

Introdução à programação C/C++

aula prática 3¹

1 Lendo e imprimindo caracteres

1.1 Funções `getchar` e `putchar`

Para ler e escrever caracteres do teclado as funções de entrada e saída mais simples são `getchar` e `putchar`, que estão na biblioteca `stdio.h` e cujos protótipos são os seguintes:

```
int getchar(void);  
int putchar(int c);
```

O programa da listagem 1 mostra exemplos de uso destas funções, e o resultado é:

```
Entre com um algarismo entre 0 e 9.  
7  
O caractere lido foi o 7
```

Listagem 1

```
#include<stdio.h>  
int main(void) {  
    char c;  
  
    printf("Entre com um algarismo entre 0 e 9.\n");  
    c = getchar();  
    printf("O caractere lido foi o ");  
    putchar(c);  
  
    printf("\n");  
    return 0;  
}
```

1.2 Lendo e imprimindo cadeias com `scanf` e `printf`

O programa 2 mostra como ler e imprimir uma cadeia usando os comando `scanf` e `printf` respectivamente. Este programa ao ser rodado apresentará um comportamento muito peculiar. Teste para ver o que acontece.

¹ Este material é praticamente uma reprodução do produzido pelo Prof. Adriano Joaquim de Oliveira Cruz, disponível em <http://equipe.nce.ufrj.br/adriano/c/praticas.htm>

```
#define DIM 40
#include<stdio.h>
int main(void) {
    char nome[DIM]; /* linha de caracteres lidos */

    /* Entrada de dados do vetor */
    printf("Por favor, qual o seu nome?\n");
    scanf("%s", nome);
    printf("Sou um computador. Posso ajudá-lo %s?\n", nome);

    return 0;
}
```

1.3 Lendo e imprimindo cadeias com gets e puts

Uma outra função usada para ler vetores de caracteres é o `gets(v)`. No vetor `v` são colocados todos os códigos dos caracteres lidos excetuando-se o da tecla <Enter>, que não é armazenado sendo substituído pelo código NULL.

```
#include<stdio.h>
char *gets (char *str);
```

A função `gets` retorna `str` caso nenhum erro ocorra. Caso o final do arquivo seja encontrado antes de qualquer caractere ser lido, o vetor permanece inalterado e um ponteiro nulo é retornado. Caso um erro ocorra durante a leitura, o conteúdo do array fica indeterminado e novamente um ponteiro nulo é retornado.

A função `puts` tem o seguinte protótipo:

```
#include<stdio.h>
int puts (const char *str);
```

Ela imprime a cadeia apontada por `str`. O programa da listagem 3 é semelhante ao exemplo anterior com as funções `printf` substituídas por `puts`. Observe que a impressão sempre termina e passa para a próxima linha. A função `puts` retorna um valor positivo caso nenhum erro ocorra. Em caso de erro é retornado um valor negativo.

```
#define DIM 40
#include<stdio.h>
int main(void) {
    char nome[DIM]; /* linha de caracteres lidos */

    /* Entrada de dados do vetor */
    puts("Por favor, qual o seu nome?");
    gets( nome );
    puts("Alo ");
    puts( nome );
}
```

```
puts("Sou um computador, posso ajudá-lo?");

return 0;
}
```

2 Exercícios

2.1 Exercício 1

Rode o exemplo mostrado na listagem 1 e verifique como ele funciona. Teste diversos caracteres tais como: letras, algarismos, sinais de pontuação, enter, espaço em branco etc.

2.2 Exercício 2

Rode o exemplo mostrado na listagem 2 e verifique como ele funciona. Teste primeiro com um nome simples, depois use nomes completos. O que acontece?

2.3 Exercício 3

Rode o exemplo mostrado na listagem 3 e verifique como ele funciona. Quais as diferenças entre este programa e o mostrado na listagem 2?

2.4 Exercício 4

- Escreva um programa que leia uma frase de até 80 caracteres e a imprima.
- Escreva um programa que leia uma frase de até 80 caracteres e imprima a frase e seu tamanho.
- Escreva um programa que leia uma frase de até 80 caracteres e imprima a frase, seu tamanho e quantas letras minúsculas foram lidas.
- Escreva um programa que leia uma frase de até 80 caracteres e imprima a frase, seu tamanho, quantas letras minúsculas e quantas maiúsculas foram lidas.
- Escreva um programa que leia uma frase de até 80 caracteres e imprima a frase, seu tamanho, quantas letras minúsculas, quantas maiúsculas e quantos caracteres em branco foram lidos.
- Escreva um programa que leia uma frase de até 80 caracteres e imprima quantas vezes os pares `rr` ou `RR` ou `rR` ou `Rr` aparecem no texto.

2.5 Exercício 5

Escreva um programa que leia e imprima várias cadeias de caracteres até que uma cadeia com zero caracteres seja fornecida.

2.6 Exercício 6

Um palíndromo é uma palavra ou frase que tenha a propriedade de poder ser lida tanto da direita para a esquerda como da esquerda para a direita. Em um palíndromo, normalmente são

desconsiderados os sinais ortográficos (diacríticos ou de pontuação), assim como os espaços entre palavras.

Exemplos de palíndromos são:

Socorram-me, subi no ônibus em Marrocos
Anotaram a data da maratona
Assim a aia a missa

Escreva um programa que leia uma frase somente com letras (sem espaços ou sinais de pontuação) e imprima se ela é um palíndromo.

2.7 Exercício 7

Escreva um programa que leia uma frase de até 20 caracteres e a imprima após trocar todas as letras minúsculas para maiúsculas. Caracteres diferentes de letras minúsculas não podem ser modificados.

Dica. Para transformar uma letra minúscula, armazenada na posição *i* de um vetor chamado *frase*, para maiúscula use o comando mostrado na listagem 4.

Listagem 4

```
/*  
 * frase[i] - 'a' indica quantas letras a frente do a  
 * minúsculo esta a letra em frase [i]  
 * a soma de frase[i] - 'a' ao valor 'A'  
 * indica quantas letras a frente de 'A' vai ficar frase[i]  
 */  
frase[i] = frase[i] - 'a' + 'A' ;
```