
    c = 0;
    for (i=0; i<=4; i++)
        if (idade[i]>m)
            c++;
    printf("%d\n", c);
    system("pause");
    return 0;
}
```