Ficha de Trabalho 8

Funções: Funções de tratamento de carateres: getchar, putchar, getche e getch e de manipulação de carateres: isdigit e toupper. **Função** setlocale.

CASTING

Atenção : Os cabeçalhos destas funções de tratamento de carateres estão incluídos no ficheiro ctype.h (getchar, toupper, isdigit) e no ficheiro conio.h (getchar, getch e getche), por isso, estas só são reconhecidas pela linguagem quando incluímos a diretiva do pré-processador que lhes está associada:

```
#include <ctype.h>
#include <conio.h>
```

Função getchar

A leitura do carateres pode ser feita utilizando a função scanf e a função getchar(), desenhada unicamente para a leitura de um caráter. Esta é invocada sem qualquer parâmetro. Lê um caráter e devolve o caráter obtido como resultado da função, evitando a escrita de parâmetros, formatos e a utilização de & .

getchar(): Esta função lê um caráter a partir da entrada padrão, normalmente o teclado.

Sintaxe: int getchar (void);

Esta função não retorna valores até que o caráter de controlo ('\n') seja lido. Este caráter, normalmente, é enviado pelo teclado quando a tecla [*enter*] é pressionada. Se forem digitados vários carateres, estes ficarão armazenados no *buffer* de entrada até que a tecla [*enter*] seja pressionada. Então, cada chamada da função *getchar*() lerá um caráter armazenado no *buffer*.

Programa exemplo:

```
#include <stdio.h>
main()
{
  char ch;
  printf("Introduza um carater");
  ch=getchar();
  printf("Carácter introduzido:  %c \n", ch);
}
```

putchar(): Envia um caráter para o ecrã.

Sintaxe: char putchar (int ch);

Programa exemplo:

```
#include <stdio.h>
#include<ctype.h>
main()
{
   char ch;
    printf("digita uma frase: \n");
    do
    {
        ch=getchar();
        putchar(toupper(ch));
     }
     while (ch!='\n');
}
```

getche()

A função *getche* permite ler um carácter do teclado. Difere da função getchar apenas porque não é necessário premir enter para que o caráter seja lido, a função devolve logo um valor. O caráter carriage Return [*return*], para a leitura de carateres.

Sintaxe: int getche (void);

getch()

A função *getch* é idêntica à função *getche*, também permite ler um caráter do teclado, mas ao invés desta não apresenta no ecrã o caracter lido (não se visualiza a introdução).

Sintaxe: int getch (void);

Programa exemplo: Qual o seu objetivo?

```
#include <stdio.h>
#include<conio.h>
main()
{
    int n=0; char ch;
    printf("digita uma frase: \n");
    do
    {
        ch=getche();
        n++;
     }
    while (ch!='\r');      /* o ciclo termina com [return]  o mesmo acontece de
                           *usarmos a função getch*/
    printf("\nForam digitados %d carateres!", n-1);
}
```

toupper(): Devolve o valor dum caráter transformado em maiúscula.

Sintaxe: `char toupper (char c);`

isdigit(): Esta função devolve o inteiro 1 caso c seja um dígito e o inteiro 0 caso contrário.

Sintaxe: `int isdigit(char c) ;`

Programa exemplo:

```
#include <stdio.h>
#include <ctype.h>
main()
{
    char ch;
    printf("Introduza um caráter de cada vez (para parar crtl c): \n");
    while(1)
    {
        ch=getchar();
        fflush(stdin);
        if(isdigit(ch))
      printf("%d é dígito! \n", ch);
        else
      printf("%d NÃO é dígito! \n", ch); // o que irá aparecer no ecrã,
     }                                    //caso o caráter seja 'A'?
 }
```

Nota: para que os carateres com acentuação ou cedilhados, datas, etc., sejam exibidos corretamente no output devemos incluir a diretiva do pré-processador **#include <locale.h>** e chamar a função **setlocale(argumento),** em que o argumento correspondente à configuração para a língua portuguesa.

```
#include <locale.h>
main()
…
setlocale(LC_ALL, "Portuguese");
```

Exercícios:

1- Escreve um programa, semelhante ao anterior, mas que imprima no ecrã os carateres introduzidos após convertidos para maiúsculas.

2- Escreve um programa que solicite ao utilizador dois carateres e que mostre no ecrã os carateres introduzidos.

3- Executa o código abaixo? Qual a função que simula?

```
#include <stdio.h>
#include <ctype.h>
main()
{
   char ch;
   ch=getchar();
   if (ch>='a' && ch <='z')
      putchar(ch + 'A' - 'a');      /*NOTA: 'A'-'a' = -32 */
}
```

Carateres e inteiros

Ao contrário da maioria das linguagens de programação, os carateres em C não são mais do que valores inteiros guardados num único byte. Desta forma, todas as operações realizadas sobre inteiros são também aplicadas ao tipo *char*.

Depois de declarada uma variável do tipo *char*

```
char ch;
```

existem diversas formas de colocar, por exemplo, o caráter 'A' nesta variável.

```
ch = 'A';
ch = 65; /* caráter cujo código ASCII é 65 */
ch = '\101' /* caráter cujo código ASCII em octal é 101 */ …
```

…estas instruções são equivalentes, pois em C, trabalhar com o caráter ou com o seu código ASCII é exatamente o mesmo. Só temos que ter alguns cuidados. Não devemos efetuar leitura de inteiros com formatos %c ou a leitura de carateres com formato %d, embora esses programas não apresentem erros de compilação, os resultados não são os esperados!

Exemplo:

```
    # include <stdio.h>
main()
 {
int num;          /*dever-se-ia ter declarado num como char */
printf("introduza um caráter :");
scanf("%c",  &num);
printf("O valor de num= %d cujo caráter = '%c'\n", num, num);
}                        /* O valor de num esperado é 65 */
```

Output: `O valor de num= 3137 cujo caráter = 'A'`

Casting

Sempre que temos numa variável ou expressão um valor de um determinado tipo e queremos, momentaneamente, modificar o tipo desse valor, indicamos o tipo que queremos entre parêntesis antes do identificador da variável ou expressão.

Sintaxe: (<tipo>) variável

Exemplos:

1. º

```
…
int n;
float f = 14.0;
n= (int) f % 2;
…
```

2. º

```
…
int i;
char ch = 'A';
i = (int) ch;
```

Exercício: Escreve um programa que solicite um inteiro (entre 0 e 255) ao utilizador e mostre o inteiro seguinte e o caráter correspondente. Usa uma variável inteira, outra do tipo caráter e casting.