

1. Seja o programa a seguir. Determine o valor do ponteiro `y` no programa:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int r, *y, s;
6
7  int main()
8  {
9      r = 5;
10     s = r + 2;
11     y = &s;
12     *y = ++(*y) + (s)++ + (r)++;
13     cout << *y << endl;
14     return 0;
15 }
16
```

2. Seja o programa a seguir. Determine o valor da variável `b`.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int b, *a, *c;
6
7  int main()
8  {
9      b = 10;
10     a = &b;
11     c = a;
12     *a = *c + b++;
13     b = (*a)++ + ++(*c);
14     cout << b << endl;
15     return 0;
16 }
```

3. Determine a ordem assintótica do (pior caso) algoritmo a seguir:

```
1     int achou = -1;
2     for (int i=0; i < n; i++) {
3         if (v[i] == alvo){
4             achou=i;
5             break;
6         }
7     }
```

4. Para o mesmo algoritmo acima, determine o melhor caso.

5. Ordene em ordem decrescente as ordens assintóticas a seguir:

- $O(n \log n)$
- $O(1)$
- $O(n^2)$

- $O(n)$

6. Determine a ordem assintótica do (pior caso) algoritmo a seguir:

```
1      for (int i=0; i < n; i++) {
2          for (int j=0; j < n; j++) {
3              if (m[i][j] = alvo)
4                  return m[i][j];
5          }
6      }
```

7. Determine a ordem assintótica do (pior caso) algoritmo a seguir:

```
1      for (int i=n-1; i > 0; i++) {
2          for (j=0;j<i;j++) {
3              if (v[j] > v[j+1]) {
4                  int temp = v[j];
5                  v[j] = v[j+1];
6                  v[j+1] = temp;
7              }
8          }
9      }
```

8. Para o algoritmo recursivo a seguir, determine o valor da chamada da função na segunda iteração.

```
1      int soma(int a, int b){
2          if (a == b)
3              return a;
4          else
5              return a + soma(a,b-1);
6      }
```

a) Para os valores de 2 até 4

9. Para o algoritmo recursivo a seguir, determine o valor da chamada da função na segunda iteração.

```
1      int multiplicacao(int a, int b){
2          if (a == b)
3              return a;
4          else
5              return (1+a) * multiplicacao(a,b-1);
6      }
```

a) Para os valores de 1 até 3