Cadeias de caracteres

Programação I
2021.2022

Teresa Gonçalves
tcg@uevora.pt

Departamento de Informática, ECT-UÉ

Sumário

Tipos de dados

Strings na ling. C

Leitura e escrita

Funções sobre strings

Revisão

Tipos básicos em C

Tipos

- int
  - inteiro
- float
  - real
- char
  - carácter

Representação

- int
  - Complemento para 2
- float
  - Vírgula flutuante
- char
  - Um carácter é representado por um código específico
  - É necessário saber qual a **codificação** (encoding) utilizada
    - ASCII, iso-latin1, UTF8...

Código ASCII

ASCII

American Standard Code for Information Interchange

Original

7 bits

Extended ASCII

8 bits

Unicode

UTF-8, UTF-16, UTF-32

USASCII code chart

b_7 b_6 b_5 → / Bits ↓	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4 b_3 b_2 b_1 Row ↓ / Column →	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

Strings

String (cadeia de caracteres)

Vetor de char terminado pelo carácter ‘\0’

‘\0’ → carácter de código 0

Declaração

```
char txt [8]; /* txt é um vetor de 8 caracteres */
```

Declaração e inicialização

```
char msg[7] = {'V', 'i', 'v', 'a', ' ', '!', '\0'};
              /* msg é um vetor com 7 caracteres, que
                 constituem a sequência "Viva !” */
```

Homework...


char msg [ ]={78, 111, 119, 32, 72, 105, 114, 105, 110, 103,
EACanada.com
VIACOM

Carácter ‘\0’

Atenção

O carácter ‘\0’ ocupa uma posição do vetor

A sua não utilização pode resultar em **erros de execução imprevisíveis**!

Strings

Declaração e inicialização

```
char msg[7] = {'V', 'i', 'v', 'a', ' ','!','\0'};
char msg[7] = "Viva !";
```

Acesso aos elementos

```
msg[i]    /* acesso ao elemento no índice i [0..6] de msg */
printf("%c", msg[2]); /* o que imprime? */
```

Impressão do conteúdo

```
for(i = 0; i < 7; i++)
    printf("%c", msg[i]);  /* caracter a caracter */
printf("%s\n", msg);  /* toda a sequência: */
```

Leitura e escrita de strings

Leitura

Utilização das funções scanf, gets, fgets

```
char nome [30];
scanf("%s", nome);
  /* lê uma palavra (seq. de caracteres até o carácter ‘ ‘) */

gets(nome);
  /* lê uma frase (seq. de caracteres até o carácter ‘\n’) */

fgets(nome, max, stdin);
  /* lê uma frase com máximo de max-1 caracteres */
```

Escrita

Utilização das funções `printf` e `puts`

```
char nome [30];
printf("%s", nome);
  /* escreve os caracteres de nome até encontrar '\0' */

puts(nome);
  /* o mesmo que printf( "%s\n", nome); */
```

Bibliotecas C

Diretiva #include

A ling. C tem um conjunto de bibliotecas com funções implementadas

Para verificar a "assinatura" das funções é obrigatória a inclusão do ficheiro com os cabeçalhos (protótipos)

Ficheiros com extensão .h

A inclusão é feita através da diretiva `#include`

Bibliotecas

stdio.h

Funções para entrada/saída (I/O)

```
#include <stdio.h>
```

stdlib.h

Funções genéricas

```
#include <stdlib.h>
```

string.h

Funções para manipulação de strings

```
#include <string.h>
```

Funções sobre strings

Funções sobre strings (1)

Comprimento

```
int strlen(char *string);
```

char * equivalente a char []

Cópia

```
char *strcpy(char *dest, char *src);
```

Concatenação

```
char *strcat(char *string1, char *string2);
```

Exemplo (1)

```
#include <stdio.h>
#include <string.h>

main (){
  char nome [30] = "Francisco Alves", nome2 [40];
  printf("%d\n", strlen(nome));  /* o que imprime? */
  strcpy(nome2, nome);
  puts(nome2);    /* o que imprime? */
  strcat(nome2, nome);
  puts(nome2);    /* o que imprime? */
}
```

Funções sobre strings (2)

Comparação

```
int strcmp(char *string1, char *string2);
  /* devolve
  0 se iguais; -1 se string1<string2; 1 se string1>string2 */
```

Localização de um carácter

```
char *strchr(char *string, int c);
```

Localização de sub-string

```
char *strstr(char *string1, char *string2);
```

Exemplo (2)

```
#include <stdio.h>
#include <string.h>
main (){
  char nome [30] = "Francisco Alves", nome2 [40];
  strcpy(nome2, nome);
  printf("%d\n",
     strcmp (nome, nome2)); /* o que imprime? */
  puts(strchr (nome, 'a')); /* o que imprime? */
}
```

Conversões string - número

Protótipos em `stdlib.h`

Para inteiro

```
int atoi(char *string);
```

Para float

```
double atof(char *string);
```

Diretiva #define

Sinónimo para um valor

Exemplos

```
#define NCAR 20
#define NTOTAL 80
```

Exercício

Programa que

- lê sucessivas strings até ser especificado “.”
- Concatena todas numa só e mostra-a no final

Exercício

```
#include <stdio.h>
#include <string.h>
#define NCAR 20    /* numero max de caracteres de cada string lida */
#define NTOTAL 80  /* numero max de caracteres da string resultado */

void reads(char *s, int n) {
    fgets(s, n, stdin);
    s[strlen(s) - 1] = '\0';
}
int main (){
    char s[NCAR], stotal[NTOTAL] = "";
    printf("Escreva uma string com menos que %d caracteres: ", NCAR-1);
    reads(s, NCAR);
    while(!strcmp(s, ".") && strlen(stotal)+strlen(s)<NTOTAL-1) {
        strcat(s, " ");
        strcat(stotal, s);
        printf("Escreva uma string com menos que %d caracteres: ", NCAR-1);
        reads(s, NCAR);
    }
    printf("%s\n", stotal);
}
```

Outras funções

Números aleatórios

Protótipos em `stdlib.h`

Geração de um número aleatório

`int rand(void)`

Devolve um nº peseudo-aleatório entre 0 e RAND_MAX

Semente para gerador de números aleatórios

`void srand(unsigned int seed)`

Usando uma semente, é possível reconstruir a sequência de números aleatórios

Funções matemáticas

Protótipos em `math.h`

`double`: tipo

- nº real de maior precisão

Trigonométricas

```
double sin(double x)
double cos(double x)
double asin(double x)
double acos(double x)
```

Outras funções

```
double exp(double x)
double log(double x)
double log10(double x)
double pow(double x,
double y)
double sqrt(double x)

double ceil(double x)
double floor(double x)
```