PROGRAMAÇÃO A

Estrutura Sequencial

INTRODUÇÃO

A estrutura mais simples que um algoritmo pode ter é conhecida como **sequencial.** Nessa estrutura os passos do algoritmo são executados, um a um, na ordem em que são especificados. Embora essa estrutura seja muito simples, ela é fundamental para a construção de todo e qualquer programa.

Normalmente, a estrutura sequencial mais básica é composta por:

- **Entrada de dados,** que lê dados digitados pelo usuário e os armazena em posições de memória representadas por variáveis.
- **Processamento,** que calcula expressões compostas por constantes e variáveis e armazena seus resultados em outras variáveis.
- **Saída de dados,** que exibe no vídeo por meio de instruções de saída os resultados dos cálculos efetuados.

INTRODUÇÃO

- **Bibliotecas** são arquivos contendo várias funções que podem incorporadas aos programas escritos em C. A diretiva **#include** faz o texto contido na biblioteca especificada ser inserido no programa.
- A biblioteca **stdio.h** permite a utilização de diversos comandos de entrada e saída de dados.
- É importante salientar que a linguagem C é **sensível** (case-sensitive) a letras maiúsculas e minúsculas, ou seja, considera que letras maiúsculas (por exemplo, **a é diferente de A**). Sendo assim, todos os comandos devem, obrigatoriamente, ser escritos com letras minúsculas.

```
#include <nome_da_biblioteca>
int main()
{
    bloco_de_comandos;
    return 0;
}
```

Tabela 1 - Estrutura básica de um programa escrito em C.

DECLARAÇÃO DE VARIÁVEIS EM C

As **variáveis** são declaradas após a especificação de seus tipos. Os **tipos de dados** mais utilizados são **int** (para números inteiros), **float** (para números reais) e **char** (para um caractere). A linguagem C não possui tipo de dados **boolean** (que pode assumir os valores verdadeiro ou falso), pois considera verdadeiro qualquer valor diferente de 0 (zero). A linguagem C não possui um tipo especial para armazenar cadeias de caracteres (**strings**). Deve-se, quando necessário, utilizar um vetor contendo vários elementos do tipo **char**.

Exemplos:

```
int x;
```

Declara uma variável chamada x em que pode ser armazenado um número inteiro.

```
float y, z;
```

Declara duas variáveis chamadas y e z em que podem ser armazenados dois números reais.

```
char sexo;
```

Declara uma variável chamada sexo em que pode ser armazenado um caractere.

DECLARAÇÃO DE VARIÁVEIS EM C

É possível também inicializar uma variável com um valor pré-definido quando a mesma é declarada.

Exemplos:

```
int x = 1;
```

Declara uma variável chamada x do tipo inteiro atribuindo-lhe o valor 1.

```
float y = 1.5, z = 100.25;
```

Declara duas variáveis chamadas y e z do tipo reais, que armazenam um valor inicial.

```
char sexo = 'M';
```

Declara uma variável chamada sexo do tipo caractere que possui o caractere 'M' como valor inicial.

Importante!

Antes de utilizar qualquer variável em um programa, a mesma deve estar previamente declarada. A declaração de variáveis em C geralmente é feita no inicio do programa dentro da função *main*. Na mesma instrução (linha) não podem ser usados tipos de dados diferentes para declarar variáveis.

DECLARAÇÃO DE VARIÁVEIS EM C

A linguagem C possui quatro tipos básicos (primitivos) que podem utilizados na declaração das variáveis: **int, float, double** e **char**. A partir desses tipos básicos, podem ser definidos outros:

Tipo	Faixa de valores	Tamanho (aproximado)
char	-128 a 127	8 bits
unsigned char	0 a 255	8 bits
int	-32.768 a 32.767	16 bits
unsigned int	0 a 65.535	16 bits
short int	-32.768 a 32.767	16 bits
long	-2.147.483.648 a 2.147.483.647	32 bits
unsigned long	0 a 4.294.967.295	32 bits
float	3.4×10^{-38} a 3.4×10^{38}	32 bits
double	1.7×10^{-308} a 1.7×10^{308}	64 bits
long double	3.4×10^{-4932} a 1.1×10^{4932}	80 bits

Tabela 2 - Tipos de dados da linguagem C.

TIPOS DE DADOS EM C

char são números.

Use **longs** para números inteiros bem grandes.

Use **shorts** para números pequenos e inteiros.

Use **ints** para a maioria dos números inteiros.

ints tem tamanhos diferentes em máquinas diferentes.

Use **floats** para a maioria dos números de ponto flutuante.

Use **doubles** para números de ponto flutuante bem precisos.

Para mais informações sobre os tipos de dados da linguagem C, consulte o documento Seu guia rápido para tipos de dados em C.pdf disponível no ambiente Moodle.

DECLARAÇÃO DE CONSTANTES EM C

As **constantes** são declaradas depois das bibliotecas (após a diretiva #include) e seus valores não podem ser alterados durante a execução do programa. A declaração de uma constante deve obedecer à seguinte sintaxe:

```
#define nome valor
```

Exemplos:

```
#define x 7
```

Define uma constante com identificador x e valor 7.

```
#define y 4.5
```

Define uma constante com identificador y e valor 4.5.

```
#define nome "DANILO"
```

Define uma constante com identificador nome e valor DANILO.

COMANDO DE ATRIBUIÇÃO EM C

O **comando de atribuição** é utilizado para conceder valores ou operações a variáveis, sendo representado pelo sinal de **= (sinal de igualdade)**.

Exemplos:

```
x = 4;
x = x + 2;
y = 2.5;
sexo = 'F';
```

Observações

- Em C, os caracteres são representados entre apóstrofos (‘). As cadeias de caracteres devem ser representadas entre aspas (“”).
- Em C, cada comando é finalizado com o sinal de ponto e vírgula (;).
- Em C, a parte inteira e a parte fracionária do número são separadas por um ponto.

COMANDO DE ENTRADA EM C

O **comando de entrada** é utilizado para receber dados digitados pelo usuário. Os dados recebidos são armazenados em **variáveis.** Um dos comandos de entrada mais utilizados na linguagem C é o **scanf.**

Exemplos:

scanf("%d%*c", &numero);	Um valor inteiro, digitado pelo usuário, será armazenado na variável numero.
scanf("%f%*c", &preco);	Um valor real, digitado pelo usuário, será armazenado na variável preco.
scanf("%c%*c", &sexo);	Um caractere, digitado pelo usuário, será armazenado na variável sexo.

No comando **scanf** é necessário indicar o tipo de variável que será lida: **%f** para variáveis que armazenam **números reais**; **%d** para variáveis que armazenam **números inteiros**; e **%c** para variáveis que armazenam um **único caractere.**

Para que o buffer seja esvaziado depois da atribuição do conteúdo a variável, utiliza-se **%*c.**

COMANDO DE SAÍDA EM C

O **comando de saída** é utilizado para mostrar dados na tela ou na impressora. Um dos comandos de saída mais utilizados na linguagem C é o **printf**.

Exemplos:

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

Mostra o número inteiro armazenado na variável valor.

```
printf("%f", x);
```

Mostra o número real armazenado na variável x.

```
printf("Conteúdo de y = %d", y);
```

Mostra a mensagem "Conteúdo de y = " e, em seguida, o número inteiro armazenado na variável y.

```
printf("%5.2f", x);
```

Mostra o número real armazenado na variável x utilizando cinco caracteres da tela e, destes, dois serão utilizados para a parte fracionária e um para o ponto, que é o separador da parte inteira e da parte fracionária.

```
printf("Conteúdo de x = %7.3f", x);
```

Mostra a mensagem "Conteúdo de x = " e, em seguida, o número real armazenado na variável x utilizando sete caracteres da tela e, destes, três serão utilizados para a parte fracionária e um para o ponto, que é o separador da parte inteira e da parte fracionária.

COMANDO DE SAÍDA EM C

No comando **printf** é necessário indicar o tipo de variável que será mostrada: **%f** para variáveis que armazenam **números reais**, **%d** para variáveis que armazenam **números inteiros** e **%c** para variáveis que armazenam um **único caractere**.

No comando **printf** pode-se utilizar caracteres para posicionar a saída, por exemplo, **\n**, que passa o cursor para a próxima linha, ou **\t**, que avança o cursor uma tabulação.

Exemplos:

```
printf("aula ");
printf("fácil");
```

```
printf("aula ");
printf("\nfácil");
```

```
printf("aula ");
printf("\tfácil");
```

CARACTERES DE CONTROLE

Dentro da cadeia de caracteres a ser exibida pela função **printf** podemos utilizar também **caracteres de controle.** A saída de um caractere de controle nem sempre causa a exibição de um símbolo no vídeo.

Controle	Efeito
\a	(*alarm*) soa o alarme do terminal
\b	(*back space*) muda o cursor para coluna anterior
\n	(*new line*) muda o cursor para a próxima linha
\r	(*return*) muda o cursor para o início da linha
\t	(*tab*) muda o cursor para a próxima marca de tabulação
\’	exibe um apóstrofo
\”	exibe uma aspa
\\	exibe uma barra invertida

Tabela 3 - Caracteres de controle e seus efeitos.

Exemplo:

printf(“\nTESTE\a”);

Quando este comando é executado, o cursor é movido para uma nova linha, a palavra TESTE é exibida e, em seguida, o alarme é soado (produzindo o som de um beep).

COMENTÁRIOS EM C

Comentários são textos que podem ser inseridos em programas com objetivo de documentá-los. Eles não são analisados pelo compilador.

Os comentários podem ocupar uma ou várias linhas, devendo ser inseridos nos programas utilizando-se os símbolos /* ..... */ ou //.

Exemplos:

```
/*
   linha de comentário
   linha de comentário
*/
```

A região de comentários é aberta com os símbolos /* e encerrada com os símbolos */.

```
// comentário
```

A região de comentários é aberta com os símbolos // e encerrada automaticamente ao final da linha.

OPERADORES E FUNÇÕES PREDEFINIDAS EM C

A linguagem C possui **operadores** e **funções** predefinidas destinados a cálculos matemáticos.

Operador	Exemplo	Comentário
=	x = y	O conteúdo da variável y é atribuído à variável x (A uma variável pode ser atribuído o conteúdo de outra, um valor constante ou, ainda, o resultado de uma função.
+	x + y	Soma o conteúdo de x e de y.
-	x - y	Subtrai o conteúdo de y do conteúdo de x.
*	x * y	Multiplica o conteúdo de x pelo conteúdo de y.
/	x / z	Obtém o quociente da divisão de x por y. Se os operadores são inteiros, o resultado da operação será o quociente inteiro da divisão. Se os operadores são reais, o resultado da operação será a divisão. Por exemplo: int z = 5/2; => a variável z receberá o valor 2. float z = 5.0/2.0; => a variável z receberá o valor 2.5.
%	x % y	Obtém o resto da divisão de x por y.

Tabela 4 - Operadores aritméticos da linguagem C.

Observação: O operador % só pode ser utilizado com operandos do tipo inteiro.

OPERADORES E FUNÇÕES PREDEFINIDAS EM C

Os operadores matemáticos de atribuição são utilizados para representar de maneira sintética uma operação aritmética e, posteriormente, uma operação de atribuição.

Operador	Exemplo	Comentário
+=	x += y;	Equivale a x = x + y.
-=	x -= y;	Equivale a x = x - y.
*=	x *= y;	Equivale a x = x * y.
/=	x /= y;	Equivale a x = x / y.
%=	x %= y;	Equivale a x = x % y.
++	x ++;	Equivale a x = x + 1.
++	y = ++x;	Equivale a x = x + 1 e depois y = x.
++	y = x++;	Equivale a y = x e depois x = x + 1.
--	x--;	Equivale a x = x - 1.
--	y = --x;	Equivale a x = x - 1 e depois y = x.
--	y = x--;	Equivale a y = x e depois x = x - 1.

Tabela 5 - Operadores matemáticos de atribuição da linguagem C.

OPERADORES E FUNÇÕES PREDEFINIDAS EM C

Operadores relacionais ou **comparativos** como o próprio nome diz, oferecem a possibilidade de estabelecer uma relação comparativa/condicional entre seus operandos.

Operador	Exemplo	Comentário
==	x == y	O conteúdo de x é igual ao conteúdo de y.
!=	x != y	O conteúdo de x é diferente do conteúdo de y.
<=	x <= y	O conteúdo de x é menor ou igual ao conteúdo de y.
>=	x >= y	O conteúdo de x é maior ou igual ao conteúdo de y.
<	x < y	O conteúdo de x é menor que o conteúdo de y.
>	x > y	O conteúdo de x é maior que o conteúdo de y.

Tabela 6 - Operadores relacionais da linguagem C.

OPERADORES E FUNÇÕES PREDEFINIDAS EM C

A linguagem C possui muitas **funções matemáticas** que podem ser observadas detalhadamente na documentação da biblioteca **math.h**.

Operador	Exemplo	Comentário
ceil	ceil(x)	Arredonda um número real para cima. Exemplo: ceil(3.2) é 4.
cos	cos(x)	Calcula o cosseno de x (x deve estar representado em radianos).
abs	abs(x)	Obtém o valor absoluto de x.
floor	floor(x)	Arredonda um número real para baixo. Exemplo: floor(3.2) é 3.
log	log(x)	Obtém o logaritmo natural de x.
log10	log10(x)	Obtém o logaritmo de base 10 de x.
pow	pow(x, y)	Calcula a potência de x elevado a y.
sin	sin(x)	Calcula o seno de x (x deve estar representado em radianos).
sqrt	sqrt(x)	Calcula a raiz quadrada de x.
tan	tan(x)	Calcula a tangente de x (x deve estar representado em radianos).

Tabela 7 - Exemplos de Funções Matemáticas da linguagem C.

PALAVRAS RESERVADAS DA LINGUAGEM C

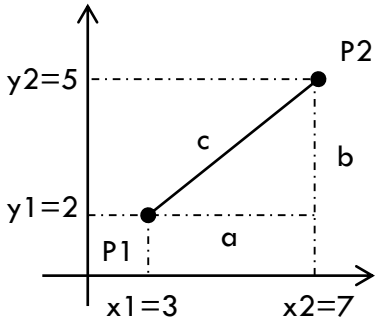
Palavras reservadas são nomes utilizados pelo compilador para representar comandos de controle do programa, operadores e diretivas. As palavras reservadas da linguagem C são:

asm	auto	break	case	cdecl	char
class	const	continue	_cs	default	delete
do	double	_ds	else	enum	_es
export	extern	far	_fastcall	float	friend
goto	huge	for	if	inline	int
interrupt	_loadds	long	near	new	operator
pascal	private	protected	public	register	return
_saveregs	_seg	short	signed	sizeof	_ss
static	struct	switch	template	this	typedef
union	unsigned	virtual	void	volatile	while

Tabela 8 - Palavras reservadas da linguagem C.

PROBLEMA ENVOLVENDO A ESTRUTURA SEQUENCIAL

Dadas as coordenadas de dois pontos no plano cartesiano, informe a distância entre eles.



Cálculo da distância entre dois pontos P1 e P2.

Teorema de Pitágoras

$$c = \sqrt{a^2 + b^2}$$

A partir de dois pontos indicados no plano cartesiano (P1 e P2), podemos desenhar um triângulo retângulo cujas medidas dos catetos a e b são dadas pelas diferenças entre as ordenadas e abscissas desses pontos e cuja medida da hipotenusa c, dada pelo Teorema de Pitágoras, é justamente a distância entre os pontos considerados.

A BIBLIOTECA DE FUNÇÕES MATEMÁTICAS

Para escrever fórmulas matemáticas (radiciação, potenciação, entre outras) em um programa escrito em linguagem C, é necessário:

- A inclusão da biblioteca matemática **math.h** é necessária sempre que usamos funções matemáticas (note que o uso de operadores aritméticos não requer a inclusão dessa biblioteca).
- A função **sqrt()** é usada para calcular a **raiz quadrada** de um valor; assim, por exemplo, **sqrt(x)** representa a raiz quadrada de x.
- A função **pow()** é usada para calcular a **potência** de um valor elevado a outro; assim, por exemplo, **pow(x,n)** representa a potência de x elevado a n.

Além destas duas funções matemáticas - **sqrt()** e **pow()** - há diversas outras funções úteis que podem ser consultadas no sistema de ajuda do compilador C.

RESOLUÇÃO DO PROBLEMA PROPOSTO

```c
/* Programa para calcular da distância entre dois pontos P1 e P2. */

#include <stdio.h>
#include <math.h>

int main()
{
    float x1, y1, x2, y2, a, b, c;

    printf("P1? ");
    scanf("%f %f", &x1, &y1);
    printf("P2? ");
    scanf("%f %f", &x2, &y2);

    a = x2 - x1;
    b = y2 - y1;
    c = sqrt(pow(a, 2) + pow(b, 2));

    printf("Distancia = %.1f\n", c);

    return 0;
}
```

INDENTAÇÃO

A indentação de um código escrito em uma linguagem de programação como C, se refere a organização e alinhamento correto das instruções de um programa. Veja os exemplos abaixo:

```c
#include <stdio.h>

int main()
{
    int x, y, soma;

    x = 1;
    y = 2;

    soma = x + y;

    printf("x + y = %d", soma);

    return 0;
}
```

```c
#include <stdio.h>
int main()
{
int x, y, soma;

x = 1;
y = 2;

soma = x + y;

printf("x + y = %d", soma);

return 0;
}
```

EXERCÍCIOS

Para cada problema a seguir, codifique um programa correspondente em linguagem C utilizando os conceitos da estrutura sequencial.

1. Dada uma distância percorrida (em quilômetros), bem como o total de combustível gasto (em litros), informe o consumo médio do veículo em km/l.

2. Dadas as medidas de uma sala em metros (comprimento e largura), bem como o preço do metro quadrado de carpete, informe o custo total para forrar o piso da sala.

3. O índice da massa corpórea (IMC) de uma pessoa é igual ao seu peso (em quilogramas) dividido pelo quadrado de sua altura (em metros). Dados o peso e a altura de uma pessoa, informe o valor de seu IMC.

4. Dado o tamanho de um arquivo (em bits), bem como a velocidade da conexão (em bits por segundo), informe o tempo necessário para o download do arquivo (em segundos).

5. Dados um capital C, uma taxa de juros mensal fixa J e um período de aplicação em meses M, informe o montante F no final do período. A fórmula a ser usada é a seguinte: $(F = C * (1 + (J/100))^M)$.

RESUMO

Declarações simples são comandos.

Cada programa precisa de uma função main.

A linguagem C possui operadores aritméticos, relacionais e lógicos.

Os nomes dos seus arquivos fonte devem terminar em .c.

#include inclui código externo para coisas como entrada e saída.

Use a função printf para mostrar um texto formatado na tela.

Você precisa compilar seu programa em C antes de executá-lo.

GCC é o compilador C mais usado.

Use a função scanf para ler um valor via teclado.

Palavras reservadas são nomes utilizados pelo compilador.

A biblioteca math.h contém as funções matemáticas.

#define é usado para definir constantes (macros).

REFERÊNCIAS BIBLIOGRÁFICAS

- ASCENCIO, A. F. G.; CAMPOS, E. A. V. D. **Fundamentos da Programação de Computadores: Algoritmos, Pascal, C/C++ (Padrão ANSI) e Java**. 3. ed. São Paulo: Pearson Education do Brasil, 2012. 569 p.
- FORBELLONE, A. L. V.; EBERSPACHER, H. F. **Lógica de Programação: A construção de algoritmos e estruturas de dados.** 3. ed. São Paulo: Prentice Hall, 2005. 218p.
- GRIFFITHS, D. **Use a Cabeça - C**. 1. ed. São Paulo: Alta Books: 2013. 632p.
- PEREIRA, S. D. L. **Algoritmos e Lógica de Programação em C: Uma abordagem didática**. 1. ed. São Paulo: Érica, 2010. 190 p.