MAC110 **PROVA 2** *23/6/99*

1. [2 pontos]

(i) Escreva uma função de protótipo

```
int leia_v(int a[]);
```

que serve para ler uma seqüência de inteiros positivos (cujo término é indicado por 0). A sua função deve devolver o número de elementos na seqüência e os elementos lidos devem ser armazenados no vetor que é dado como argumento. Você pode assumir que o usuário não vai fornecer uma seqüência mais longa que o vetor pode armazenar.

(ii) Escreva uma função de protótipo

```
void imprima_v(int a[], int n);
```

que imprime os primeiros `n` elementos do vetor `a[]`.

2. [2 pontos] Escreva uma função de protótipo

```
int conte_ocorrencias(string p, string t);
```

que recebe como entrada os strings `p` e `t` e devolve o número de ocorrências de `p` em `t`. Nesta questão, você pode usar as funções da interface `strlib.h` de Roberts ou da interface `string.h` do ANSI C. Você pode achar útil o seguinte trecho de `strlib.h`:

```
/* Function: FindString
 * Usage: p = FindString(str, text, start);
 * ----------------------------------------
 * Beginning at position start in the string text, this function searches for
 * the string str and returns the first index at which it appears or -1 if no
 * match is found.
 */
int FindString(string str, string text, int start);
```

3. [3 pontos] Considere o seguinte programa:

```
#include <stdio.h>
#include "genlib.h"
#include "simpio.h"
#include "graphics.h"
void mark(double x, double h)
{ MovePen(1.0 + x, 1.0); DrawLine(0.0, h); }
void rule(double l, double r, int h)
{ double m = (l+r)/2;
  if (h > 0) { rule(l, m, h-1); mark(m, h); rule(m, r, h-1); }
}
main()
{ InitGraphics(); MovePen(1.0, 1.0); DrawLine(5.0, 0.0);
  rule(0, 5, 3);
}
```

Descreva a saída deste programa. Como se alteraria sua resposta se ao invés de `rule(0, 5, 3)` tivéssemos `rule(0, 5, 5)`?

4. [3 pontos] Existe uma fórmula de recorrência simples para determinar os numeradores e denominadores da seqüência de Farey (veja, por exemplo, Knuth, vol. 1, Seção 1.3.2, exercício 18, 3a. edição). Para a seqüência de Farey de ordem n (isto é, com denominadores $\leq n$), se os termos desta seqüência são x_0/y_0, x_1/y_1, $x_2/y_2, \ldots$, temos

$$x_0 = 0, \qquad y_0 = 1; \qquad\qquad x_1 = 1, \qquad y_1 = n;$$

e, para $k \geq 0$,

$$x_{k+2} = \lfloor (y_k + n)/y_{k+1} \rfloor x_{k+1} - x_k,$$

$$y_{k+2} = \lfloor (y_k + n)/y_{k+1} \rfloor y_{k+1} - y_k.$$

Acima, $\lfloor x \rfloor$ indica o maior inteiro não maior que x. Em `C`, existe uma função exportada em `<math.h>` chamada `floor()` que devolve $\lfloor$`x`$\rfloor$ quando chamada com argumento `x`. Escreva uma função de protótipo

```
void Farey(int n, int k, int *pnum, int *pden);
```

que recebe como entrada a ordem n da seqüência de Farey em que estamos interessados e põe em `*pnum` e `*pden` o numerador e denominador do `k`-ésimo elemento desta seqüência de Farey. Por exemplo, para `n = 7`, e `k = 5`, a função deve determinar a fração `2/7`.