ACH2001 - Introdução à Programação

Caracteres (*chars*) e *strings*

Prof. Flávio Luiz Coutinho
flcoutinho@usp.br

Caracteres

Assim como os tipos `int`, `long`, `float`, `double` (que representam valores numéricos), o tipo `char` representa símbolos que podem ser:

- Letras: ‘a’, ‘b’, ‘c’, …, ‘z’, ‘A’, ‘B’, ‘C’, …, ‘Z’, versões acentuadas das letras
- Dígitos numéricos: ‘0’, ‘1’, ‘2’, ‘3’, …, ‘9’
- Caracteres de pontuação e outros símbolos: ‘.’, ‘!’, ‘?’, ‘:’, ‘;’, ‘#’, ‘(‘, ‘)’, etc.
- Caracteres invisíveis e de controle (tab, quebra de linha, etc).

Caracteres

Na realidade, uma variável do tipo `char` também guarda um valor numérico (na prática é uma variável `int` com menor capacidade). Mas este valor numérico é usualmente interpretado como o código de um símbolo.

No C, uma variável do tipo `char` ocupa tipicamente 1 byte (ou 8 bits), o que permite representar até 256 caracteres diferentes. A codificação ASCII usa 1 byte para representar cada caractere, porém codificações mais modernas como a UTF-8 pode empregar mais de um byte.

Caracteres

Alguns exemplos de caracteres e seus códigos segundo a codificação ASCII:

- ‘0’: 48
- ‘1’: 49
- ‘9’: 57
- ‘A’: 65
- ‘B’: 66
- ‘C’: 67
- ‘Z’: 90
- ‘a’: 97
- ‘b’: 98
- ‘c’: 99
- ‘z’: 122

Caracteres

Exemplos envolvendo o tipo **`char`**.

Strings

Na linguagem C, strings (cadeias de caracteres) nada mais são do que vetores (*arrays*) de chars.

Strings

Na linguagem C, strings (cadeias de caracteres) nada mais são do que vetores (*arrays*) de chars.

A única característica que diferencia uma string de um vetor de qualquer outro tipo, é o fato de que uma string é sempre terminada por um caractere nulo (um caractere vazio, cujo código é zero).

Strings

Na linguagem C, strings (cadeias de caracteres) nada mais são do que vetores (*arrays*) de chars.

A única característica que diferencia uma string de um vetor de qualquer outro tipo, é o fato de que uma string é sempre terminada por um caractere nulo (um caractere vazio, cujo código é zero).

O caractere nulo sinaliza o fim da cadeia de caracteres que representam algum tipo de informação (note que o espaço de memória reservado para o vetor, pode ser maior que o tamanho efetivo da string).

Strings

Na linguagem C, strings (cadeias de caracteres) nada mais são do que vetores (*arrays*) de chars.

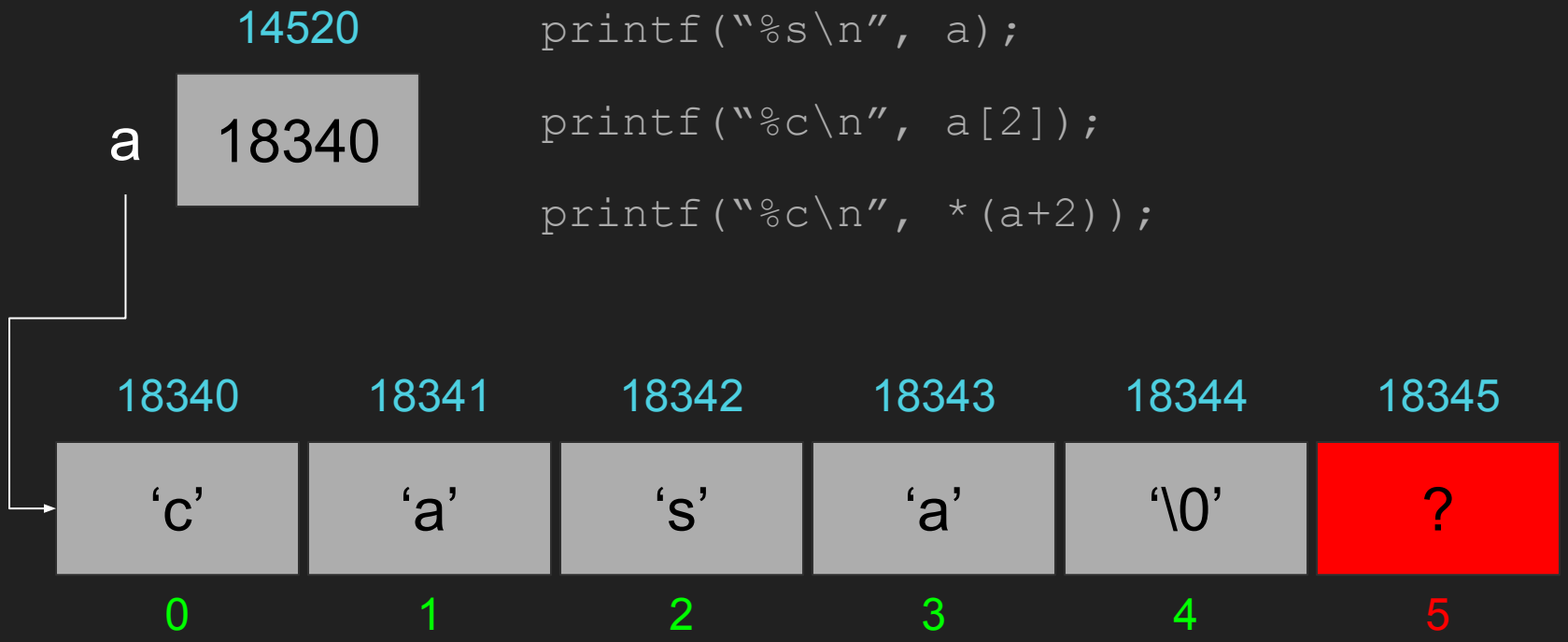
A única característica que diferencia uma string de um vetor de qualquer outro tipo, é o fato de que uma string é sempre terminada por um caractere nulo (um caractere vazio, cujo código é zero).

O caractere nulo sinaliza o fim da cadeia de caracteres que representam algum tipo de informação (note que o espaço de memória reservado para o vetor, pode ser maior que o tamanho efetivo da string).

Devido a existência do caractere nulo, uma string de n caracteres precisa de um vetor de chars com ao menos (n + 1) posições.

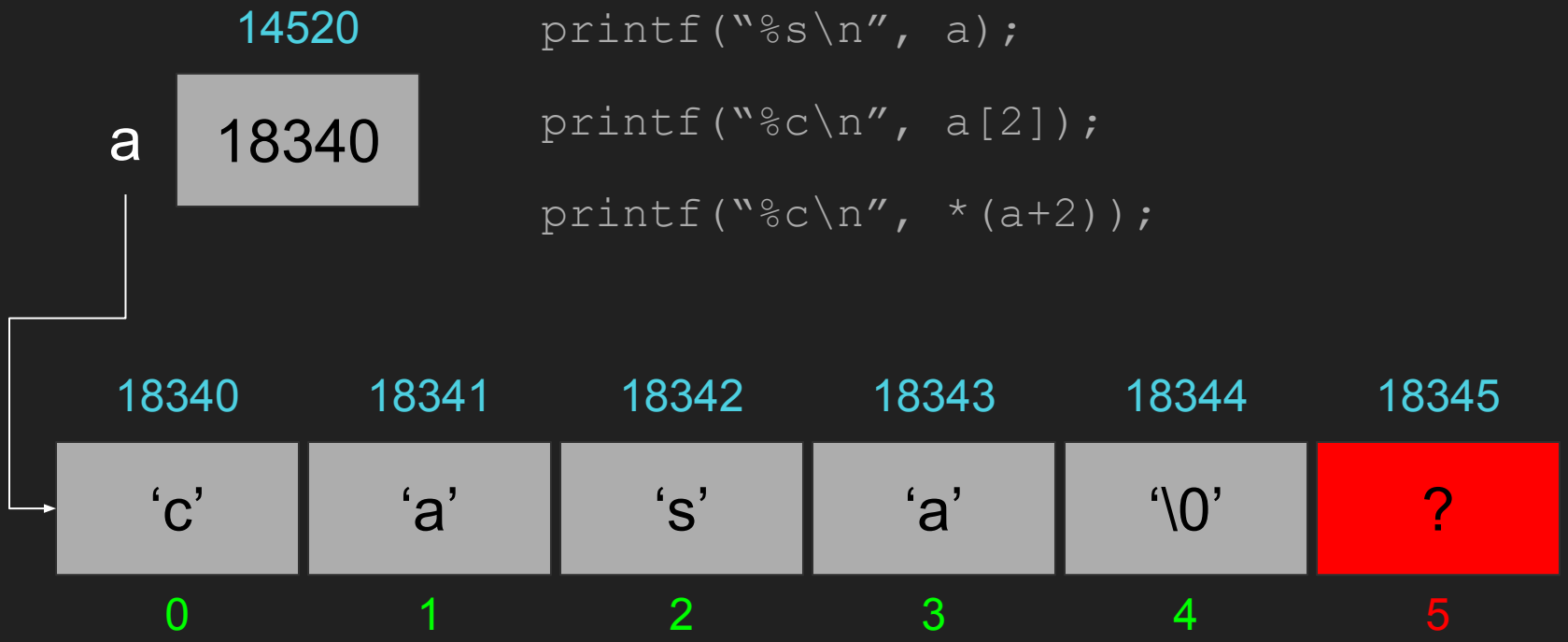
Strings

```
char a[5] = "casa";
printf("%s\n", a);
printf("%c\n", a[2]);
printf("%c\n", *(a+2));
```



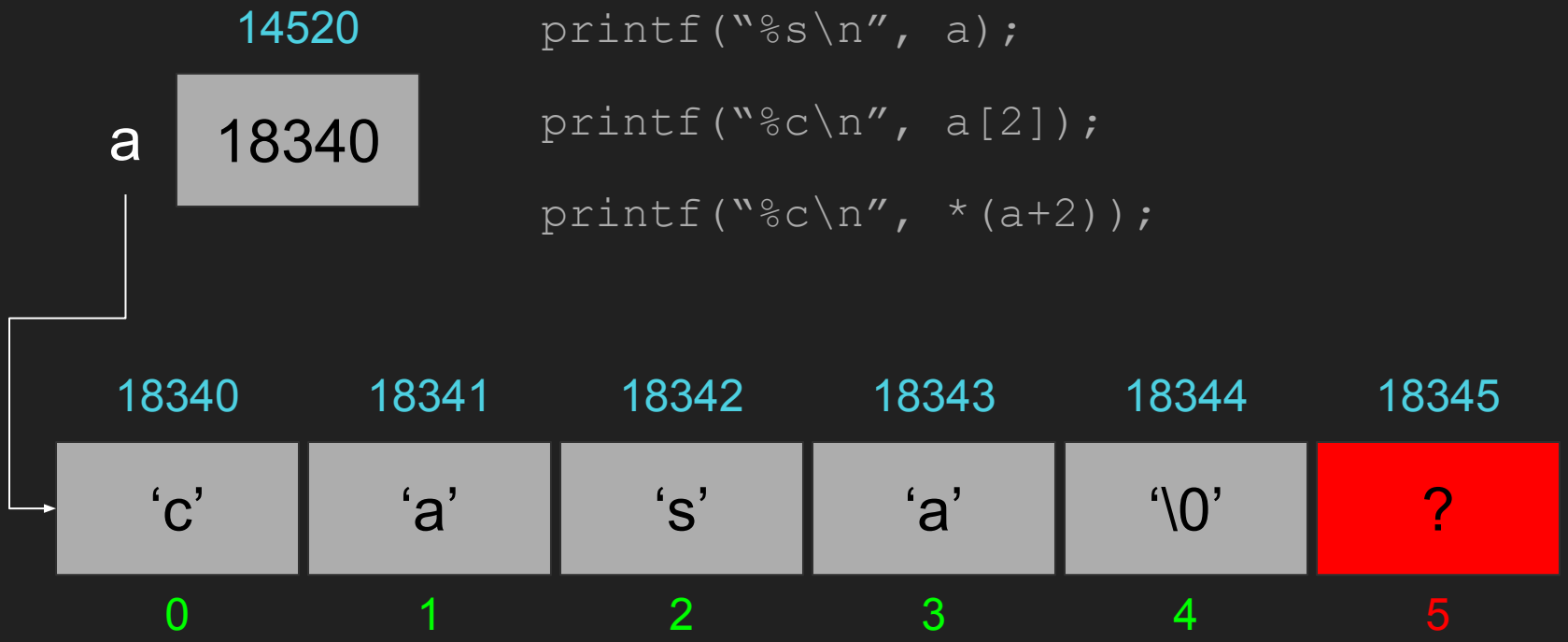
Strings

```
char a[5] = {'c', 'a', 's', 'a', '\0' };
printf("%s\n", a);
printf("%c\n", a[2]);
printf("%c\n", *(a+2));
```



Strings

```
char * a = "casa";
printf("%s\n", a);
printf("%c\n", a[2]);
printf("%c\n", *(a+2));
```



Strings

Alguma funções úteis declaradas em `<string.h>`:

- `strlen`: devolve o tamanho da string (caractere nulo não é contabilizado).
- `strcmp`: compara duas strings.
- `strncmp`: compara apenas os primeiros n bytes das duas strings.
- `strcpy`: copia o conteúdo de uma string em outra (que deve estar alocada).

Strings

Alguma funções úteis declaradas em `<string.h>`:

- `strlen`: devolve o tamanho da string (caractere nulo não é contabilizado).
- `strcmp`: compara duas strings.
- `strncmp`: compara apenas os primeiros n bytes das duas strings.
- `strcpy`: copia o conteúdo de uma string em outra (que deve estar alocada).

Um bom exercício é implementarmos nossas próprias versões destas funções!