

G1	Erros:	<pre>int i; n; r; arranjos; // separadores entre variáveis devem ser vírgulas, não ponto-e-vírgulas if(n&lt;r &amp;&amp; r&lt;1); // em vez de conjunção deve existir uma disjunção if(n&lt;r &amp;&amp; r&lt;1); // não deve haver um ponto-e-vírgula arranjos=0; // a variável acumuladora deve inicializar com o valor 1 while(i&lt;=n) // o while deve seguir-se uma chaveta, de início de bloco printf(" i=%d; arranjos=%d\n",arranjos,i); // argumentos trocados printf("Resultado: %d\n",n); // o resultado fica na variável arranjos, e não na variável n</pre>	
	Cotação:	<p>0,5 por cada erro detectado, até ao máximo de 3 valores (basta identificar 6 dos 7 erros).</p> <p>0,5 por erro introduzido não existente (cancela um erro bem identificado)</p> <p>Houve quem tivesse colocado o código corrigido, e quem indique os erros. Ambas as formas são válidas.</p> <p>0,3 por um erro identificado mas não corrigido</p> <p>0,2 erro detectado, mas corrigido de forma incorreta</p> <p>Apenas são considerados erros os que impedem o programa de funcionar corretamente. Erros identificados relativos a outras naturezas, são considerados erros mal assinalados, como a utilização de chavetas quando só há uma instrução, ou retornar sempre um número.</p> <p>Dois falsos erros relacionados com o mesmo assunto, desconta apenas uma vez</p> <p>1 valor por código completamente distinto e funcional, não sendo corrigido</p>	
G2		<p>Positivos (soma de 0 a 1,5)</p> <p>+ 0,5 - assinatura correta</p> <p>+ 0,5 - retorno da função correto (ou seja nada, tipo void)</p> <p>+ 0,5 - lógica relativamente correta</p> <p>Negativos (subtrai de 3 a 1,5)</p> <p>- 0,5 - erro no código (expressão numérica/lógica incorreta; instrução incorreta; instrução em falta)</p> <p>- 0 - um erro sintático (inclui troca de aspas por plicas e vice-versa)</p> <p>- 0,5 - dois ou mais erros sintáticos menores.</p> <p>Solução considerada mais simples:</p>	
		<pre>int Find(char *str, char *procura) { char *pt; pt = strstr(str, procura); if (pt != NULL) return pt - str; return -1; }</pre>	
G3			



		<pre> int main(int argc, char **argv) {     char str[256];     FILE *f;     int k, linha=0;      if (argc != 3)         printf("Coloque o nome de um ficheiro seguido da procura a realizar.");     else {         f = fopen(argv[1], "rt");         if (f != NULL) {             while (!feof(f)) {                 fgets(str, 256, f);                 linha++;                 if ((k=Find(str, argv[2]))&gt;=0)                     printf("\n%d:%s\nProcura localizada na posicao %d.", linha, str, k);             }              fclose(f);         }         else             printf("Ficheiro %s nao abre.", argv[1]);     } } </pre>								
G5	Alínea A)	Estruturas definidas para Livro, Genero, Local.								
		0,5 de penalização para uma só estrutura, ou utilização de nomes de tamanho fixo, ou data com um só número, ou um só registo (nem apontador para o próximo, nem vetor declarado na função main)								
		As penalizações vão até 1 valor, no caso de ser considerado uma boa tentativa, não passando de 1 valor.								
		0,5 valores é atribuído a tentativas mais fracas, em que não uma estrutura declararam corretamente, sendo classificados neste critério casos em que todos os nomes são declarados do tipo char (um só character)								
	Alínea B)	Programa com duas funções, uma para ler e outra para gravar a estrutura de dados definida na alínea anterior								
		0,5 de penalização para não verificação de ficheiro existente, ou não fechar o ficheiro, ou não abrir o ficheiro, ou alocação de memória bem sucedida, ou utilizar um nome fixo de ficheiro, ou não ler os registos todos, ou fazer só a função para gravar (e não fazer a de leitura), ou não alocar memória para todos os nomes, ou não gravar toda a estrutura de dados.								
		As penalizações vão até 1 valor, no caso de ser considerado uma boa tentativa, não passando de 1 valor.								
		0,5 valores é atribuído a tentativas com instruções incoerentes, mas com algumas instruções no sentido correto. Todas as resoluções de um só registo, são classificadas neste critério.								
		Função de leitura de um só registo, com um ficheiro fin e ficheiro fout, em que em fout colocam o nome dos campos, não faz qualquer sentido. Devem utilizar o ficheiro para entrada de dados em modo de leitura, e o ficheiro em para saída de dados em modo de escrita, um só ficheiro para cada operação. Vários estudantes com esta situação, devem ter consultado alguma atividade formativa que num outro contexto pode ser que faça sentido, aqui não. Tinham de definir duas funções, uma para ler toda a estrutura de dados para memória (o ficheiro teria de estar em modo de leitura), e outra para gravar toda a estrutura de dados para ficheiro (o ficheiro teria de estar em modo de escrita).								
	Alínea C)	Função que percorre a estrutura de dados e conta o número de livros, bem como outra função para contar o número de livros em cada ano, e também para extrair o menor/menor ano								

