COMANDOS DE ENTRADA DE DADOS

Comandos de entrada

Função Scanf ()

No cotidiano do programador, além de exibir a escrita formatada na tela, é preciso ler os dados informados pelo usuário. Para isso, utilizam-se **comandos de entrada**, permitindo a leitura formatada, principalmente, a partir do teclado, como é o caso do **scanf()**.

A função **scanf()** permite que o valor informado pelo usuário seja armazenado em uma variável e, posteriormente, usado para diversos cálculos.

Para chamar essa função, basta passar dois parâmetros entre os parênteses. Observe:

```
#include <stdio.h>

void main(){

int numero;

printf("Entre com um número inteiro:\n");

scanf("%d", &numero);

}
```

Veja a seguir o que consiste cada um desses parâmetros:

O primeiro

É composto pela string que traz o formato de leitura, com **%d**, **%f** ou **%c** entre aspas.

O segundo

Armazena o valor recebido, sendo o nome dessa variável precedido de **&**.

É importante que você observe alguns detalhes:

- O formato de leitura se mantém igual ao da escrita na tela: **%d** para as variáveis do tipo **int**, **%f** para as do tipo **float** e **%c** para as do tipo **char**;
- Não vamos nos aprofundar, por enquanto, no porquê do uso do **&** antes do nome da variável. Saiba que não seguir essa recomendação pode causar consequências inesperadas;
- Não confunda o símbolo **&** (comercial) com o operador lógico **&&**;
- Não inclua o caractere especial '**\n**' na string parâmetro da função **scanf()**.

Vamos entender melhor como usar a função scanf()?

Observe o código:

```
#include <stdio.h>

int main(){

int numero;

printf("Entre com um número inteiro:\n");

scanf("%d", &numero);

return 0;

}
```

Ao término de sua execução, a variável **numero** armazenará o valor informado pelo usuário via teclado. Poderíamos incluir mais uma linha, após a função **scanf()**, para escrever na tela a confirmação do número armazenado.

Vale a pena você testar essa inclusão. Escreva a linha a seguir:

```
printf("O valor informado pelo usuário foi %d\n", numero);
```

A função scanf() também pode ler **mais de uma** variável simultaneamente. Para isso, você precisa colocar os símbolos de formato de leitura na quantidade desejada e indicar as variáveis correspondentes, que vão armazenar os valores recebidos. Vejamos a aplicação dessa função.

Observe o código a seguir. Se o usuário digitar 10<enter>2, você verá o seguinte:

```
#include <stdio.h>

int main(){

float dividendo, divisor;

printf("Entre com dois numeros reais:\n");

scanf("%f %f", &dividendo, &divisor);

printf("A divisao de %.2f por %.2f vale %.2f", dividendo, divisor,
dividendo/divisor);

return 0;

}
```

Se o usuário digitar duas letras, você verá o seguinte resultado no código abaixo:

```
#include <stdio.h>

int main(){

char ch1, ch2;

printf("Entre com duas letras:\n");

scanf("%c", &ch1);

scanf("%c", &ch2);

printf("As letras inseridas foram %c e %c.\n", ch1, ch2);

return 0;

}
```

Você sabe o que aconteceu? Por que não foi possível inserir a segunda letra? Por causa do teclado!

Ele armazena temporariamente tudo o que digitamos, mas não repassa instantaneamente para o sistema. Podemos digitar alguma letra e apagá-la com a tecla ***backspace*** (←), mas quando apertamos a tecla *enter*, o sistema recebe a letra que digitamos e o *enter*.

Esse armazenamento temporário ocorre no chamado *buffer* do teclado. Como as variáveis do exemplo anterior recebem caracteres, a letra e o *enter* são armazenados, respectivamente, em **ch1** e **ch2**. Por isso, ocorre esse comportamento inesperado.

Existem duas formas de evitar que isso aconteça: A primeira é que, quando antes do símbolo de formato de leitura, você pode utilizar a função **scanf()** com um espaço na **string**. Isso fará com que sejam ignorados caracteres especiais, como o *enter*.

Assim, o código seria alterado para:

```
scanf(" %c", &ch2);
```

Após a primeira chamada da função scanf(), efetue a limpeza do buffer do teclado com a seguinte instrução, caso seu sistema operacional seja o Windows:

```
fflush(stdin);
```

Caso seja usuário do Linux, utilize a função:

```
__fpurge()
```

Temos usado a função **scanf()** com os nomes das variáveis precedidos de **&**. Esse operador deve ser lido como o **endereço de**. Assim, quando passamos o

parâmetro **&numero** para a função **scanf**, estamos informando o endereço na memória da variável numero. Por essa razão, todas as variáveis dos tipos **char**, **int**, **float** e **double** devem ser precedidas de **&**.

Outra função que pode ser usada para a leitura de **char**, a partir do teclado, é a **getc**, traduzida do inglês como “pegar o caractere”. Dessa forma, se declararmos a variável:

char ch1;

Tanto getc (ch1); quanto scanf(“%c”, ch1); terão o mesmo efeito.

Vamos praticar

Você receberá agora uma série de práticas para realizar em seu ambiente de programação. Tente executá-las. Vamos lá!

Prática 1

```
#include <stdio.h>

int main(){

char inicial;

int idade;

printf("Entre com a sua idade e a sua inicial:\n");

scanf("%d %c", &idade, &inicial);

printf("Voce tem %d anos e seu nome comeca com %c\n",idade, inicial);

return 0;

}
```

Digite esse código no compilador de código abaixo; entre com os seguinte valores:

30

H

Após a execução dos códigos, o conteúdo exibido na tela será:

Voce tem 30 anos e seu nome comeca com H.

O valor 30 será armazenado na variável **idade**, enquanto a variável **inicial** guardará o caractere ‘H’.

Prática 2

Vamos considerar o seguinte código:

```
#include <stdio.h>
int main(){
char inicial;
int idade;
printf(“Entre com a sua idade e a sua inicial:\n”);
scanf("%d %c", idade, inicial);
printf("Voce tem %d anos e seu nome comeca com %c\n",idade, inicial);
return 0;
}
```

Digite esse código no compilador de código abaixo; entre com os seguintes valores:

30

H

Após a execução dos códigos, ocorrerá um erro e nada será exibido na tela. Isso aconteceu porque a função **scanf()** apresenta variáveis sem o operador **&**.

Prática 3

Vamos considerar o seguinte código:

```
#include <stdio.h>
int main(){
char inicial;
int idade;
printf("Entre com a sua idade e a sua inicial:\n");
scanf("%d", &idade);
scanf("%c", &inicial);
printf("Voce tem %d anos e seu nome comeca com %c\n", idade, inicial);
return 0;
```

}

Digite esse código no compilador de código abaixo; entre com os seguintes valores:

30

H

Após a execução dos códigos, o conteúdo exibido na tela será:

Voce tem 30 anos e seu nome comeca com.

Ao apertar 30 e enter, a variável idade armazenará o valor 30 e a variável inicial, o enter.