

DIM0108.0 - 2011.1 - Turma 1 - Prova 02

David Déharbe - UFRN/CCET/DIMAp

19 de Maio de 2011

Considere a seguinte sub-rotina:

```
int w (int n)
{
    int i = 0;
    while (i * i <= n)
        i = i + 1;
    return i - 1;
}
```

Exercício 1 Preencha a tabela com os resultados de chamadas a w .

n	0	1	2	3	4	6	7	8	9	10	11	12	13	14	15	16
$w(n)$	0	1	1	1	2	2	2	2	3	3	3	3	3	3	3	4

Exercício 2 Descreva o papel da sub-rotina w .

Dado um número n , sub-rotina w retorna o menor inteiro positivo i tal que $(i + 1)^2 > n$. Logo $i^2 \leq n < (i + 1)^2$. O resultado portanto é a parte inteira da raiz quadrada de n , ou *raiz quadrada inteira* de n .

Exercício 3 Defina uma sub-rotina r equivalente a w que usa uma função recursiva $r2$: é proibido usar qualquer tipo de laço ou chamar outra função que não seja $r2$.

```
int r2 (int i, int n)
{
    if (i * i <= n)
        return r2(i+1, n)
    else
        return i - 1;
}
int r (int n)
{
    return r2(0, n);
}
```

Você foi contratado para escrever sub-rotinas que serão usadas para fazer estatísticas acerca do jogo de dados.

Exercício 4 Escreva uma sub-rotina que `conta6` que tem como papel contar o número de ocorrências de cada um dos lados em uma sequência de jogadas com um dado de seis lados. Os parâmetros de `conta6` são:

- `n`, o número de jogadas,
- `d`, um arranjo de inteiros entre 1 e 6 com o resultado de cada jogada, e
- `c`, um arranjo de 6 inteiros.

A sub-rotina deve preencher cada uma das seis posições de `c` com a quantidade de vezes que a jogada deu o número correspondente. Por exemplo, se `n = 7`, e o conteúdo de `d` for {1, 2, 1, 5, 6, 4, 5}, a chamada à sub-rotina `conta` deve preencher o arranjo `c` com os valores {2, 1, 0, 1, 2, 1}.

```
void conta6(int n, int d[n], int c[6])
{
    int i;
    /* inicializacao de c: o contador de cada lado eh zerado */
    for (i = 0; i < 6; ++i)
        c[i] = 0;
    /* contagem das ocorrencias: para cada elemento de d, incrementar o contador correspondente */
    for (i = 0; i < n; ++i) {
        int l = d[i];
        c[l-1] = c[l-1] + 1;
    }
}
```

Ou:

```
void conta6(int n, int d[n], int c[6])
{
    int i;
    /* inicializacao de c: o contador de cada lado eh zerado */
    for (i = 0; i < 6; ++i)
        c[i] = 0;
    /* contagem das ocorrencias: para cada elemento de d, incrementar o contador correspondente */
    for (i = 0; i < n; ++i)
        c[d[i]-1] += 1;
}
```

A empresa de dados resolveu expandir seu mercado e vender dados com um número de faces diferentes de seis. Você deve então escrever uma nova versão da sub-rotina desenvolvida no exercício anterior.

Exercício 5 Escreva uma sub-rotina `conta` que tem como papel contar o número de ocorrências de cada um dos lados em uma sequência de jogadas com um dado de `L` lados. Os parâmetros de `conta` são:

- `n`, o número de jogadas,
- `L`, o número de lados do dado,
- `d`, um arranjo de inteiros entre 1 e `L` com o resultado de cada jogada,
- `c`, um arranjo de `L` inteiros.

A sub-rotina deve preencher cada uma das `L` posições de `c` com a quantidade de vezes que a jogada deu o número correspondente. Por exemplo, se `L = 4`, `n = 7`, e o conteúdo de `d` for {1, 2, 1, 3, 2, 4, 1}, a chamada à sub-rotina `conta` deve preencher o arranjo `c` com os valores {3, 2, 1, 1}.

```
void conta(int n, int L, int d[n], int c[L])
{
    int i;
    /* inicializacao de c: o contador de cada lado eh zerado */
    for (i = 0; i < L; ++i)
        c[i] = 0;
    /* contagem das ocorrencias: para cada elemento de d, incrementar o contador correspondente */
    for (i = 0; i < n; ++i)
        c[d[i]] += 1;
}
```

O jogo do campo minado utiliza um tabuleiro retângular, onde cada célula é minada ou não. Neste jogo, o jogador deve determinar as células minadas. Para ele descobrir isto, ele pode saber, para cada célula não minada, quantas células vizinhas não são minadas.

Exercício 6 Defina uma sub-rotina, chamada `vizinhos` que tem como parâmetros `M` e `N`, as dimensões do tabuleiro, e duas matrizes `minas` e `num` ambas de `M` por `N` inteiros. A matriz `minas` contém, para cada célula, o valor 1 se for minada, e 0 se não for minada. A sub-rotina deve preencher a matriz `viz` com o número de vizinhos minados para as células não minadas e o valor 9 para as células minadas.

```
int testa (int M, int N, int minas[M][N], int i, int j)
{
    if (0 <= i && i < M && 0 <= j && j < N) {
        if(minas[i][j] == 1)
            return 1;
        else
            return 0;
    } else
        return 0;
}

void vizinhos (int M, int N, int minas[M][N], int num[M][N])
{
    int i, j;
    for (i = 0; i < M; ++i) {
        for (j = 0; j < N; ++j) {
            if (minas[i][j] == 1)
                num[i][j] = 9;
            else {
                num[i][j] =
                    testa(M, N, minas, i-1, j-1) +
                    testa(M, N, minas, i-1, j) +
                    testa(M, N, minas, i-1, j+1) +
                    testa(M, N, minas, i, j-1) +
                    testa(M, N, minas, i, j+1) +
                    testa(M, N, minas, i+1, j-1) +
                    testa(M, N, minas, i+1, j) +
                    testa(M, N, minas, i+1, j+1);
            }
        }
    }
}
```

Ou:

```
int testa (int M, int N, int minas[M][N], int i, int j)
{
    if (0 <= i && i < M && 0 <= j && j < N)
        return minas[i][j];
    else
        return 0;
}
```

Ou:

```
int testa (int M, int N, int minas[M][N], int i, int j)
{
    if (0 <= i && i < M && 0 <= j && j < N && minas[i][j] == 1)
        return 1;
    else
        return 0;
}
```