

Exemplo de prova.

Eu faço uma prova nova todos os anos, então não espere que as próximas serão parecidas com essa.

Nome (legível):

Assinatura:

- 1) (1,5 Ponto) Simule a execução do programa abaixo, mostrando o que será impresso na tela na coluna da direita após o printf de “Exemplo”.

#include <stdio.h>	Exemplo
<pre> void set(int *i, int v) { *i = v; } void soma(int *a, int v) { *a = *a + v; } void main() { int i = 1, h = 3, n = 0, r = 0, *p; int v[5]; printf("Exemplo\n"); v[0] = 5; v[1] = 4; v[2] = 3; v[3] = 2; v[4] = 1; set(&i, 0); set(&h, 0); p = &h; while(i < 3) { printf("%d\n", h); soma(p, i); i++; } n = r + i; printf("n = %d\n", n); printf("H = %d\n", *p); set(&v[0], 20); p = &v[0]; *p = 90; set(&r, n + 2); set(&n, n * n); printf("v[0] = %d\n", *p); printf("r = %d\n", r); printf("n = %d\n", n); printf("P = %d\n", *(p + 1)); p++; printf("P = %d\n", *(p + 1)); } </pre>	<pre> 0 0 1 n = 3 H = 3 v[0] = 90 r = 5 n = 9 P = 4 P = 3 </pre>

--	--

2) (2 Pontos) Qual a complexidade no pior caso do seguinte algoritmo? Justifique. **$O(n)$**

```
int fun(int n){
    int count = 0, i, j;
    for (i = n; i > 0; i /= 2)
        for (j = 0; j < i; j++)
            count += 1;
    return count;
}
```

3) (2 Pontos) Assumindo n grandes, existe uma situação em que o algoritmo de ordenação “inserção” tem melhor desempenho (custo computacional) que o “quick-sort”? Justifique. Tirando essa situação em todas as demais o “quick-sort” é sempre melhor que o “inserção”? Justifique.

Sim, quando a entrada é ordenada.

Não, no pior caso ambos são iguais.

4) (1,5 Pontos) Dado o seguinte vetor que já passou pela primeira partição do quick sort, identifique o(s) pivô(s). Justifique.

2	5	1	3	7	9	12	11	10
---	---	---	---	---	---	----	----	----

5) (1,5 Pontos) Dado o seguinte vetor mostre passo-a-passo a ordenação utilizando o algoritmo “seleção”:

5	15	20	17	18	30	3	29
3	15	20	17	18	30	5	29
3	5	20	17	18	30	15	29
3	5	15	17	18	30	20	29
3	5	15	17	18	20	30	29
3	5	15	17	18	20	29	30

6) (1,5 Pontos) A operação de destruição de uma árvore requer um tipo de percurso em que a liberação de um nó é realizada apenas após todos os seus descendentes terem sido também liberados. Segundo essa descrição, a operação de destruição de uma árvore deve ser implementada utilizando um dos 3 tipos de percursos vistos em aula. Diga qual e mostre em que ordem os nós seriam apagados na seguinte árvore caso fossemos apagar toda a árvore. (pós-ordem: 2,6,8,4,16,12)

