Manipulação de Caracteres

Disciplina: Algoritmos e Programação
Curso: Tecnologia em Análise e Desenvolvimento de Sistemas

Professora: Mariza Miola Dosciatti
mariza@utfpr.edu.br

Verificar se caractere faz parte do alfabeto

```c
#include <stdio.h>

int main(void)
{
	char ch;

	printf("Digite um caractere: ");
	scanf("%c", &ch);

	if(ch >= 'A' && ch <= 'Z' || ch >= 'a' && ch <= 'z')
	{
		printf("Caractere faz parte do alfabeto\n");
	}
	else
	{
		printf("Caractere nao faz parte do alfabeto\n");
	}

	return 0;
}
```

Verificar se caractere é uma letra minúscula

```c
#include <stdio.h>

int main(void)
{
	char ch;

	printf("Digite um caractere: ");
	scanf("%c", &ch);

	if(ch >= 'a' && ch <= 'z')
	{
		printf("Caractere eh uma letra minuscula\n");
	}
	else if(ch >= 'A' && ch <= 'Z')
	{
		printf("Caractere nao eh uma letra minuscula\n");
	}
	else
	{
		printf("Caractere nao eh uma letra\n");
	}

	return 0;
}
```

Verificar se caractere é uma letra maiúscula

```c
#include <stdio.h>

int main(void)
{
	char ch;

	printf("Digite um caractere: ");
	scanf("%c", &ch);

	if(ch >= 'A' && ch <= 'Z')
	{
		printf("Caractere eh uma letra maiuscula\n");
	}
	else if(ch >= 'a' && ch <= 'z')
	{
		printf("Caractere nao eh uma letra maiuscula\n");
	}
	else
	{
		printf("Caractere nao eh uma letra\n");
	}

	return 0;
}
```

Converter letra para maiúscula

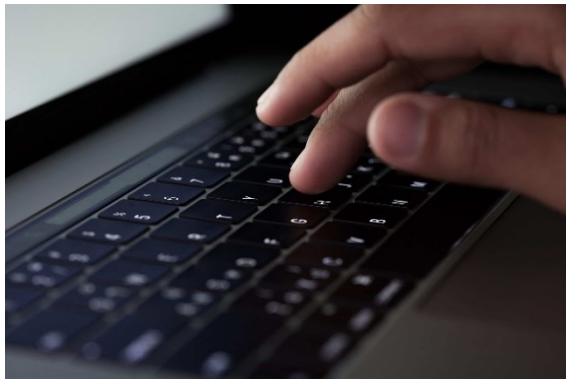
```c
#include <stdio.h>

int main(void)
{
    char ch;

    printf("Digite um caractere alfabetico minusculo: ");
    scanf("%c", &ch);

    if(ch >= 'a' && ch <= 'z')
    {
        printf("%c\n", ch - 'a' + 'A');
    }
    else
    {
        printf("%c\n", ch);
    }

    return 0;
}
```

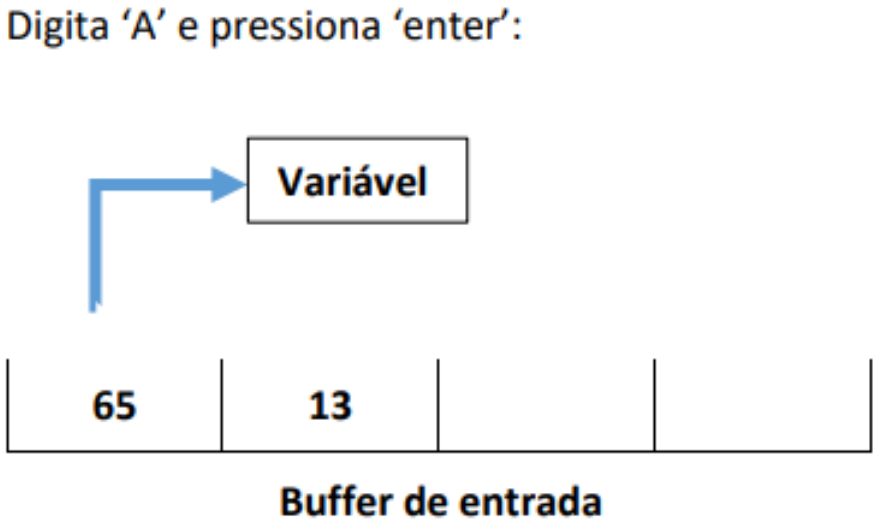
Converter letra para minúscula

```c
#include <stdio.h>

int main(void)
{
    char ch;

    printf("Digite um caractere alfabetico maiusculo: ");
    scanf("%c", &ch);

    if(ch >= 'A' && ch <= 'Z')
    {
        printf("%c\n", ch + ('a' - 'A'));
     }
     else
     {
         printf("%c\n", ch);
     }

     return 0;
 }
```

Buffer de Teclado

- Toda a informação que é digitada no teclado é armazenada em um buffer de entrada e fica disponível para ser utilizada.
- Quando é usada a função scanf(), ela recupera a informação do buffer. Porém, ela pode deixar “sujeira” no buffer, comprometendo futuras leituras.





Buffer de Teclado (cont.)

- Para resolver esse problema, é necessário limpar o buffer de teclado entre uma leitura e outra.
- Para isso é recomendável o uso da função setbuf com os parâmetros a seguir:

setbuf(stdin, NULL);

- Na função **setbuf()** o buffer da entrada padrão (**stdin**), ou seja, o teclado, é preenchido com o valor vazio (**NULL**).
- Na linguagem C a palavra **NULL** **é** uma constante padrão que significa um valor nulo.
- Um buffer preenchido com **NULL** é considerado limpo/vazio.

Buffer de Teclado (cont.)

- A função **fflush()** com o parâmetro **stdin** também vai limpar o buffer, mas tem um comportamento indefinido para o buffer de teclado.
- Ela é mais adequada para a limpeza do buffer de saída e não para o buffer de teclado, que é o buffer de entrada.
- Dependendo do sistema, a função **fflush()** pode funcionar para a limpeza do buffer, mas o seu comportamento indefinido faz com que ela possa funcionar adequadamente em uma máquina e não em outra.

Função setbuf()

- Exemplo de uso da função setbuf():

```c
#include <stdio.h>

int main(void)
{
    char ch1, ch2;

    printf("Insira um caractere: ");
    scanf("%c", &ch1);
    printf("Insira outro caractere: ");
    setbuf(stdin, NULL); //limpa o buffer do teclado
    scanf("%c", &ch2);

    printf("Voce digitou '%c' e '%c'", ch1, ch2);

    return 0;
}
```

Buffer de teclado em Linux

- Para limpar o buffer de teclado em Linux:

  __fpurge(stdin);

Referência

- Backes, André R. **Linguagem C**: Descomplicada. Disponível em: http://www.facom.ufu.br/~backes/publi_peq/apostilaC.pdf. Acesso em: 16 mar. 2021.