

INE5416 - Paradigmas da Programação (2015/2)

Relatório 8: Listas e Array Caique Rodrigues Marques 13204303

Parte 1

- Na linguagem de programação C, os arrays podem ser declarados usando colchetes [], como por exemplo `int array[]`, ou usando a notação de ponteiros, como `int *array`. Por conta da tipagem, C não permite que um array contenha elementos de diferentes tipos, portanto, é inviável algo como `{1, 2, 3, 'a', '0'}`. A inicialização dos arrays pode ser de forma dinâmica (com o uso de `malloc`, `calloc` ou `alloca`) ou com o uso de chaves compondo os elementos presentes no array, como por exemplo `{1,2,3}`. O conceito de listas pode variar dependendo da linguagem, em C não há implementações nativas quanto a listas, mas há suporte para estruturas de dados como listas ligadas e listas circulares.
- Em Python há uma alocação mais dinâmica de listas, que pode ser declarado simplesmente como `list = []` e, como não há preocupação com tipagens na linguagem, pode existir listas com elementos de diferentes tipos, como `[1, 2, 'a', 'c']`. Inclusive, em Python é possível realizar operações dentro de colchetes para listas, algo como `[i+1 for i in range(10)]`, onde gerará uma lista com elementos de 1 a 10, isto é chamado de *list comprehension* (baseado na implementação existente em Haskell que é semelhante à notação de definição de conjuntos na matemática). Arrays em Python é um pouco diferente, sendo que ela é limitada quanto à tipagem de seus elementos.
- Em C, arrays passados como parâmetros apontam para a posição de memória onde tal elemento está. Em Python, os elementos são modificados quando passados por parâmetros, para evitar inconsistências, cópias são criadas.

Parte 2

- `relatorio8.c`
Foi usada a estratégia de VLA (*variable-length array*) para evitar o uso de alocação dinâmica, outra facilidade foi a especificação de um macro para definir uma função de tamanho.
Nota: Para compilação, é necessário especificar biblioteca math: `gcc relatorio8.c -lm -std=c11`
- `relatorio8.py`
Nota-se o aumento da complexidade espacial, em compensação é mais fácil de manipular as estruturas. Foi usada uma alocação estática para as operações.