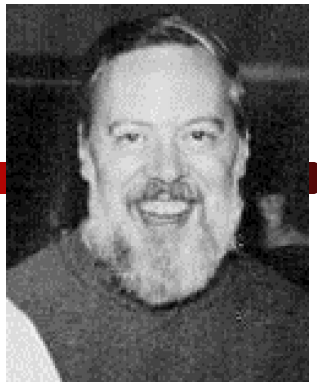
Programação em C

De Algoritmos para Programas

- **Programação é...**
  - *A sequência de planejamento, projeto, escrita e testes de instruções desempenhadas pelo computador.*
  - *Envolve obediência às regras (léxicas, sintáticas e semânticas), organização, otimização, documentação.*

A Linguagem C



- A linguagem C foi criada por Dennis Ritchie, em 1972, no centro de Pesquisas da Bell Labs.
- Sua primeira utilização importante foi a reescrita do Sistema Operacional UNIX que, até então, era escrito em Assembly.
- Em meados de 1973, o UNIX saiu do laboratório para ser liberado para as universidades.
- Foi o suficiente para que o sucesso da linguagem atingisse proporções tais que, por volta de 1980, já existiam várias versões de compiladores C oferecidas por várias empresas, não sendo mais restritas apenas ao ambiente UNIX, e compatíveis com outros sistemas operacionais.
- O C é uma linguagem de propósito geral, sendo adequada à programação estruturada.

Estrutura de um Programa em C

- **Bibliotecas**

  #include <*nome biblioteca*>

- Feito no início do arquivo do programa, serve para avisar ao compilador que serão usados procedimentos, funções, variáveis ou constantes declarados no arquivo especificado

- Bibliotecas de uso comum em C

```
#include <stdio.h> /* Funções de entrada e saída */
#include <math.h> /* Funções matemáticas */
#include <stdlib.h> /* Funções de gerência de memória */
#include <string.h> /* Funções de manipulação de strings */
```

Estrutura de um Programa em C

- **Função Principal**
- Todo programa em C inicia sua execução chamando a função principal main(), sendo obrigatória a sua declaração.

```
int main()
{
   instruções
}
```

- As instruções se encerram com ; (ponto e vírgula).
- Exemplo:

```
#include <stdio.h>
int main()
{
   printf("Este e um programa em C\n");
}
```

Programando em Linguagem C

- **Palavras reservadas:** palavras que tem significado especial previamente definido, não podendo ser usadas como identificador de variáveis.

  Exemplo: if, for, main.

- **Comentários:** texto explicativo entre /* e */ (comentário de bloco) ou após // (comentário de linha), para auxiliar no entendimento dos diversos blocos do programa.

- **A linguagem C é *case-sensitive*:** diferencia maiúsculas de minúsculas.

Programando em Linguagem C

- **Operadores Aritméticos**

  - Unários: - (valor negativo)

    - 5

  - Binários: +, -, *, /, %

10 + 3 = 13

10 - 3 = 7

10 * 3 = 30

10.0 / 3 = 3.3333...

10 % 3 = 1 (resto da divisão inteira)

Programando em Linguagem C

- **Operador de Atribuição (=)**
  - Para armazenar um valor em uma constante ou variável.

    nota = (7 + 8 + 6) / 3;
  - Primeiro, avalia-se o que está a direita.
  - Depois, o valor é armazenado na variável à esquerda.
  - À esquerda, só uma variável.
  - O valor anterior da variável é perdido.
  - O valor a ser atribuído deve ser compatível com o tipo da variável.
  - Qual o valor das variáveis ao final de cada operação?

    x = 4;

    y = x + 3;

    z = y * 2;

    x = z + x;

Programando em Linguagem C

- **Operadores de Incremento e Decremento**
  - Operadores para incrementar variáveis. O operador incremento soma 1 ao seu operando, e o decremento subtrai 1.
    - x++; equivale a x = x + 1;
    - x--; equivale a x = x - 1;
  - Podem ser usados como operadores pré-fixo(++x) ou pós-fixo(x++):
    - ++x incrementa x antes de utilizar o seu valor;
    - x++ incrementa x depois de ser utilizado;

```
x = 23;
y = x++;
no final tem-se y = 23 e x = 24
```

```
x = 23;
y = ++x;
```

Programando em Linguagem C

- **Atribuições especiais com operadores compostos**

  - num1 += 10; equivale a num1 = num1 + 10;
  - num2 *= 10; equivale a num2 = num2 * 10;
  - num2 -= 10; equivale a num2 = num2 - 10;
  - num2 /= 10; equivale a num2 = num2 / 10;

Tipos de Dados

- **Tipos de dados básicos:**
  - **char:** representa um caractere;
  - **int:** representa um número inteiro;
  - **float:** representa um número real com certa precisão;
  - **double:** representa um número real com precisão maior que float;
  - **void:** tipo vazio.

- **Modificadores de tipos de dados:**
  - **unsigned** = sem sinal;
  - **long** = domínio estendido;
  - **short** = domínio reduzido;

✔ **Ao float não se pode aplicar nenhum modificador e ao double pode-se aplicar apenas o long.**

Tipos de Dados

- **int** normalmente terá o tamanho natural para uma determinada máquina. Assim, numa máquina de 16 bits, **int** provavelmente terá 16 bits. Numa máquina de 32, **int** deverá ter 32 bits.

- Cada compilador é livre para escolher tamanhos adequados para o seu próprio hardware.

- As únicas restrições são de que **short int** e **int** devem ocupar pelo menos 16 bits; **long int**, pelo menos 32 bits; e **short int**, não pode ser maior que int, que não pode ser maior que long int.

Tipos de Dados

Tipo	Num de bits	Formato para leitura com scanf	Intervalo	
			Inicio	Fim
char	8	%c	-128	127
unsigned char	8	%c	0	255
signed char	8	%c	-128	127
int	16	%i	-32.768	32.767
unsigned int	16	%u	0	65.535
signed int	16	%i	-32.768	32.767
short int	16	%hi	-32.768	32.767
unsigned short int	16	%hu	0	65.535
signed short int	16	%hi	-32.768	32.767
long int	32	%li	-2.147.483.648	2.147.483.647
signed long int	32	%li	-2.147.483.648	2.147.483.647
unsigned long int	32	%lu	0	4.294.967.295
float	32	%f	3,4E-38	3.4E+38
double	64	%lf	1,7E-308	1,7E+308
long double	80	%Lf	3,4E-4932	3,4E+4932

Variáveis

- **Declarações de variáveis**

```
tipo_de_dado nome_variável;
```

- **Exemplo**

```
#include <stdio.h>
int main()
{
   /*declaração de variável para números  reais*/
   float nota;
   float prova1, prova2;
   ...
}
```

Variáveis

- **Inicialização de Variáveis**

```
int main()
{
   float nota;
   printf("%f", nota );
}
```

- **O que será mostrado?**

  - Declaração de variável = Reserva de espaço na memória.

  - Espaço pode estar limpo (zerado) ou não (lixo).

Variáveis

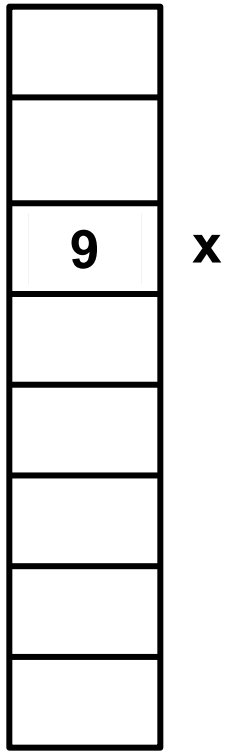
- **Exemplos de Inicialização de Variáveis**

```
int main()
{
      float nota1, nota2, media;

      nota1 = 10;
      nota2 = 5;
      media = nota1 + nota2;
      printf("%f", media );
}

int main()
{
      float nota1 = 10, nota2 = 5, media;

      media = nota1 + nota2;
      printf("%f", media );
}
```

Variáveis

- **Variáveis na memória**

```
int main()
{
 int x;

 x = 4 + 3;
 x = 9;
 …
}
```



Constantes

- Definem um valor fixo a ser utilizado.

- **Exemplo:** quando usamos um mesmo valor várias vezes.

```
/* Programa que calcula e exibe o tamanho da circunferência de um
círculo de raio 3,5,7 */
#include <stdio.h>
int main()
{
   printf ("Tamanho da circunferencia de um circulo de raio 3:");
   printf ("%f\n", 2*3.1415926536*3);
   printf ("Tamanho da circunferencia de um circulo de raio 5: ");
   printf ("%f\n", 2*3.1415926536*5);
   printf ("Tamanho da circunferencia de um circulo de raio 7: ");
   printf ("%f\n", 2*3.1415926536*7);
}
```

Constantes

- Podemos definir uma constante
  - Seu conteúdo não pode ser modificado durante a execução de um programa.

```
/* Programa que calcula e exibe o tamanho da circunferência de um
círculo de raio 3,5,7 */
#include <stdio.h>
#define PI 3.1415926536 /* PI tem o valor 3.1415926536 */
int main()
{
  printf ("Tamanho da circunferencia de um circulo de raio 3: ");
  printf ("%f\n", 2*PI*3);
  printf ("Tamanho da circunferencia de um circulo de raio 5: ");
  printf ("%f\n", 2*PI*5);
  printf ("Tamanho da circunferencia de um circulo de raio 7: ");
  printf ("%f\n", 2*PI*7);
}
```

Constantes

- **Outras constantes:**
  - Número inteiro:

```
#define V 1
/* V tem valor 1 */
```

  - Caractere:

```
#define UNIDADE 'm'
/* UNIDADE tem valor 'm' */
```

  - Texto (string ou cadeia de caracteres):

```
#define MSG "Informe um número inteiro
   positivo: "
/* um texto é o valor de MSG */
```

Saída de Dados

printf*("expressão de controle",argumentos*);

- **É uma função de I/O, que permite escrever no dispositivo padrão (tela).**
- **A expressão de controle pode conter caracteres que serão exibidos na tela e os especificadores de formatação que indicam o formato em que os argumentos devem ser impressos.**
- **Cada argumento deve ser separado por vírgula.**

Saída de Dados

Especificadores de formato

%c	caractere simples
%d	decimal
%e	notação científica
%f	ponto flutuante
%o	octal
%s	cadeia de caracteres
%u	decimal sem sinal
%x	hexadecimal

Caracteres de Escape:

\n	nova linha
\t	tab
\b	retrocesso
\"	aspas
\\	barra

- **Exemplo:**

```
int main()
{
   int x = 2;
   printf("%d e o conteudo da variavel x", x);
   printf("%s esta a %d milhoes de milhas \ndo Sol.", "Venus", 67);
}
```

Saída de Dados

Exemplos:

```
printf("%d",26);

printf("%f",123.4);

printf("%e",123.4); /*1.234e+2*/

printf("26 em hexa: %x",26); /*1A*/

printf("A primeira letra é %c",'A');

printf("Disciplina: %s","EXA841");
```

Saída de Dados

- **Tamanho de campos na expressão:**

```
main()
{
   printf ("\n%2d", 350); //350
   printf ("\n%4d", 350); // 350
   printf ("\n%6d", 350); //   350
}
```

- **Para arredondamento:**

```
main()
{
   printf ("\n%4.2f", 3456.78); //3456.78
   printf ("\n%3.2f", 3456.78); //3456.78
   printf ("\n%3.1f", 3456.78); //3456.8
   printf ("\n%10.3f", 3456.78);//  3456.780
   printf ("\n%.2f", 23.4567);  //23.46
}
```

Saída de Dados

- **Para alinhamento:**

```
main()
{
  printf ("\n%10.2f %10.2f %10.2f", 8.0, 15.3, 584.13);
  printf ("\n%10.2f %10.2f %10.2f", 834.0, 1500.55, 4890.21);
}
    8.00      15.30       584.13
  834.00    1500.55     4890.21
```

- **Complementando com zeros à esquerda:**

```
main()
{
   printf ("\n%04d", 21);
   printf ("\n%06d", 21);
}
0021
000021
```

Saída de Dados

- **Imprimindo caracteres:**

```
main()
{
    printf ("%d %c %x %o\n", 'A', 'A', 'A', 'A');
    printf ("%c %c %c %c\n", 'A', 65 , 0x41, 0101);

}
```

```
65A  41  101
 A   A   A   A
```

- A tabela ASCII possui 256 códigos de 0 a 255;
- Se imprimirmos em formato caractere um número maior que 255, será impresso o resto da divisão do número por 256;
- Se o número for 3393 será impresso A pois o resto de 3393 por 256 é 65.

Saída de Dados: Tabela ASCII

DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
0	0		32	20		64	40	@	96	60	`
1	1	☺	33	21	!	65	41	A	97	61	a
2	2	☻	34	22	"	66	42	B	98	62	b
3	3	♥	35	23	#	67	43	C	99	63	c
4	4	♦	36	24	$	68	44	D	100	64	d
5	5	♣	37	25	%	69	45	E	101	65	e
6	6	♠	38	26	&	70	46	F	102	66	f
7	7	•	39	27	'	71	47	G	103	67	g
8	8	◘	40	28	(	72	48	H	104	68	h
9	9	○	41	29	)	73	49	I	105	69	i
10	A	◙	42	2A	*	74	4A	J	106	6A	j
11	B	♂	43	2B	+	75	4B	K	107	6B	k
12	C	♀	44	2C	,	76	4C	L	108	6C	l
13	D	♪	45	2D	-	77	4D	M	109	6D	m
14	E	♫	46	2E	.	78	4E	N	110	6E	n
15	F	☼	47	2F	/	79	4F	O	111	6F	o
16	10	►	48	30	0	80	50	P	112	70	p
17	11	◄	49	31	1	81	51	Q	113	71	q
18	12	↕	50	32	2	82	52	R	114	72	r
19	13	‼	51	33	3	83	53	S	115	73	s
20	14	¶	52	34	4	84	54	T	116	74	t
21	15	§	53	35	5	85	55	U	117	75	u
22	16	▬	54	36	6	86	56	V	118	76	v
23	17	↨	55	37	7	87	57	W	119	77	w
24	18	↑	56	38	8	88	58	X	120	78	x
25	19	↓	57	39	9	89	59	Y	121	79	y
26	1A	→	58	3A	:	90	5A	Z	122	7A	z
27	1B	←	59	3B	;	91	5B	[	123	7B	{
28	1C	∟	60	3C	<	92	5C	\	124	7C	|
29	1D	↔	61	3D	=	93	5D	]	125	7D	}
30	1E	▲	62	3E	>	94	5E	^	126	7E	~
31	1F	▼	63	3F	?	95	5F	_	127	7F	⌂

DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	β
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	∅
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	∈
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	▒	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	|	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	ⁿ
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	

Entrada de Dados

scanf*("expressão de controle", argumentos*);

- **É uma função de I/O, que permite ler dados formatados da entrada padrão (teclado).**

- **Expressão de controle**
  - Cadeia de caracteres que informa como será a entrada;
  - Sempre contém o especificador de formato para a variável, indicado pelo caractere '%';
- **Argumentos**
  - Dependem da lista de especificadores de formato;
  - Sempre são variáveis, precedidas por '&' (importante!);

Entrada de Dados

- **Exemplos:**

```
...
int mat;
float media;
printf("Digite a matricula: ");
scanf("%d", &mat);
printf("Digite a media: ");
scanf("%f", &media);
...

...
int x,y;
printf ("Entre com o par ordenado:");
scanf ("%d%d", &x, &y);
printf ("x=%d y=%d",x,y);
...
```

Programando em C

- Regras práticas para programas legíveis:
  - Usar comentários e documentar código;
  - Dar nomes significativos às variáveis;
  - Colocar parênteses nas expressões para que a ordem das operações fique clara;
  - Usar recuo para separar visualmente os blocos do programa;
  - Alinhar informações relacionadas;
  - Evitar longas linhas de programação, quebrando-as quando necessário;

Programando em C

```
#include <stdio.h>
int main(){
int A,B,C;
A=1;B=2;C=A+B;
printf("%d",C);
}
```

ou

```
#include <stdio.h>

int main(){

    int num1,num2,soma;

    num1 = 1;
    num2 = 2;

    /* Realiza a soma */
    soma = num1 + num2;

    /* Exibe resultado */
    printf("A soma é %d",soma);
}
```

Integrated Development Environment (IDE)

- Ambientes de Desenvolvimento Integrado (IDEs) são softwares ou pacotes de softwares que facilitam a tarefa de programação.
- Geralmente contam com um **editor de texto** (com recursos de ressaltar a sintaxe por meio de cores, identificação de erros, identificação automática, auto completar, etc.), um **compilador e** um **depurador**.
- IDEs livres voltados para a linguagem C/C++:
  - Dev C++,CodeBlocks, etc.

Exercícios

- Escreva um programa que leia o peso e a altura de uma pessoa. Em seguida o programa deve calcular e imprimir índice de massa corpórea (IMC) dessa pessoa. Dado:

$$IMC = \frac{peso}{altura^2}$$

- Um grupo de amigos pretende alugar um carro por um único dia. Consultadas duas agências, a primeira cobra R$62,00 pela diária e R$1,40 por quilômetro rodado. A segunda cobra diária de R$80,00 e mais R$1,20 por quilômetro rodado. Escreva um programa que leia a quantidade de quilômetros a serem rodados e calcule e imprima na tela o preço a ser pago em cada uma das agências.

Exercícios

- Escreva um programa que calcule o valor do desconto de uma mercadoria paga a vista e o valor total a ser pago. O programa deve ler o valor da mercadoria e a porcentagem do desconto. Depois o programa deve calcular e imprimir na tela o valor do desconto e o novo valor da mercadoria com o desconto.
- Escreva um programa para ajudar a calcular a quantidade de gotas de um remédio que uma determinada criança precisa tomar. A bula desse remédio pediátrico recomenda a seguinte dosagem: 5 gotas para cada 2 kg do peso da criança. Você deve fazer um programa que leia o peso desta criança, calcule e imprima na tela a quantidade de gotas a ser tomada.
- Faça um programa que leia a idade de uma pessoa expressa em dias e mostre-a expressa apenas em anos, meses e dias. Assuma, neste programa, que um ano tem 365 dias e que um mês tem 30 dias**.** **Exemplo:** Se a pessoa digitar que viveu 10260 dias significa que ela tem 28 anos, 1 mês e 10 dias.