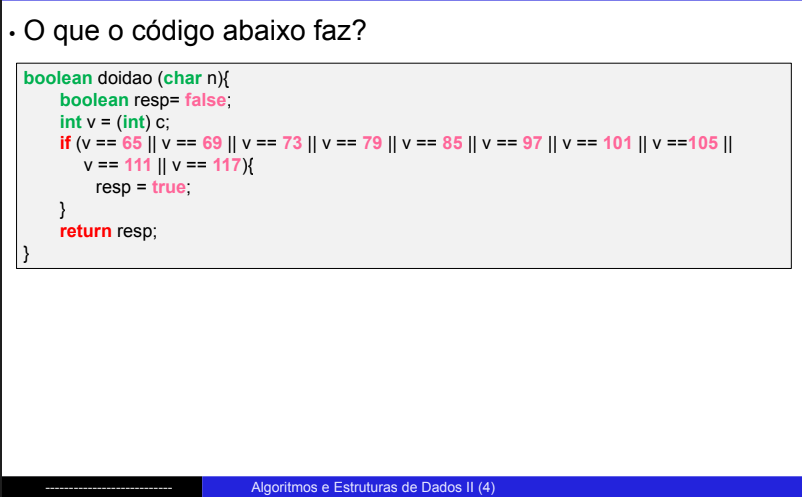
**Exercícios Unidade00b:**

**1.**



**Obs:**

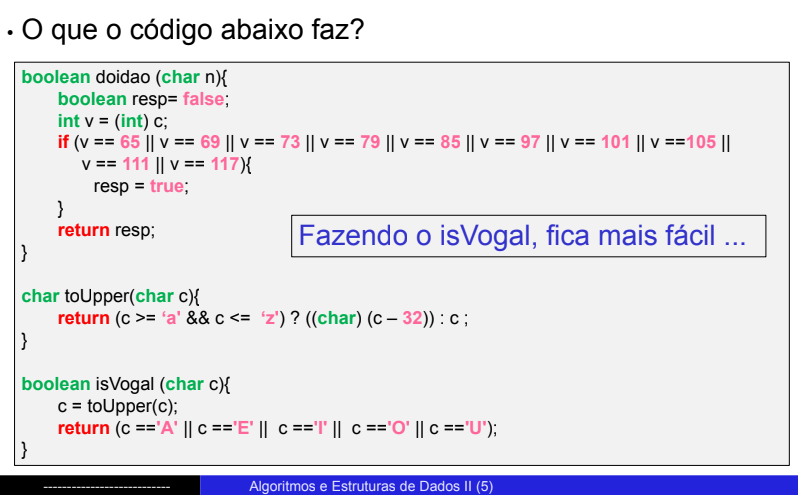
|  |
| --- |
| **boolean** doidao (**char** n) |
| (char n) deveria ser (char c) |

***Resposta***:

a função ***doidao()*** recebe como parâmetro um caractere que logo em seguida se transforma, por meio de um typecast em sua numeração correspondente. Em seguida ocorre uma verificação deste numero do caráter com outros numerais.

Função retorna ***FALSE*** se numeração do caractere não corresponde as do exemplo, retorna ***TRUE*** se corresponde.

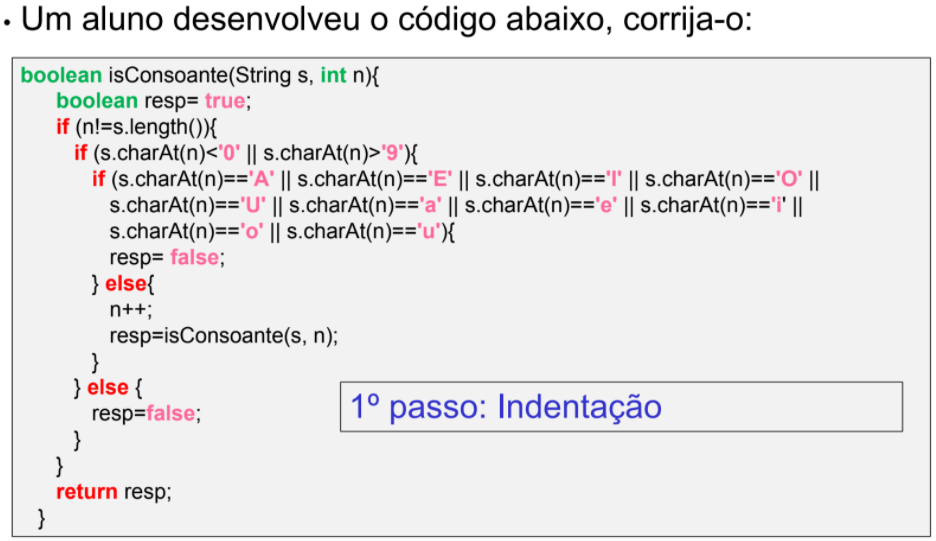
**2.**

****

***Resposta***:

* toUpper()
* Recebe como parâmetro um caráter.
* isVogal()
* Recebe como parâmetro um caráter.
* Transforma variável que possui caráter em maiúsculo.
* Retorna **TRUE** se caractere for igual a {A, E, I, O ou U}.
* Retorna **FALSE** se caractere não for igual a {A, E, I, O ou U}.

**3.**



***Resposta***:

boolean isConsoante(String str, int i){

boolean resp = true;

if(n == str.length()){

resp = true;

} else if(!isConsoante(str.charAt(i))){

resp = false;

} else{

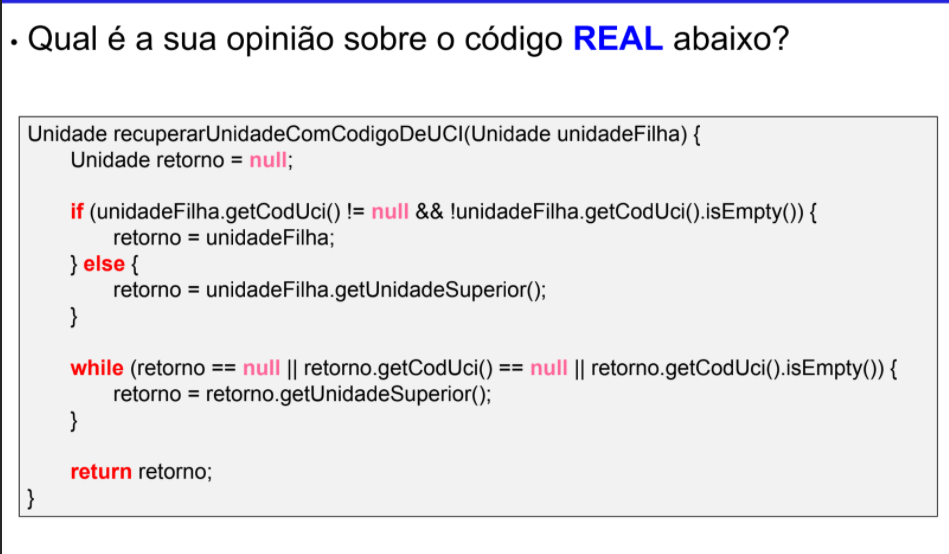
resp = isConsoante(str, i+1);

}

return resp;

} // end isConsoante()

**4.**

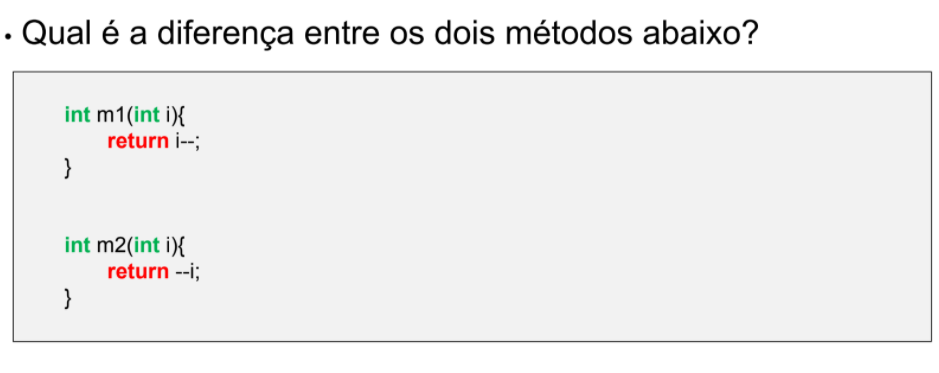


***Resposta***:

1- O nome da função é muito grande, ao invés desse tamanho poderia ter um comentaria explicando o que a função faz.

2- O código da muita volta. poderia haver uma simplificação do código.

**5.**

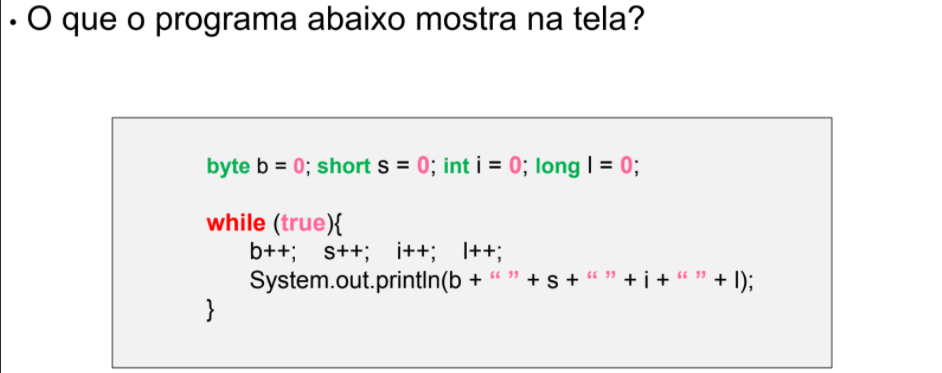


***Resposta***:

i-- retornar o valor antes de decrementado.

--i retorna o valor depois de decrementado.

**6.**

****

***Resposta***:

O programa acima mostra cada variável sendo incrementada +1 a cada loop.

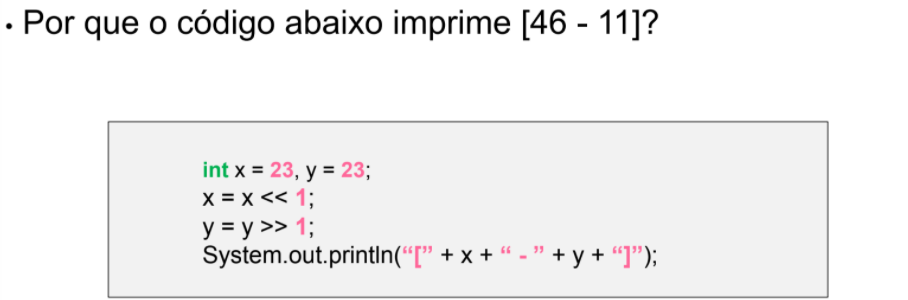
Variável **b** poderá ser incrementada no máximo 28 vezes

Variável **s** poderá ser incrementada no máximo 216 vezes

Variável **i** poderá ser incrementada no máximo 232 vezes

Variável **l** poderá ser incrementada no máximo 264 vezes

**7.**



***Resposta***:

1. 2310 = 101112
2. 101112 << 1011102 = 4610    (Shift left <<, desloca bits para esquerda adicionando um 0 na posição 0)
3. 101112 >> 010112   = 1110    (Shift right >>, desloca bits para direita, removendo o bit da posição 0)