

Programação Imperativa – Folha das aulas práticas nº 3

Temas: CÓDIGOS “ASCII” DOS CARACTERES. MANIPULAÇÃO DE FICHEIROS ATRAVÉS DA UTILIZAÇÃO DA ENTRADA PADRÃO; USO DAS INSTRUÇÕES `getchar` E `putchar`.

Nome _____ Ncd _____

Nota. Nos problemas propostos nesta folha os dados devem ser lidos com a função `getchar`. Usando o redireccionamento da entrada-padrão, os caracteres podem ser lidos de um ficheiro teste:

```
a.out < teste
```

De modo semelhante, pode redireccionar-se a saída padrão para um ficheiro.

1. *Códigos* A seguinte tabela (fim da página) foi impressa por um programa em C (com muito poucas linhas!) cuja instrução de impressão é “`printf(“%4d %1c\n”,c,c);`”.

- (a) Explique o efeito desta instrução.

Esta instrução imprime o caracter `c` num espaço de quatro caracteres e o código `ascii` correspondente num espaço de um caracter.

- (b) Escreva um programa que imprima uma tabela análoga.

```
#include<stdio.h>

main(){
int c;
for(c=32;c<=126;c++)
printf(“\%4d\t\%1c\n”,c,c); // imprime todos os caracteres
                             // correspondentes aos códigos ascii
                             // entre 32 e 126
}
```

Tabela (princípio e fim):

```
32
33  !
34  "
35  #
...  ...
122 z
123 {
124 |
125 }
126 ~
```

Notas. `getchar()` dá como resultado um inteiro que é o próximo caracter lido da entrada padrão ou EOF (que é igual a -1) se o ficheiro de leitura (no caso de redirecção da entrada padrão) já tiver acabado. A função `putchar(c)` tem como resultado enviar o inteiro “`c`” (entendido como um caracter) para a saída padrão (ou para o ficheiro para que esta estiver desviada.).

2. Quantas letras?

Pretende-se saber quantas letras existem num ficheiro.

- (a) Comece por escrever uma função

```
int letra(int c)
```

que retorna 1 se *c* é (o código ASCII) de uma letra (maiúscula ou minúscula) e 0 caso contrário.

```
int letra(char c)
  if (c>='a' && c<='z' || c>='A' && c<='Z'){
    return 1;
  }
  return 0;
}
```

-
- (b) Escreva um programa que, utilizando a função escrita na alínea anterior, imprima o número de letras existentes num ficheiro. Possível esquema do programa:

```
#include <stdio.h>

int letra(char c){
  if (c>='a' && c<='z' || c>='A' && c<='Z')
    return 1;
  return 0;
}

main(){
  int n, c;
  n=0;
  c=getchar(); // Lê o primeiro caracter
  while(c!=EOF){ // Enquanto o ficheiro não terminar
    if(letra(c)) // Se for letra, incrementa o contador
      n++;
    c=getchar(); // Lê o caracter seguinte
  }
  printf("%d letras\n",n); // Imprime o número de letras
}
```

3. *Conversas...*

Nas mensagens SMS é usual escrever “k” em vez de “qu”.

- (a) Escreva um programa que leia os caracteres de um ficheiro, trocando cada ocorrência de “k” por “qu”.

Uso do programa: a.out < fich1 > fich2

Exemplo: Se fich1 contém “ke keres?”, fich2 deverá passar a conter “que queres?”.

```
#include<stdio.h>

main(){
    int c;
    while((c=getchar())!=EOF)    // Enquanto o ficheiro não terminar
    {
        if(c=='k'){              // Se o caracter lido for 'k'
            putchar('q');        // Imprime 'q'
            putchar('u');        // Imprime 'u'
        }
        else
            putchar(c);          // Caso contrário, imprime o caracter lido
    }
}
```

- (b) Escreva um programa que efectue a transformação inversa, isto é, que substitua “qu” por “k”.

```
#include<stdio.h>

main(){
    int c,d;
    while((c=getchar())!=EOF)    // Enquanto o ficheiro não terminar
    {
        if(c=='q'){              // Se o caracter lido for igual a 'q'
            if((d=getchar())=='u') // Lê o próximo caracter e verifica se
                // é 'u'
                putchar('k');      // Se for, imprime 'k'
            else{
                putchar(c);        // Caso contrário, imprime 'q' e o
                // caracter d
                putchar(d);
            }
        }
        else
            putchar(c);          // Caso contrário, imprime todos os caracteres
    }                             // diferentes de 'q'
}
```

