**Questão única** O processo de seleção de uma empresa para diversos cargos envolve 2 etapas. Cada cargo possui notas mínimas para que o candidato seja classificado para a segunda etapa.

A primeira etapa do processo de seleção é composta de prova de títulos e 4 provas específicas. **Para cada candidato**, tem-se os seguintes dados, nesta ordem:

* código do candidato: inteiro;
* número de inscrição: inteiro;
* código do cargo desejado: inteiro;
* 5 notas (reais): nota da prova de títulos e nota em cada uma das 4 provas específicas;

O candidato será **desclassificado** se:

* pelo menos uma das 5 notas for 0 (zero) ou
* não atingir as notas mínimas necessárias, em cada uma das 5 provas, para o cargo desejado.

As notas mínimas necessárias, para classificação em cada um dos, no máximo 10, cargos estão armazenadas no arquivo CARGOS.TXT. **Para cada cargo**, haverá 2 linhas em sequência nesse arquivo, cada uma contendo, respectivamente:

* código do cargo: inteiro;
* 2 valores reais: nota mínima na prova de títulos e nota mínima nas provas específicas;

Faça um programa que, inicialmente leia os dados do arquivo CARGO.TXT. A seguir, o programa deverá ler, do teclado, os dados de cada um dos 1573 candidatos exibindo na tela os códigos dos candidatos que foram desclassificados com o motivo da desclassificação e gravando no arquivo ETAPA2.TXT o código do candidato, o código do cargo desejado e a nota final dos candidatos que se classificaram para a segunda etapa.

**Observações**:

1. Os dados de cada candidato classificado para a segunda etapa deverão estar em uma linha separada do arquivo.
2. Se o usuário digitar um código de cargo inválido, deverá ser exibida uma mensagem apropriada na tela e os demais dados deste candidato não deverão ser lidos.

**Obrigatoriamente**, o seu programa deverá utilizar uma matriz para armazenar as notas mínimas de cada cargo e as seguintes funções, desenvolvidas por você:

1. **(2,0 pontos) le\_dados\_cargos():** realiza a leitura dos dados do arquivo **CARGOS.TXT**, preenchendo os parâmetros recebidos e retorna a quantidade de cargos;
2. **(3,0 pontos) calcula\_nota\_final():** lê as 5 notas de **um** candidato do teclado, calcula a nota final armazenando-a na variável cujo endereço é fornecido na chamada da função. Esta função recebe, também, como parâmetros a nota mínima na prova de títulos e a nota mínima nas provas específicas do cargo desejado de um candidato. A função deverá retornar a quantidade de notas iguais a 0 (zero). Protótipo da função:

int calcula\_nota\_final (float mintitulos, float minespec, float \*notafinal);

**Cálculo da nota final:**

* + 0 (zero): caso alguma nota seja 0 (zero) ou caso alguma nota não tenha atingido a nota mínima necessária para o cargo desejado;
  + Média aritmética das 5 notas;

**Atenção**: os demais dados serão lidos na função main().

1. (0,5 ponto) **busca()**: implementa o algoritmo de busca sequencial.
2. (4,5 pontos) **main()** e demais funções (caso ache necessário). Não é obrigatório fazer outras funções além da main().

EXEMPLO DE RESULTADO NA TELA PARA 4 CANDIDATOS:

**ATENÇÃO:**

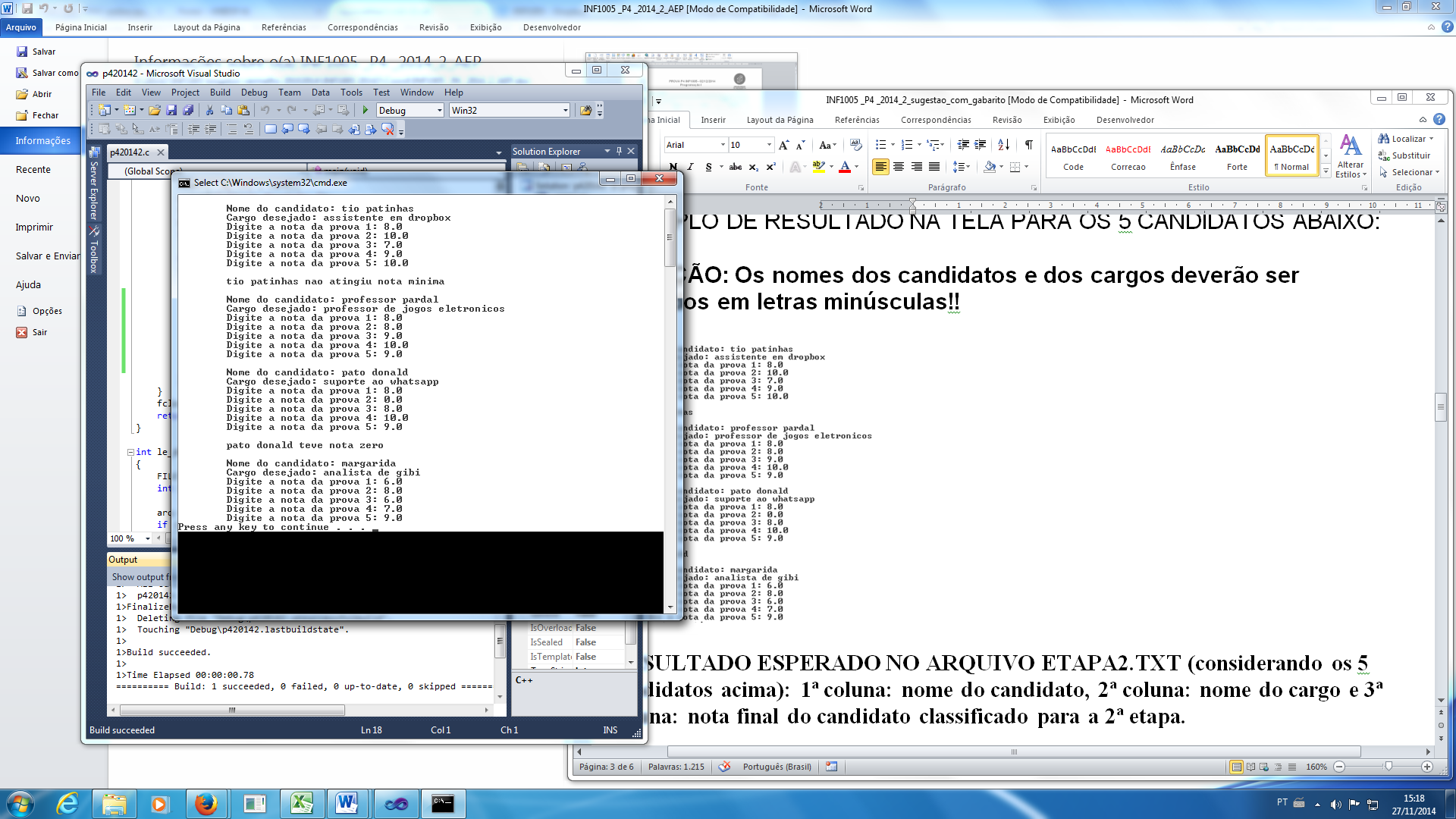
**Ao testar usando o exemplo abaixo, substituir**

**Tio patinhas por 777 e assistente em dropbox por 77666**

**Professor pardal por 202 e professor em jogos eletrônicos por 32165**

**Pato donald por 131 e suporte ao whatsapp por 99911**

**Margarida por 153 e analista de gibi por 10089**



**RESULTADO ESPERADO NO ARQUIVO ETAPA2.TXT (considerando os 4 candidatos acima): 1ª coluna: código dos candidatos, 2ª coluna: código dos cargos e 3ª coluna: notas finais dos candidatos classificado para a 2ª etapa.**

**202 32165 8.80**

**153 10089 7.20**