

Universidade Federal do Espírito Santo – Departamento de Informática
Estruturas de Dados (INF092092) - Período: 2018/2
Segunda Prova Parcial - 29/11/2018
Prof^a 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 retângulos. A aplicação deve gerar um relatório dos retângulos lidos e, para cada um, deve-se listar suas dimensões, bem como sua respectiva área. Além disso, ao final do arquivo deve ser exibida a média das áreas dos retângulos lidos. Para o cálculo das áreas, considere a seguinte fórmula:

$$\text{Área} = \text{Base} * \text{Altura}$$

O arquivo de entrada contém uma sequência de descrições dos retângulos, cada linha representando um retângulo com sua base e altura (dois números reais). O quadro a seguir mostra um exemplo de um arquivo de entrada:

```
10.00 20.00
1.00 0.50
20.00 5.00
4.00 5.00
1.00 1.00
1.00 2.00
2.00 3.00
```

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 retângulos. Para cada retângulo deve ser gerada uma primeira linha contendo apenas a palavra RETANGULO. As linhas seguintes, devem conter as dimensões do retângulo e sua área. Por fim, uma linha deve ser gerada informando a média das áreas dos retângulos. As inserções devem ser feitas ao final da lista. O arquivo de saída para o exemplo acima seria:

```
RETANGULO
  Base: 10.00
  Altura: 20.00
  Area: 200.00

RETANGULO
  Base: 1.00
  Altura: 0.50
  Area: 0.50

RETANGULO
  Base: 20.00
  Altura: 5.00
  Area: 100.00

RETANGULO
  Base: 4.00
  Altura: 5.00
  Area: 20.00

RETANGULO
  Base: 1.00
  Altura: 1.00
```

```
Area: 1.00
RETANGULO
Base: 1.00
Altura: 2.00
Area: 2.00
RETANGULO
Base: 2.00
Altura: 3.00
Area: 6.00
MEDIA RETANGULOS = 47.07
```

Regras importantes:

- 1) Os retângulos manipulados pelo cliente devem, necessariamente, ser armazenados em uma **lista genérica**;
- 2) O cálculo das áreas **NÃO** pode ser realizado durante a leitura do arquivo. Primeiramente todos os retângulos devem ser armazenados na **lista genérica**. Uma vez que a lista foi construída, a mesma deve ser percorrida (com o auxílio da função **percorre** implementada no TAD lista). É nesse momento que a área de cada um dos objetos geométricos deve ser calculada;
- 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 retângulos 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
```