

# Notes LP

Fernanda Guimarães

## 1 1

```
int a = 7; // static
const int b = 9; // static
const int c = std::rand(); // static
void foo() {
    int d = 9; // stack
    int e = 9; // stack
    const int f = std::rand(); // stack
    static int g = 7; // static
    static const int h = 9; // static
    static const int i = std::rand(); // static
    int* j = (int*) malloc (sizeof(int)); // heap
}
```

## 2 2

## 3 3

1. Lisp, John McCarthy em 59.
2. Caso o programador queira usar um ponteiro que já foi desalocado, comportamento inesperado.
3.
  - Mark and sweep: find the live heap links and mark those blocks that are reachable. Then make a pass over the heap and return unmarked free blocks to the free pool.
  - Copying collection: memory is divided in two; only one half is used at a time. When the used half is full, copy used blocks to the other location, and erase the old one.
  - Reference counting: each block has a counter of links that point to it. This counter is incremented when a heap link is copied, decremented when the link is discarded. When the counter goes to zero, the block is freed.

## 4 4

1. Paramétrico e subtipagem.
2. 14.cpp:15:6: error: no member named 'print' in 'std::\_cxx11::basic\_string<char>'.
3. ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0): o catch pegou a exceção, não gerando erros.
4. ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0): o catch pegou a exceção.

## 5 5

1. Cópia e coleta: tempo previsível e sistema com tempo definido.
2. Macação e varredura: possui um baixo custo de tempo e oferta de memória costuma ser maior.