

Questão 02: (2,0pts – 0,5cd)

Abrir com ▼

Analise o algoritmo abaixo e responda às perguntas seguintes.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. int main()
4. {
5.     int a, b, c, d;
6.     printf("Digite dois números inteiros: ");
7.     scanf("%d%d", &a, &b);
8.     c=1;
9.     d=1;
10.    while(d<=b)
11.    {
12.        c = c * a;
13.        d = d + 1;
14.    }
15.    printf("%d\n", c);
16.    return 0;
17. }
```

WHILE

1° VEZ

1<=4 ?? V
1 = 1*3 LOGO C = 3
1 = 1+1 LOGO D = 2

2<=4 ?? V **2° VEZ**

3 = 3*3 LOGO C = 9
2 = 2+1 LOGO D = 3

3<=4 ?? V **3° VEZ**

9 = 9*3 LOGO C = 27
3 = 3+1 LOGO D = 4

4<=4 ?? V **4° VEZ**

27 = 27*3 LOGO C = 81
4 = 4+1 LOGO D = 5

5<=4 ?? F **5° VEZ**

**Logo a estrutura de
repetição encerra e
printamos o valor de c.**

- Qual a finalidade do programa acima?
- Se entrarmos com os valores 3 e 4 como dados de entrada, o que o algoritmo apresenta como resultado? Justifique sua resposta.
- Qual é a condição de parada da estrutura de repetição descrita na linha 10?
- O que representam as variáveis a, b, c e d no algoritmo acima?

Questão 04: (2,0pts - 0,4cd)

Abrir com

Análise o algoritmo abaixo e responda às perguntas seguintes.

```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.  int func1 (int a, int b)
4.  {
5.      int c;
6.      c = a * b;
7.      return c;
8.  }
9.  int main()
10. {
11.     int n, i, j, x;
12.     printf("Digite um numero inteiro: ");
13.     scanf("%d", &n);
14.     for(i=1; i<=n; i++)
15.     {
16.         for (j=1; j<=10; j++)
17.         {
18.             x = func1(j,i);
19.             printf("%d * %d = %d\n", j,i,x);
20.         }
21.         printf("\n");
22.     }
23.     return 0;
24. }
```

- b) Se entrarmos com o valor 3 como dado de entrada, o que o algoritmo apresenta como resultado? Justifique sua resposta. **o algoritmo irá mostrar a taboada de 1 até 3. inicialmente vamos entrar no primeiro "for", visto que "i" inicialmente irá valer 1 e é menor que 3, então passamos ao segundo for, onde inicializamos a variável "j" valendo 1, o algoritmo irá testar a condição $j \leq 10$ e será verdadeira, e logo atribuímos o valor da função ($j \cdot i$) para dentro da variável "x" ($1 \cdot 1 = 1$, logo $x = 1$), logo após a atribuição vamos printar " 1*1 = 1". O Algoritmo irá repetir todo o processo até que o "J" seja igual a 11 e o programa tenha imprimido a taboada inteira de 1, após isso J valendo 11 a condição do segundo for irá ser falsa e retornaremos para o primeiro for onde o "i" irá passar a valer 2, e todo este processo irá se repetir até que cheguemos a taboada de 3.**