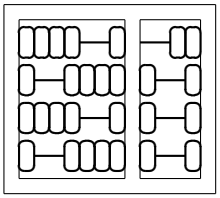

Laboratório 03 - *Triangle Wave*

Solução comentada

MC102 - Algoritmos e Programação de Computadores
Instituto de Computação - UNICAMP
Prof.: Hélio Pedrini

Solução por
Alexandre Medeiros – alexandre.medeiros@students.ic.unicamp.br

Problema

Dadas as amplitudes e frequências, gerar ondas triangulares correspondentes. A forma de onda triangular é caracterizada por uma ascendência linear até a amplitude máxima da onda, seguida imediatamente por uma descendência linear até a amplitude mínima. A Figura 1 mostra uma onda triangular.

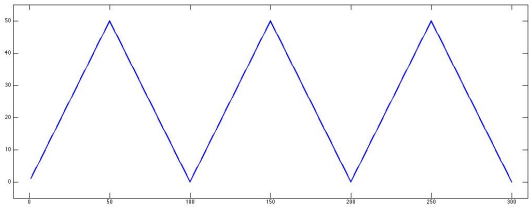

Figura 1: Onda Triangular

Crie um programa que recebe pares de amplitude e frequência, e imprima as ondas correspondentes.

Solução

Este é o primeiro laboratório que cobra o uso de laços e já cobra algo interessante, um laço que você só sabe quando parar quando a entrada tem um valor específico, o que permite o uso do comando `break`.

Para elaborar a solução desse laboratório, não há muito segredo, você precisará de um laço principal, onde você lerá a entrada e imprimirá sua respectiva onda. Esse laço pode ser feito de diversas maneiras, com um *do-while*, *while*. Para a minha solução, eu optei por fazer um programa que só lê dentro do laço e caso o valor lido seja o valor de saída, ele quebra o laço com um `break`, então esse meu laço principal não necessita de nenhuma condição, ele deve repetir indefinidamente. Para fazer isso, existem várias maneiras, as duas mais utilizadas são:

- *while*:

```
while (1) {
    ...
}
```

- *for*:

```
for (;;) {
    ...
}
```

Para a minha solução, eu optei por utilizar o *for*, que pode não ser intuitivo a primeira vista, porém tem uma notação simples.

Sabendo como fazer a leitura e eventualmente parar o programa quando necessário, só nos falta fazer a parte importante, para cada par de amplitude e frequência, imprimir sua onda triangular. O melhor jeito de se fazer isso é com laços!

Primeiramente, precisaremos de um laço para fazer cada "período" da onda. O jeito mais prático de se fazer isso é com um laço do tipo *for* com i variando de 0 até $f - 1$, a frequência da onda.

Agora para cada período, qual seria o jeito mais adequado de fazê-los? Como sabemos que as amplitude só variam de 1 até 9 poderíamos fazer um *switch-case* para isso e imprimir a onda para cada uma dessas possibilidades. Isso seria uma solução, mas não seria uma solução com um código muito limpo, além de não exercitar adequadamente o que aprendemos nas aulas.

Para fazer a impressão do período de uma maneira adequada, usaremos mais laços! Mas como usar laços para imprimir um "triângulo"? A melhor maneira de se fazer isso, seria dividindo a nossa tarefa em duas partes: parte crescente da onda e parte decrescente. Assim fica fácil pensar em como estruturar os laços.

Novamente usaremos laços *for*, com j variando de 1 até a, a amplitude, na parte crescente e com j variando de $a - 1$ até 1 na parte decrescente. Note que em cada iteração desses laços, teremos na variável j o valor da amplitude da onda no momento atual, ou seja, o número de caracteres e o número que precisamos imprimir.

Para imprimir a amplitude da onda podemos novamente utilizar *switch-case*, porém essa também não é a solução mais elegante. A solução mais interessante seria... LAÇOS! Agora precisamos apenas de um laço for onde a variável k varia de 0 até $j - 1$, ou seja, precisamos de j iterações.

Com o número de iterações completo, só precisamos imprimir nosso caractere agora, para isso podemos novamente utilizar *switch-case*, mas novamentes temos soluções melhores. Lembrem-se do laboratório anterior, onde vimos que a tabela ASCII nada mais é do que uma associação de números a caracteres e que os dígitos nessa tabela são consecutivos. Note agora que a variável j possui exatamente o número que queremos imprimir. Então, basta somar j ao caractere '0' e o que teremos? Precisamente o caractere do número que queremos imprimir. Logo para imprimir a onda, só precisamos utilizar a função `putchar`, não precisamos chamar a `printf`.

Para finalizar a linha, de cada amplitude, precisamos de uma quebra de linha, para isso, basta colocar um `putchar('n')` dentro do *for* que varia a variável j que teremos uma linha para cada amplitude.

E com isso, terminamos o laboratório 2 e temos nossas divertidas ondas triangulares no terminal.

Programa 1: Laço principal do programa, com todas as estruturas utilizadas para resolver o problema.

```
1   for (;;) {
2       /* Le amplitude e frequencia. */
3       scanf("%i%i", &a, &f);
4
5       /* Condicao de saida. */
6       if (a == 0 && f == 0)
7           break;
8
9       for (i = 0; i < f; ++i) {
10          /* Parte crescente da onda. */
11          for (j = 1; j <= a; ++j) {
12              for (k = 0; k < j; ++k)
13                  putchar('0' + j);
14              putchar('\n');
15          }
16          /* Parte decrescente da onda. */
17          for (j = a-1; j > 0; --j) {
18              for (k = 0; k < j; ++k)
19                  putchar('0' + j);
20              putchar('\n');
21          }
22
23          /* Quebra de linha entre ondas. */
24          puts("\n\n");
25      }
26  }
```