

Nome :

Número :

NOTA: Responda às questões números **1 e 2** na folha do enunciado e responda às questões números **3, 4 e 5** em folhas separadas.

Nas **questões 1 e 2 seguintes** classifique as afirmações apresentadas em verdadeiras (V) ou falsas (F)

1. Numa *shell*/linux um utilizador `userX` situado num diretório `~/deapc` deveria executar os seguintes comandos [2 valores]
  - a. `cp recurso.txt ./dir1` para copiar o ficheiro `~/deapc/recurso.txt` para `~/deapc/dir1`.
  - b. `chmod a+x msyscript.sh` para alterar as permissões do ficheiro `msyscript.sh` (ilustradas a seguir) de modo permitir a sua execução por todos os utilizadores do sistema, deixando as restantes permissões inalteradas  

```
-rw-r--r-- 1 userX grupoA 3290 2016-09-16 15:19 myscript.sh
```
2. Classifique as seguintes afirmações em verdadeiras (V) ou falsas (F). [2 valores]
  - a. `__` - As páginas WEB responsivas apenas podem ser apresentadas nos browsers de terminais móveis, eg. *tablets*.
  - b. `__` - O AJAX permite solicitar ao servidor os objetos de uma página individualmente.
  - c. `__` - Na interação entre um *browser* e o servidor, o método POST permite transmitir maior quantidade de bytes para o servidor.
  - d. `__` - Em *javascript*, a seguinte sequência de instruções irá colocar na variável `x` o valor "235": `a = 2; b = 3; c = "5"; x = a + b + c;`
3. Considere que o código de um programa `volumes`, desenvolvido em C, se encontra disperso pelos seguintes ficheiros:
  - `volumes.c` – que contém a função `main` que invoca funções existentes nos ficheiros `poliedros.c` e `naopoliedros.c`
  - `poliedros.c` – que contém o código das funções de cálculo de volumes dos sólidos poliedros;
  - `naopoliedros.c` – que contém o código das funções de cálculo de volumes dos sólidos não poliedros;
  - `poliedros.h` – que contém os protótipos das funções desenvolvidas em `poliedros.c`
  - `naopoliedros.h` – que contém os protótipos das funções desenvolvidas em `naopoliedros.c`.
  - a. Construa as regras da `Makefile` que permitam gerar os ficheiros objetos `poliedros.o`, `naopoliedros.o` e `volumes.o` e ainda, o programa `volumes`. [2 valores]
4. Desenvolva em linguagem C o programa `contaseqs` que conte o número de vezes que uma dada sequência de caracteres surge num texto com menos de 200 caracteres. A sequência a pesquisar e o texto a considerar são ambos passados ao programa pela linha de comando como parâmetros, como se exemplifica na seguinte execução do programa:

```
$/contaseqs -s de -t "Exame de deapc da época de recurso de 2017"
A sequencia "de" aparece 4 vezes!
```

A sequência a pesquisar é sempre precedida da opção `-s` enquanto que o texto onde será efetuada a pesquisa deve ser precedido da opção `-t` (considere que as opções serão sempre usadas pela ordem apresentada no exemplo anterior). Considere que o código C do programa `contaseqs` está dividido pelos ficheiros `main.c`, `contadores.c` e `contadores.h`, sendo que:
  - `main.c` – contém a função `main(...)` que recebe os argumentos da linha de comando e aloca um vetor com o espaço necessário para armazenar o texto. A função `main(...)` invoca a função `conta(...)` passando-lhe 3 argumentos: a sequência de caracteres a pesquisar, o vetor com o texto onde será efetuada a pesquisa, e a dimensão do vetor que contém o texto. Como retorno recebe da função `conta(...)` valor resultante da contagem que deverá imprimir antes de terminar de acordo com o formato ilustrado no exemplo acima.
  - `contadores.c` – contém o código da função `conta(...)` que recebe os argumentos passados pela função `main(...)`, efetua a contagem de ocorrências da sequência no texto passado no vetor e retorna o resultado dessa contagem.
  - `contadores.h` – contém a seguinte definição da função `conta`

```
int conta(char *seq, char* text, int tsize);
```
  - a. Desenvolva o código da função `conta(...)` de modo a que satisfaça os requisitos referidos acima. [3 valores]
  - b. Construa o ficheiro `main.c` de modo a que satisfaça os requisitos referidos acima. [3 valores]

5. Pretende desenvolver-se uma aplicação WEB que permita calcular o valor a pagar pelo estacionamento de uma viatura num parque. Essa aplicação deve ter uma página seleção da matrícula e classe da viatura e de inserção do tempo de estacionamento. Depois de preenchida e submetida a informação solicitada deverá calcular, de acordo com a tabela de preços seguinte, o valor a pagar pelo estacionamento da viatura em questão, e apresentar uma página com o resultado apurado.

Classe de viatura	Tempo de estacionamento (T)		
	$T \leq 1$ hora	1 hora < $T \leq 8$ horas	$T > 8$ horas
Moto	1 €/hora	0,8 €/hora	0,6 €/hora
Ligeiro	1,5 €/hora	1,3 €/hora	1 €/hora
Pesado	2 €/hora	1,7 €/hora	1,5 €/hora

- a. Construa o código HTML da página **parque.html** (ao lado) com o formulário que será usado para recolher os dados necessários ao funcionamento da aplicação. Considere que os dados recolhidos neste formulário serão enviados usando o método GET para a script **contas.php** que se encontra no diretório da página **parque.html** no servidor local. [2 valores]

- b. Desenvolva em PHP o script **contas.php** que processará os valores enviados pela página **parque.html** referida na alínea anterior, e apresente uma página com os resultados dos cálculos efetuados com a formatação ilustrada abaixo. [3 valores]

- c. Acrescente à página **parque.html** o código *javascript* necessário para assegurar que os campos estão todos preenchidos com valores válidos antes da informação ser enviada para a script **contas.php**. [3 valores]