

# MEFT - Programação

## 1º Ano - 1º Semestre de 2019/2020

### Série 8 (02/12/2019)

1. Construa um programa que calcula numericamente a solução da equação diferencial:

$$\frac{dx}{dt} = -Kx$$

o utilizador deve fornecer, na linha de comandos, o valor de  $K$ , o valor inicial de  $x$  ( $x_0$ ) e os valores iniciais e finais de  $t$ .

Com vista à representação gráfica dessa solução particular, escreva, num ficheiro, os resultados para que possam ser lido pelo 'Gnuplot', pelo 'Libre-Office', pelo 'OpenOffice' ou pelo 'Excel'.

2. Construa um programa que desenha no ecran um triângulo, como o representado na figura abaixo em que o número de linhas é fornecido pelo utilizador na linha de comandos.

```
      *
     * *
    * * *
   * * * *
  * * * * *
 * * * * * *
```

3. Triângulo de Pascal

a) Faça um triângulo análogo ao do problema anterior em que se substituem as '\*' pelos valores do triângulo de Pascal. Mostre-o no ecran. Para o cálculo das combinações utilize a expressão recursiva indicada na nota abaixo;

b) Guarde o valores do triângulo de Pascal num ficheiro binário;

d) Pergunte em seguida ao utilizador qual a combinação de que deseja saber o valor  $(n, k)$ , mostre o resultado no ecran e volte a perguntar. Inclua um processo de acabar o programa quando o utilizador não quer mais combinações. A obtenção das combinações deve ser feita a partir da leitura do ficheiro binário.

**Nota1:** Como é sabido os valores do triângulo de Pascal são as combinações de 'n' elementos 'k' a 'k', que podem ser obtidas a partir das expressões:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \prod_{i=1}^k \frac{n+1-i}{i} = \binom{n}{k-1} \frac{n+1-k}{k} \quad (k > 0)$$

em que a expressão do produtório só é válida para  $k > 0$ . Para  $k = 0$ , como é sabido, o resultado é '1'.

**Nota2:** O número de nós antes da linha **N** é dado por:  $Q_t = \frac{n(n+1)}{2}$ .

4. Faça uma implementação elementar da função 'ls' de unix (ou 'dir' do MS/DOS ou janela de comandos do Microsoft Windows), que mostra os ficheiros contidos numa directoria:

```
ls [-l] [ directoria ]
```

em que os [...] indicam que se pode omitir essa informação, "*directoria*" representa a pasta de que se pretende listar o conteúdo e "-l" significa que se deverá acrescentar a essa informação o número de bytes do ficheiro e a data em que foi alterado pela última vez.

O resultado final deverá ser uma lista em que cada linha apresenta um ficheiro diferente e, no caso da opção "-l", o seu aspecto deve ser o seguinte:

```
273635 Nov 16 09:34 ficheiro.ext
```

Para tal deverá usar as funções:

- **opendir** ('dirent.h'): Abre uma directoria para leitura. Vagamente equivalente a 'fopen' para ficheiros.
- **readdir** ('dirent.h'): cada vez que é chamada retorna um ponteiro (struct dirent \*) para a informação do membro da directoria em causa.
- **stat** (para windows) e **lstat** (para unix) ('sys/stat.h'): dão a informação referente a um dado ficheiro (struct stat).
- **closedir** ('dirent.h'): fecha uma directoria aberta com 'opendir'.
- **localtime** ('time.h'): recebe como argumento um ponteiro para o tempo 'time\_t' e retorna um ponteiro para a estrutura 'struct tm'.
- **strftime** ('time.h'): idêntica a *sprintf* para escrever o tempo numa string.

**Nota1:** Para mais detalhes sobre as funções ver manuais da biblioteca de C.

**Nota2:** o programa deverá poder ser compilado em unix ou em windows fornecendo os resultados correctos.