

O texto abaixo se refere às questões 1, 2, 3 e 4:

Um amigo seu, dono de uma tecelagem, tem uma máquina muito antiga, que é usada para costurar padrões vermelhos e azuis em faixas de tecido com 6 cm de largura. Prestes a se desfazer da máquina, seu amigo lembrou-se que ela continha um antigo disquete de 3.5". Com a ajuda de um velho computador, vocês conseguiram analisar o conteúdo do disquete, e para sua surpresa, ali estava a programação do padrão que a máquina deveria costurar – em uma versão interpretada da linguagem C! Este é o programa encontrado:

```
#define LARGURA_FAIXA 6 /* Largura da faixa de tecido, em cm. */

void main ()
{
    int i;

    /* Funciona até desligar ou o tecido acabar. */
    while (1)
    {
        for (i = 0; i < LARGURA_FAIXA; i++)
        {
            if (i == 1)
                pontoRolo1 ();
            else if (i == LARGURA_FAIXA-2)
                pontoRolo2 ();
            else
                moveAgulha ();
        }

        rolaTecido ();
    }
}
```

Analisando o código, você inferiu que o sistema tem algumas funções embutidas:

- pontoRolo1: a máquina faz um ponto de 1cm usando a linha do rolo 1 (vermelha).
- pontoRolo2: a máquina faz um ponto de 1cm usando a linha do rolo 2 (azul).
- moveAgulha: a máquina move a agulha 1cm para baixo, sem dar ponto.
- rolaTecido: a máquina leva a agulha novamente ao topo da faixa, sem dar ponto, e move a faixa de tecido 1 cm para a esquerda.

O padrão desenhado por este código pode ser descrito da seguinte forma (v indica um ponto com a linha vermelha, a um ponto com a linha azul, ( . . . ) que o padrão se repete indefinidamente):

[illegible]

(\*\*) 1. Seu amigo resolveu que pode ser interessante, em vez de se desfazer da máquina, modificar a sua programação. Mas como você não vai ficar desperdiçando material, resolveu analisar o comportamento do código que já estava lá no seu computador. Isso gera um problema: como fazer um teste se o código tem funções embutidas da máquina?

Seu primeiro objetivo é criar funções que possam simular no seu computador o comportamento da máquina. Para isso, você deve implementar “substitutas” para as funções `pontoRolo1`, `pontoRolo2`, `moveAgulha`, e `rolaTecido`. Você deve ser capaz de executar o programa original da máquina sem modificações, de forma que através das funções “substitutas” ele gere algo análogo ao padrão costurado.

(\*\*\*) 2. A pedido do seu amigo, reescreva o código da máquina para que ela costure o padrão abaixo:

[illegible]

(\*\*\*) 3. Reescreva o código da máquina para que ela costure um padrão de triângulos intercalados, como mostrado abaixo:

aaaaaaaaavvvvvvaaaaaaaaavvvvvvaaaaaaaaavvvvvvaaaaaaaaavvvvvvaaaaaaaaavvvvvvaaaaaaaaavvvvvv	(...)
aaaaaa vvvvvv aaaaaa vvvvvv aaaaaa vvvvvv aaaaaa vvvvvv aaaaaa vvvvvv aaaaaa v	(...)
aaaaa vvvvv aaaaa vvvvv aaaaa vvvvv aaaaa vvvvv aaaaa vvvvv aaaaa	(...)
aaa vvvv aaa vvvv aaa vvvv aaa vvvv aaa vvvv aaa	(...)
aa vv aa vv aa vv aa vv aa vv aa vv aa	(...)
a v a v a v a v a v a v a	(...)

(\*\*\*) 4. Seu amigo encontrou nos arquivos da tecelagem várias folhas de formulário contínuo contendo códigos-fonte para a máquina, impressos por uma impressora matricial. Mas o padrão gerado por um dos programas não está claro – o programador, talvez por inexperiência, pressa ou incompetência, não comentou o código, e os nomes das variáveis não ajudam muito.

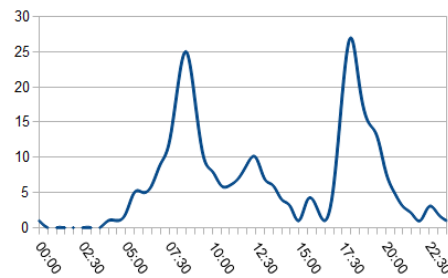
Análise o código e descubra como é o padrão que este programa gera. Use a mesma notação dos padrões das questões anteriores. Além disso, monte uma tabela contendo os valores das variáveis `p1`, `p2`, `c1` e `c2` no início do bloco do `while`, para as 12 primeiras iterações do `loop`. Ah, como vocês estão em um lugar sem computador e sem sinal de celular, você vai precisar analisar o código diretamente, sem executá-lo!

```

0      #define W 6
1
2      void main ()
3      {
4          int p1 = 0, p2 = W-1;
5          int c1 = 1, c2 = -1;
6
7          while (1)
8          {
9              int i;
10             for (i = 0; i < W; i++)
11             {
12                 if (i == p1)
13                     pontoRolo1 ();
14                 else if (i == p2)
15                     pontoRolo2 ();
16                 else
17                     moveAgulha ();
18             }
19
20             p1 += c1;
21             p2 += c2;
22
23             if (!p1 || p1 == W-1)
24                 c1 *= -1;
25             if (!p2 || p2 == W-1)
26                 c2 *= -1;
27
28             rolaTecido ();
29         }
30     }

```

5. (\*\*\*) A prefeitura de Curitiba está desenvolvendo um projeto-piloto para avaliar o uso do transporte público e testar melhorias no sistema. Um sistema de visão computacional analisa periodicamente as imagens enviadas por câmeras instaladas em estações tubo, contando quantos passageiros aguardam um ônibus. O gráfico abaixo exemplifica a ocupação de uma estação durante um dia.



A ideia do projeto é, com base em informações sobre a ocupação das estações, otimizar o uso da frota e estimular o uso do transporte público. Por exemplo, é possível avaliar o impacto de um aumento do número de ônibus em circulação, ou mesmo de uma mudança no preço da passagem.

Como parte do seu trabalho no projeto, você deve criar um pequeno programa em C para identificar certos padrões de comportamento na ocupação de uma estação tubo, que servirão para análises futuras. Os valores de entrada se referem ao número de passageiros em uma estação. Um padrão pode ser identificado agrupando-se esses valores 3 a 3, na ordem em que são gerados, e observando o segundo valor do grupo. Os padrões que devem ser identificados são:

- Pico: o valor é maior que ambos os vizinhos.
- Vale: o valor é menor que ambos os vizinhos.
- Subida: o valor é maior que o anterior, mas menor que o próximo.
- Descida: o valor é menor que o anterior, mas maior que o próximo.

Por exemplo, se a sequência de valores for 0, 5, 4, 3, 3, 1, 3, 3, 4, 6, temos:

- Um pico no segundo valor (0, **5**, 4).
- Uma descida do terceiro valor (5, **4**, 3).
- Um vale no sexto valor (3, **1**, 3).
- Uma subida no nono valor (3, **4**, 6).

O número total de valores é arbitrário. Para fornecer os valores (coletados em tempo real), outra pessoa da equipe escreveu a função `leProximoDado`, que deve ser chamada pelo seu programa para ler cada valor. A cada chamada, a função retorna o próximo dado, ou -1 se não existirem mais dados disponíveis. O protótipo da função é o seguinte:

```
int leProximoDado ();
```

Para registrar os padrões, você deve usar a função `registraPadrao`:

```
void registraPadrao (int posicao, int tipo);
```

Essa função recebe como parâmetros a posição do valor na série e um valor que indica o tipo de padrão observado. Os tipos de padrão são identificados pelas macros `PICO`, `VALE`, `SUBIDA` e `DESCIDA`, que têm valores inteiros. No exemplo acima, a função `registraPadrao` seria chamada 4 vezes, sendo que na primeira chamada teria como parâmetros 2 e `PICO`, e na última chamada teria 9 e `SUBIDA`. Pressuponha que as duas funções, assim como as macros, já estão definidas. Ou seja, você deve escrever apenas a função `main` que recebe os dados e registra os padrões.

**Atenção:** a função `main` não deve chamar as funções `scanf` e `printf`, ela deve usar as funções `leProximoDado` e `registraPadrao`. Você pode usar `printf` e `scanf` para simular o comportamento interno destas funções, já que não é possível criá-las como seriam na realidade. Também não use vetores, matrizes ou outras estruturas de dados que não foram vistas em aula.

(🐼🐼) 6. Escreva uma função que recebe como parâmetros dois inteiros positivos *a* e *b* e verifica se o menor deles é segmento do outro. A função deve retornar *ASEGB* se *a* for segmento de *b*, *BSEGA* se *b* for segmento de *a*, ou 0, caso contrário. *ASEGB* e *BSEGA* são duas constantes que devem ser definidas com a diretiva `#define`, cada uma com um valor diferente. A função criada para esta questão deve obrigatoriamente chamar a função criada na questão 2 da lista de exercícios de fixação. Exemplo.

a	b	retorno
1243	2212435	ASEGB
567890	678	BSEGA
235	236	0