Aulas teórico-práticas de Programação Estruturada

Folha de exercícios

2000/2001

1. *Espaços!*

   (a) Implemente uma função com o protótipo `void esp(char *s);` que troca todos os caracteres de `s` que não são letras nem dígitos por espaços. Por exemplo `"a, Ah Ah!"` fica transformado em`"a  Ah Ah "`.

   (b) Implemente uma função com o protótipo `char *esp1(char *s);` que retorna o "string" resultante da troca de todos os caracteres de `s` que não são letras nem dígitos por espaços (`s` não é alterado). O espaço para o resultado deve ser obtido através de `malloc`.

2. *Fracções ...*

   (a) Defina através de um `typedef` uma estrutura `FRAC` que seja apropriada para conter uma fracção com o numerador e o denominador inteiros. Deve ser incluida a seguinte informação: sinal, numerador (positivo), denominador (positivo), indicação de erro. Por exemplo, $-1/4$ é representado por

   ```
   Sinal: -1, Num: 1, Den: 4, Erro: 0 (não tem erro)
   ```

   (b) Defina uma função com o protótipo

   ```
   FRAC simp(FRAC f);
   ```

   que retorna o argumento simplificado.

   (c) Defina uma função com o protótipo

   ```
   FRAC soma(FRAC,FRAC);
   ```

   que retorna a soma dos dois argumentos.

   (d) Defina uma função com o protótipo

   ```
   FRAC sub(FRAC,FRAC);
   ```

   que retorna a diferença dos dois argumentos.

   (e) Defina uma função com o protótipo

   ```
   FRAC mult(FRAC,FRAC);
   ```

   que retorna o produto dos dois argumentos.

(f) Defina uma função com o protótipo

```
FRAC div(FRAC,FRAC);
```

que retorna o quociente do primeiro argumento pelo segundo.

(g) Implemente um programa completo que simule uma máquina de pilha para trabalhar com fracções e que tenha os comandos explicados ne seguinte função de ajuda:

```
void ajuda(void){
  printf("\nComandos:\n");
  printf("  q: fim (quit)\n");
  printf("  h: esta ajuda (help)\n");
  printf("  f: ler uma fracção para o topo da pilha (push do valor)\n");
  printf("  +: somar as duas fracções no topo da pilha\n");
  printf("  -: subtrair as duas fracções no topo da pilha\n");
  printf("  *: multiplicar as duas fracções no topo da pilha\n");
  printf("  /: dividir as duas fracções no topo da pilha\n");
  printf("  t: trocar o sinal à fracção no topo da pilha\n");
  printf("  e: trocar as duas fracções no topo da pilha\n");
}
```

Nas quatro operações aritméticas, os dois elementos que estão no topo da pilha são substituídos pelo resultado da operação, como exemplificado pela instrução (não "portátil" se a operação não for comutativa)

```
push(soma(pop(),pop()));
```