ANSI

C

Introdução à Programação

Eduardo Silva Lira

XLVIII Programa de Verão do IME-USP

São Paulo - SP, Jan 2019

INSTITUTO DE MATEMATICA E ESTATÍSTICA
UNIVERSIDADE DE SÃO PAULO

USP

Agora tem Wi-fi!

Utilize a rede wireless do programa de verão!

Rede: `verao`

Senha: `programaverao2019`

Ambiente Online

https://saw.atp.usp.br/

Usuário: e-mail cadastrado no verão

Senha: Verao2019_intro_N

(sendo N = numero de inscricao do aluno)

Prepare-se!

Vídeo Tutorial para Download do CodeBlocks

Windows:
https://www.youtube.com/watch?v=CiwPDUOvIMU

Linux:
https://www.youtube.com/watch?v=-rxhENQKCHw

Ambiente online para programação C

https://www.tutorialspoint.com/compile_c_online.php

Bibliografia

DEITEL, H.M. e DEITEL, P.J., "Como Programar em C", 2a ed., Livros Técnicos e Científicos, 1999.

ASCENCIO, A. F. G. e CAMPOS, E. A. V Fundamentos da Programação de Computadores – Algoritmos, Pascal e C/C++. Prentice Hall. 2007.

FORBELLONE, A. L. V. e EBERSPACHER, H. F. Lógica de Programação – A Construção de Algoritmos e Estrutura de Dados. 3a Edição. Prentice Hall, 2005.

SOUZA, M. A. F. et al. Algoritmos e Lógica de Programação. C. Learning. 2008.

Data das Provas

Prova 1 - 24 de Janeiro
Prova 2 - 14 de Fevereiro

Revisão dos tópicos anteriores

- Paradigma Estrutural x Orientado a Objetos
- Linguagem Baixo x Alto nível
- Linguagem Compilada x Interpretada
- Linguagem C
- Softwares necessários para programar
- Nosso primeiro programa em C

Como trabalhar com dados em C?

- Dados necessitam ser armazenados em algum local:

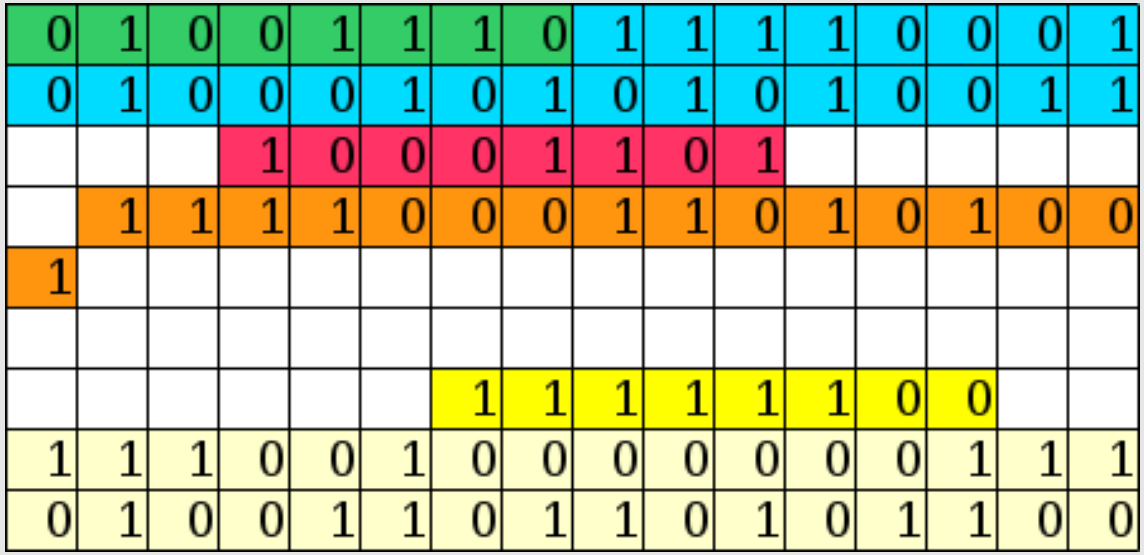
Como trabalhar com dados em C?

- Dados necessitam ser armazenados em algum local:
  - Memória
  - Variáveis

Memória

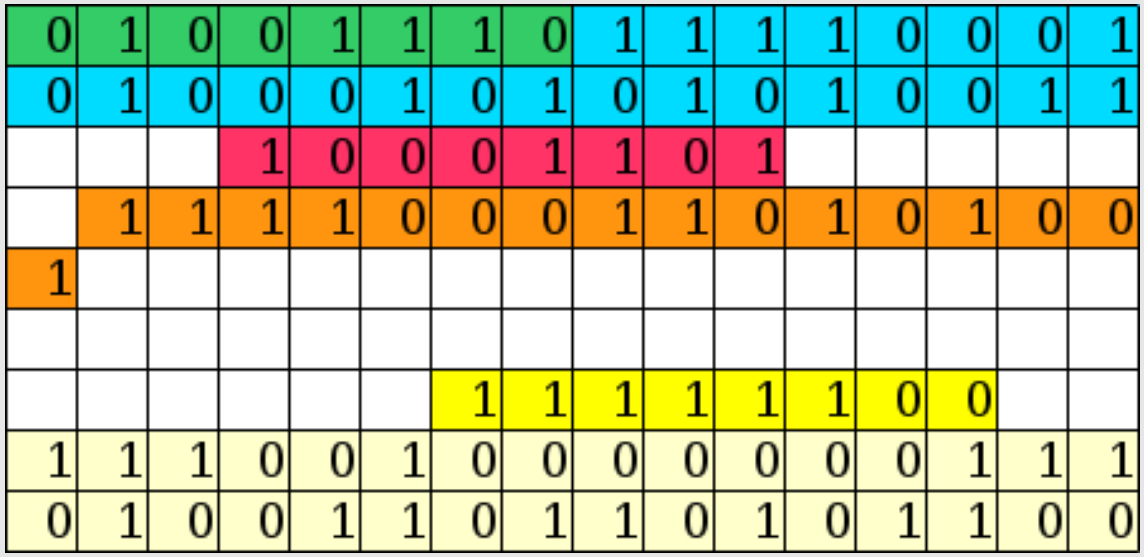
- Memória é um recurso finito?

Representação da memória como uma matriz:



Memória

- Unidades básicas
  - Bit: **0** ou **1**
  - Byte: conjunto de **8 bits**



Variáveis - Identificadores

- Nome que identifica um endereço de memória
- Lembre-se: C é *case-sensitive*
- Para nomes de variáveis, utilize os caracteres…
  - "**A**-**Z**"
  - "**a**-**z**"
  - "**0**-**9**"
  - e o *underline* "**_**"

Atenção: Identificadores não podem começar por números!

Identificadores

- Convenções (padrão de nomenclatura):
  - lower_case_with_underscores
  - mixedCase

- Ex:
  - precoUnitario, qtdeEstoque, idade, autor, dataNasc
  - preco_unitario, qtde_estoque, idade, autor, data_nasc

Identificadores - Palavras-chave

- Palavras-chave não podem ser utilizadas como identificadores

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Palavras-chave em ANSI C. Fonte: Livro “C Completo e Total”.

Variáveis em C

- Necessitam ser:
  - Declaradas
    - Onde?
  - Inicializadas
    - Atribuir um valor inicial

Como declarar variáveis em C?

- Variáveis locais
  - Declarar no escopo da função

- Definir um **tipo** e um **identificador**
  - Ex:
    - **int idade;**
    - **float preco;**

Tipos de dados em C

- int
- float
- double
- char
- void

Tipo int

- Representação com sinal: positivos e negativos
- 32 bits armazenáveis (**4 bytes**)
- Formatadores para entrada: **%d** ou **%i**
- Formatadores para saída:

  %d, **%i**, **%o**, **%x** ou **%X**

Tipo int

- Formatadores para entrada:
  - **%d**: entrada de números em base 10
  - **%i**:
    - Entrada em base 10. Ex: 21
    - Entrada em base 8. Ex: 057
    - Entrada em base 16. Ex: 0x4AF

- Formatadores para saída:
  %d, **%i**, **%o**, **%x** ou **%X**

Tipo int

- Formatadores para saída:
  - **%d**: exibe em base 10
  - **%i**: igual %d
  - **%o**: exibe em base 8
  - **%x**: exibe em base 16 com letras minúsculas
  - **%X**: exibe em base 16 com letras maiúsculas

Tipo float

- Utilizado com números fracionários
  - Precisão **simples**
- **32 bits** (**4 bytes**) armazenáveis
- Representação com sinal
- Formatadores de entrada: **%f**
- Formatadores de saída: **%f**, **%e**, ou **%g**

Tipo float

- Formatadores de saída: **%f**, **%e**, ou **%g**
  - **%f**: representação normal do número real
  - **%e**: representação em notação científica
  - **%g**: representação mais curta possível, em notação ou não

Tipo double

- Utilizado com números fracionários
  - Precisão **dupla**
- 64 bits (8 bytes) armazenáveis
- Representação com sinal
- Formatadores de entrada: **%lf**
- Formatadores de saída: **%f**, **%e**, ou **%g**

Tipo char

- Utilizado com códigos de caractere
  - Tabela ASCII, UTF8
- 8 bits (1 byte) armazenáveis
- Por padrão, representação com sinal

  (-128 a 127)
- Formatador **%c**

Tipo char - Tabela ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Fonte: http://www.asciitable.com/

Tipo char - Tabela ASCII Extendida

128	Ç	144	É	160	á	176	▒	192	└	208	╨	224	α	240	≡
129	ü	145	æ	161	í	177	▒	193	┴	209	╤	225	ß	241	±
130	é	146	Æ	162	ó	178	▓	194	┬	210	╥	226	Γ	242	≥
131	â	147	ô	163	ú	179	│	195	├	211	╙	227	π	243	≤
132	ä	148	ö	164	ñ	180	┤	196	─	212	╘	228	Σ	244	⌠
133	à	149	ò	165	Ñ	181	╡	197	┼	213	╒	229	σ	245	⌡
134	å	150	û	166	ª	182	╢	198	╞	214	╓	230	µ	246	÷
135	ç	151	ù	167	º	183	╖	199	╟	215	╫	231	τ	247	≈
136	ê	152	ÿ	168	¿	184	╕	200	╚	216	╪	232	Φ	248	°
137	ë	153	Ö	169	⌐	185	╣	201	╔	217	┘	233	Θ	249	∙
138	è	154	Ü	170	¬	186	║	202	╩	218	┌	234	Ω	250	·
139	ï	155	¢	171	½	187	╗	203	╦	219	▉	235	δ	251	√
140	î	156	£	172	¼	188	╝	204	╠	220	▄	236	∞	252	n
141	ì	157	¥	173	¡	189	╜	205	═	221	▌	237	φ	253	²
142	Ä	158	₧	174	«	190	╛	206	╬	222	▐	238	ε	254	■
143	Å	159	ƒ	175	»	191	┐	207	╧	223	▀	239	∩	255	

Source: www.LookupTables.com

Fonte: http://www.asciitable.com/

Tipo void

- Utilizado para funções sem retorno ou parâmetros
  - Representa o nada!

Modificadores

- **unsigned**
  - Sem sinal
- **signed**
  - Com sinal
- **long**
- **short**

Constantes

- Constantes:
  - Valores, guardados em um espaço de memória, que **não** devem ser **alterados durante a execução de um programa**.
  - Diferencie pelo padrão de nomenclatura:
  - UPPERCASE
  - UPPER_CASE_WITH_UNDERSCORES
  - Exemplo (não é código fonte):
    - MAX_LINHA = 1000
    - MIN_IDADE = 18

Constantes

- Constantes:
  - Valores, guardados em um espaço de memória, que **não** devem ser **alterados durante a execução de um programa**.
  - Utilize **#define** ou **const**

Constantes

- Exemplo:

```
#include<stdio.h>

int main(){

  #define PI 3.1415
  int const MAIORIDADE = 18;

  printf("Agora posso utilizar constantes!\n%f\n%d\n\n", PI, MAIORIDADE);

  return 0;

}
```

Dicas!

Exemplo: macros para renomear comandos!

```c
/*Definindo macros!*/
#include<stdio.h>

/*Definir macros para leitura e exibição*/
#define exiba printf
#define leia  scanf

/*Funcao principal*/
int main(){

  double preco;

  exiba("Digite o preco do produto: ");
  leia("%lf", &preco);

  exiba("Preco informado: R$ %.2f\n", preco);

  return 0; /*Execucao ocorreu normalmente*/
}
```

Dicas para desenvolvedores

Como desenvolvedor de soluções, sempre pense:

O programa que estou desenvolvendo será utilizado por outras pessoas. Como posso deixá-lo da melhor maneira em relação a **usabilidade**?

Outros desenvolvedores poderão efetuar **manutenções** em seu código. Como posso deixar o código fonte da maneira mais clara possível?

Dicas para desenvolvedores

Informações interessantes sobre formatadores da função printf:

http://planeta-globo.com/2012/03/04/aprendendo-c-formatacao-do-printf/

Como resolver este problema?

- Construa um programa que tenha a informação da idade e do salário de um usuário.
- Exiba na tela a seguinte mensagem: “Você tem **x** anos e seu salário é R$ **y**”.

Como trabalhar com dados em C?

exSalario.c

```c
/*
Construa um programa que tenha a informação da idade e do salário de um usuário.
Exiba na tela a seguinte mensagem: “Você tem x anos e seu salário é R$ y”.
*/

#include<stdio.h>

int main(){
   int idade;
   float salario;

   idade = 25;
   salario = 1890.00;

   printf("Voce tem %d anos e seu salário é R$ %f.\n", idade, salario);

   return 0;
}
```

Alguns exercícios

1. Crie um programa para imprimir o cubo de um número inteiro.

2. Faça um programa para somar dois inteiros. Guarde o resultado em outra variável e o exiba.

3. Crie um programa para converter polegadas para cm.

4. Crie um programa que calcule a área de um triângulo retângulo.

Dúvidas?