

1. (adaptado de CESGRANRIO 2010 - EPE) Observe o fragmento de código abaixo desenvolvido em Linguagem C:

x=3  
y=4  
z=5  
como (x-1=2) não é maior que 2  
então decrementa y  
y=3  
z=x+y => 3+3=>6  
no laço for o valor de y será  
incrementado 8x, portanto seu  
valor final será y=11  
z=z+y => 6+11=>17

```
...
x=3;
y=4;
z=5;
if((x-1)>2)
    y++;
else
    y--;
z=x+y;
for(i=1;i<9;i++)
    y++;
z=z+y;
...
```

Após a sua execução, qual será o valor de z?

- (a) 12  
(b) 15  
(c) 16  
(d) 17  
(e) 20
2. Assuma o seguinte trecho de programa e considere os endereços de memória na Tabela 1:

```
...
int a=10, b=20, c;
double x=1.50;
int* P1;
int* P2;
double* P3;
P1=&a;
P2=&b;
P3=&x;
c=*P1+*P2;
...
```

Tabela 1	
Variável	Endereço de Memória
a	24FF2
b	2F341
c	1FF23
x	55FFF
P1	34EFE
P2	21EFEF
P3	3FE567

Marque V (verdadeiro) ou F (falso) para as seguintes expressões:

- |                 |                     |
|-----------------|---------------------|
| ( F ) b == &a   | ( V ) *P3 == 1.50   |
| ( F ) P1 == a   | ( F ) P2 == P1      |
| ( F ) a == &P1  | ( F ) c == 21.50    |
| ( F ) *P1 == &a | ( V ) &P2 == 21EFEF |
| ( F ) &P1 == &a | ( F ) a == 24FF2    |
| ( V ) *P2 == b  | ( V ) c == a + b    |

3. (adaptado de ENADE 2011) Considerando a execução do trecho de programa abaixo, responda ao que se pede nos itens a e b.

```
1. ...
2. const int T = 10;
3. int[] vetA = new int[T];
4. int[] vetB = new int[T];
5. int i;
6. for(i=0;i<T;i++)
7. {
8.     vetB[i] = 0;
9.     if(i%2 == 0)
```

```

10.         vetA[i] = i;
11.     else
12.         vetA[i] = 2*i;
13.     }
14.     for(i=0;i<T;i++)
15.     {
16.         while(vetA[i] > i)
17.         {
18.             vetB[i] = vetA[i];
19.             vetA[i] = vetA[i] - 1;
20.         }
21.     }
22. ...

```

Algoritmo 1

a) Apresente os dados dos vetores **vetA** e **vetB** ao término da execução da linha 13.

**vetA** = [0,2,2,6,4,10,6,14,8,18]

**vetB** = [0,0,0,0,0,0,0,0,0,0]

b) Apresente os dados dos vetores **vetA** e **vetB** ao término da execução da linha 21.

**vetA** = [0,1,2,3,4,5,6,7,8,9]

**vetB** = [0,2,0,4,0,6,0,8,0,10]