

O objetivo deste exercício é construir um programa capaz de automatizar a correção de uma prova de múltipla escolha. Para isso, você disporá de um arquivo de entrada chamado “gabarito.txt”, no qual estarão contidas as respostas corretas e o valor da pontuação de cada questão da prova. Um exemplo:

```
2 2.0
1 2.5
4 2.0
3 2.5
3 1.0
```

Nesse exemplo, tem-se uma prova com cinco questões, na qual: a resposta certa da primeira questão é a letra “B” (número 2) e a pontuação dessa questão é 2.0 pontos; a resposta certa da segunda questão é a letra “A” (número 1) e a pontuação é 2.5 pontos; e assim sucessivamente.

Dispomos também de um arquivo “respostas.txt”, no qual se coloca o número do aluno (um inteiro que o identifica) seguido de uma quantidade de número inteiros (entre 0 e 5) que é exatamente o número de questões da prova, representando a resposta dada pelo aluno àquela questão (0 representa resposta em branco ou nula). Exemplo:

```
23 2 3 4 0 3
1 3 3 3 3 3
...
```

Nesse exemplo, o aluno 23 respondeu à primeira questão com a letra B, a segunda com a letra C, a terceira com a letra D, deixou a quarta em branco e respondeu a última com a letra C.

- a) Implemente uma função auxiliar que leia arquivo “gabarito.txt” e preencha dois vetores: o de respostas e o de pontuação. Protótipo da função:

```
int le_gabarito(int gabarito[], float pontos[]);
```

A função deve abrir o arquivo de entrada, preencher os vetores e fechar o arquivo de entrada, retornando o número de questões da prova.

- b) Implemente uma função auxiliar que receba três vetores de inteiros e um número inteiro como parâmetros. O número representa quantas posições dos vetores estão preenchidas. Os dois primeiros vetores devem ser comparados e o terceiro vetor deve ser preenchido de tal forma que, em cada posição onde os dois primeiros vetores tem valores idênticos, coloca-se o valor 1; em cada posição na qual os dois primeiros vetores tem valores diferentes, coloca-se o valor 0. A função retorna o número de posições idênticas. Protótipo da função:

```
int compara(int v1[], int v2[], int v3[], int num);
```

- c) Implemente uma função auxiliar que leia uma “linha” do arquivo “respostas.txt” e armazene as respostas de um aluno (o próximo no arquivo) em um vetor de respostas. A função deve retornar um inteiro que é o número identificador do aluno, ou 0 (zero) caso o final do arquivo seja encontrado (todos os alunos já foram lidos). Protótipo da função:

```
int le_resp (FILE *arq_resp, int resp[], int num);
```

- d) Implemente uma função auxiliar que retorne a nota (float) de um aluno com base em dois vetores: o de acertos (que contém 1 nas posições onde a resposta é correta e 0 nas demais) e o de pontos de cada questão. Protótipo da função:

```
float calc_nota (int acertos[], float pontuacao[], int num);
```

- e) Implemente a função main() que use as funções acima para gerar um arquivo de saída “notas.txt”, no qual coloca-se o número identificador do aluno seguido de sua nota.