ALGORITMOS DE PROGRAMAÇÃO

Marcela Santos

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS

Entrada e saída de dados

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Descrever o conceito de função.
- Aplicar as funções printf() e scanf().
- Usar códigos de formatação.

Introdução

Os programas de computador são sistemas que processam dados. A tarefa de um programa de computador é receber dados por meio de instruções de entrada, processar essas informações de acordo com a lógica criada e traduzida para uma determinada linguagem de programação e entregar esses dados processados, utilizando as instruções de saída de dados.

Neste capítulo, você estudará uma parte importante no desenvolvimento de sistemas: as instruções necessárias para realizar a entrada e saída de dados em um programa de computador.

O conceito de função

Para explicar o conceito de função, vamos começar com o seguinte código:

```
1  #include <stdio.h>
2  int main()
3  {
4      printf("Bem-vindo ao seu primeiro Jogo em C!\n");
5      return 0;
6  }
```

Figura 1. Código-fonte: o conceito de função.

Na linha 4, temos o uso de uma função existente em C, a `printf`. Esta função escreve na tela o texto existente em parênteses e entre aspas. Existe uma complexidade gigante para que seja escrito na tela do computador um valor e existam instruções específicas para isso.

Fique atento

Uma **função** é um conjunto de instruções que realizam uma determinada tarefa. Para que fique mais fácil o uso, elas têm nome.

Seria preciso um conjunto dessas instruções todas as vezes em que essa operação fosse necessária em um programa. Para facilitar esse trabalho, em programação, existem dois conceitos muito importantes: biblioteca e funções.

Na vida real, também temos algo parecido. Imagine que você tenha que fazer café. Automaticamente nós já realizamos todas as tarefas que estão envolvidas: colocar água para ferver, colocar o pó do café, esperar a água ferver e despejar a água quente sobre o pó de café (essa é a maneira que eu faço café!)

Então todas as vezes que eu vou fazer café, mando essas informações automaticamente para o meu cérebro. Em programação, é a mesma coisa: uma função é um conjunto de instruções organizadas em bloco, que possuem um nome e, quando for preciso, é chamada por meio deste.

As funções podem ser criadas pelos programadores ou podem estar incluídas nas bibliotecas de uma determinada biblioteca — e é aqui que entra o nosso segundo conceito: uma biblioteca de programação é um conjunto de funções que já foram pré-definidas por outros programadores; você precisa simplesmente adicionar a biblioteca ao seu programa e utilizar as funções.

Esse é o caso da função `printf`, presente na linha 4 do código apresentado na Figura 1. Agora, vamos aprender mais sobre essa e outra função, que são as funções básicas de entrada e saída na linguagem C.

As funções printf() e scanf()

Um computador é uma máquina que faz uso de programas e dados para realizar tarefas. Esses programas recebem esses dados, processam e entregam-nos como resposta a algum tipo de problema. Os programas são compostos de instruções que fazem esse processamento. Essas instruções podem ser funções de entrada e de saída, e é sobre elas que iremos tratar.

Vale a pena ressaltar que estamos tratando da linguagem de programação C, e essas funções sãos funções desta linguagem.

Fique atento

Para usar as funções scanf e printf, é preciso incluir a biblioteca stdio.h no código-fonte, por meio da linha

#include <stdio.h>

A função `printf` é a função de saída de dados no vídeo. Todas as funções têm uma sintaxe, que é uma regra que mostra como ela deve ser usada e qual o resultado esperado. A sintaxe de `printf` está representada na Figura 2, a seguir:

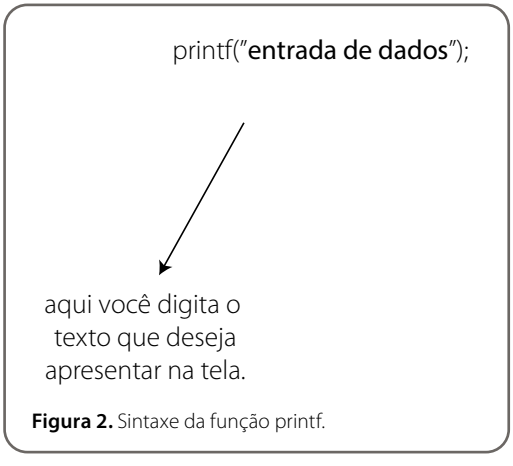

Figura 2. Sintaxe da função printf.

Essa é a forma mais básica para utilizarmos essa função, quando queremos simplesmente apresentar na tela uma mensagem de texto. Mas pode haver situações em que queremos apresentar o valor de uma variável. Quando isso for necessário, basta que, em vez do texto, coloquemos somente a variável sem as aspas.

Mas ainda pode existir a situação em que queremos apresentar um texto e uma variável na mesma saída de dados. Para entendermos como isso pode ser feito, vamos analisar o seguinte código, na Figura 3:

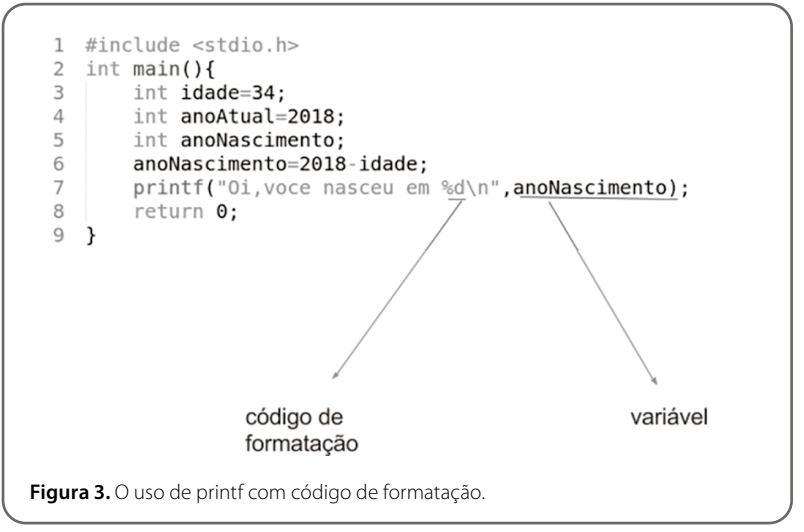

Figura 3. O uso de printf com código de formatação.

Esse código mostra o ano que a pessoa nasceu, usando, para isso, uma variável que contém a idade da pessoa. Foram definidas 3 variáveis do tipo inteiro: idade, anoAtual e anoNascimento.

As variáveis idade e anoAtual possuem valores iniciais, ou seja, não temos uma entrada de dados por parte do usuário que executa o programa criado com esse código-fonte. A variável anoNascimento é calculada usando-se dois operadores: o de atribuição e o de subtração — esses operadores serão vistos em capítulos posteriores.

Nesse primeiro momento, vamos entender o que a linha 6 faz, conforme representação na Figura 4:

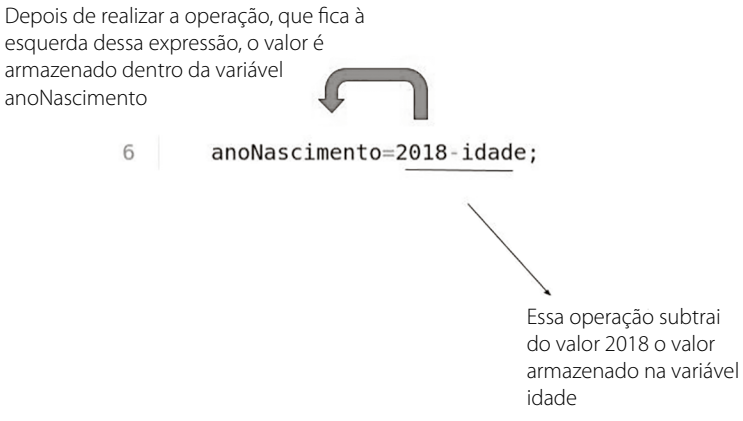

Figura 4. Expressão usando operadores e uma variável.

Com o valor armazenado na variável anoNascimento, falta realizarmos a saída de dados, ou seja, escrever esse valor na tela. Isso é feito na linha 7, utilizando-se a função `printf`, só que desta vez essa função é usada com um código de formatação e a concatenação de uma variável.

Concatenar uma variável é unir variáveis e/ou textos na mesma saída de dados, por exemplo. Para isso, adicionamos, ao final do texto que deverá aparecer na tela, uma vírgula (,), seguida das variáveis. Cada variável que é adicionada tem um tipo, e é preciso "avisar" ao `printf` que tipo é esse, para tal, o C tem o conceito de códigos de formatação. Mas antes de entendermos o que sãos os códigos de formatação, vamos à nossa função de entrada de dados.

Bom, agora vamos para a próxima função, a `scanf`. Da mesma forma que a função `printf` é uma função que está em uma biblioteca padrão do C, `scanf` também está; a diferença é que `scanf` é uma função de entrada de dados.

Vamos usar um exemplo de uso, apresentado na Figura 5, a seguir:

```
1  #include <stdio.h>
2  int main(){
3      int idade;
4      int anoAtual;
5      int anoNascimento;
6      printf("Digite sua idade: ");
7      scanf("%d",&idade);
8      printf("Digite o ano atual: ");
9      scanf("%d",&anoAtual);
10     anoNascimento=anoAtual-idade;
11     printf("Oi,voce nasceu em %d\n",anoNascimento);
12     return 0;
13 }
```

Figura 5. Código-fonte com o uso de printf e scanf.

Esse programa pede ao usuário que ele digite sua idade e o ano atual e, com esses dados, ele calcula o ano do nascimento. Na linha 6, temos o uso da função `printf`, que mostra na tela o texto, informando ao usuário a solicitação do dado.

Em seguida, aparece o uso da função `scanf`. Quando o usuário digitar um valor e teclar enter, a linha 7 pode ser traduzida para: "armazene na variável idade o valor inteiro que foi digitado". A sintaxe básica da função `scanf` é a que está representada na Figura 6:

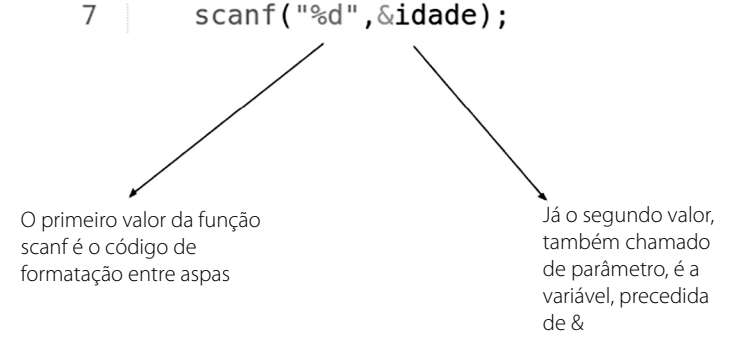

Figura 6. Sintaxe da função scanf.

Podemos, também, receber mais de um número no mesmo `scanf`, conforme pode ser visto no exemplo da Figura 7, a seguir:

```
1  #include <stdio.h>
2  int main()
3  {
4      int num1, num2;
5      printf("Insira dois numeros: ");
6      scanf("%d %d", &num1, &num2);
7      printf("Você digitou: '%d' e '%d'", num1, num2);
8  }
```

Figura 7. Código-fonte de um programa que soma dois valores, recebidos no mesmo scanf.

Códigos de formatação

Os códigos de formatação em C começam com % e permitem que as funções de entrada (`scanf`) e saída (`printf`) expressem os tipos de dados armazenados nas variáveis. Indicam o tipo do dado.

- %c: imprime o conteúdo da variável com representação ASCII.
- %d: imprime o conteúdo da variável com representação decimal com sinal.
- %u: imprime o conteúdo da variável com representação decimal sem sinal.
- %o: imprime o conteúdo da variável com representação octal sem sinal.
- %x: imprime o conteúdo da variável com representação hexadecimal sem sinal.
- %f: imprime o conteúdo da variável com representação com ponto decimal.
- %e: imprime o conteúdo da variável com representação em notação científica (exponencial).
- %g: formato geral, escolhe a representação mais curta entre %f e %e.

Além desses códigos de formatação em C, nós temos as sequências de escape que podem auxiliar para impressão de texto na tela de forma formatada. Dentre elas, os mais usados são.

- \n: quebra de linha
- \t: realiza uma tabulação horizontal
- \b: adiciona um “backspace” ao texto
- \r: volta o cursor para o começo da linha sem mudar de linha
- \a: emite um sinal sonoro
- \”: quando você precisa, no seu texto, do uso de aspas duplas
- \’: quando você , no seu texto, do uso de aspas simples
- \\: quando você precisa, no seu texto, do uso de barra invertida
- \0: caractere nulo

No código da Figura 8, a seguir, você pode ver o uso de códigos de formatação e sequências de escape.

```
1  #include <stdio.h>
2  int main ()
3  {
4     int a;
5     printf ("Digite um número: ");
6     scanf ("%d", &a);
7     printf ("\nO número digitado foi %d\n", a);
8     return (0);
9  }
```

Figura 8. Uso de código de formatação, sequência de escape e funções de entrada e saída.

Leituras recomendadas

PAES, R. B. *Introdução à Programação com a Linguagem C*. São Paulo: Novatec, 2016. 296p.

PINHEIRO, F. A. C. *Elementos de programação em C*. Porto Alegre: Bookman, 2012. 548p.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS