Universidade Federal do Espírito Santo – Departamento de Informática
Estruturas de Dados (INF092092) - Período: 2018/2
Segunda Prova Parcial - 29/11/2018
Profª Patrícia Dockhorn Costa
Email: pdcosta@inf.ufes.br

Regras importantes: A prova é sem consulta. Em caso de cola, a prova vale zero. A clareza e organização são importantes critérios de avaliação. Todo código deve ser claramente comentado.

Considere uma aplicação que tem por objetivo criar uma lista de Professores e outra lista de Alunos. A aplicação deve gerar um relatório dos alunos e professores lidos e, para cada um destes grupos, deve-se gerar médias de salários e de CR de professores e alunos, respectivamente.

O arquivo de entrada contém uma sequência de descrições dos alunos (identificados por A) e professores (identificados por P). Para aluno, as seguintes informações são esperadas: nome (simples, sem espaços), CPF e CR, nesta ordem. Para o professor, as informações são: nome (simples, sem espaços), CPF e salário.

O quadro a seguir mostra um exemplo de um arquivo de entrada:

```
A Joao 12345 8.43
A Maria 384739 9.50
P Maria 838928 1500.00
A Fulano 234556 2.30
P Sicrano 9876265 10000.00
P Fulano 736252 2000.00
A Jose 982736 3.00
P Alice 435261 8000.00
```

Escreva um programa que leia as informações de um arquivo "entrada.txt", o qual segue o formato descrito anteriormente, e gere um arquivo de saída com o nome "saida.txt" com as informações dos alunos e professores, como mostrado no arquivo de saída abaixo:

```
PROFESSORES
   Maria, CPF: 838928 e Salário: 1500.00
   Sicrano, CPF: 9876265 e Salário: 10000.00
   Fulano, CPF: 736252 e Salário: 2000.00
   Alice, CPF: 435261 e Salário: 8000.00

Média de salário dos 4 professores: 5375.00

ALUNOS
   Joao, CPF: 12345 e CR: 8.43
   Maria, CPF: 384739 e CR: 9.50
   Fulano, CPF: 234556 e CR: 2.30
   Jose, CPF: 982736 e CR: 3.00

Média de CR dos 4 alunos: 5.81
```

Regras importantes:

1) Os professores e alunos manipulados pelo cliente devem, necessariamente, ser armazenados em uma **lista genérica**;

2) O cálculo das médias **NÃO** pode ser realizado durante a leitura do arquivo. Primeiramente todos os alunos e professores devem ser armazenados em suas respectivas **listas genéricas**. Uma vez

que as listas tenham sido construídas, as mesmas devem ser percorridas (com o auxílio da função **percorre** implementada no TAD lista). É nesse momento que as médias devem ser calculadas;

3) As inserções devem ser feitas ao final da lista;

4) Utilize tipos de dados estruturados;

5) Escolha estruturas de dados que foram vistas na disciplina;

6) **NÃO** utilize variáveis globais;

7) Separe sua prova em TADs (listagen.h, listagen.c, seunome.c);

8) **Libere** toda a memória alocada (faça *callbacks* para conseguir liberar os dados dos alunos e professores no cliente);

9) O arquivo com a função main deve ter o seu nome: <nome>.c.

BOA PROVA!

Dicas :

Funções de manipulação de strings <string.h>:

- strlen: Retorna o tamanho da string (sem o '\0').
- strcpy: Faz a cópia de strings, desde que a string destino tenha tamanho suficiente. strcpy (destino, origem).
- strcmp: Compara duas strings. strcmp (str1, str2). Retorna:
  - 0 se str1 for igual a str2.
  - <0 se str1 for menor que str2 (como a sequência das palavras no dicionário)
  - >0 se str1 for maior que str2

Para compilar e gerar o executável

gcc –c arq1.c (compila o arquivo e gera arquivo arq1.o)
gcc –c arq2.c (compila o arquivo e gera arquivo arq2.o)
gcc arq1.o arq2.o(faz a linkagem e gera o executável a.out)

Sugestões para leitura/escrita de arquivo

- fopen (para abrir o arquivo)

```
...
FILE* fp;
fp = fopen ("entrada.txt", "rt");
if (fp == NULL) {
    printf ("Erro na abertura do arquivo!\n");
    exit(1);
}
...
```

- fclose (para fechar o arquivo)

```
int fclose (FILE* fp);
```

- fgetc (para ler caracter a caracter)

```
//exemplo de função que lê uma palavra
int le_palavra( FILE *fp, char *s ){
  int i = 0;
  int c;
  while ((c = fgetc( fp )) != EOF){
    if (isalpha(c))
      break;
  }
  if (c == EOF)
    return 0;
  else
    s[i++] = c;
  while (i<NPAL-1 && (c = fgetc(fp))!= EOF && isalpha(c))
    s[i++] = c;
  s[i] = '\0';
  return 1;
}
```

- fprintf (para escrever no arquivo)

```
fprintf(arquivo, %s, string)
fprintf(arquivo, %d, inteiro)
fprintf(arquivo, %.2f, float) //imprime 2 casas decimais
```