Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

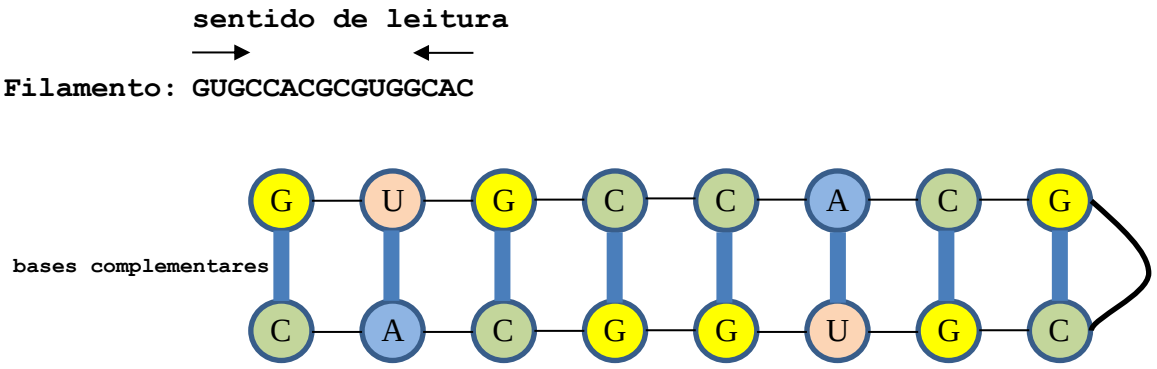
Curso de Tecnologia em Sistemas de Computação
Disciplina: Projeto e Desenvolvimento de Algoritmos
AD2 1° semestre de 2015

Nome:

1ª questão (valor 5.0)

Em Biologia, uma estrutura *hairpin* (grampo de cabelo) é um padrão que pode ocorrer em RNAs de cadeia simples em que um filamento, quando lido em sentidos opostos, contém somente pares de bases complementares.

A figura a seguir ilustra uma estrutura *hairpin*.



As bases complementares são listadas na tabela a seguir:

Base	Complemento
Adenina	Uracila
Guanina	Citosina
Uracila	Adenina
Citosina	Guanina

Uma estrutura *hairpin* deve ter um mínimo de quatro bases e, obviamente, um número par de bases.

Sua tarefa: Escreva o algoritmo da função **`ehHairpin(entradas: dna)`** que receba como parâmetro uma string representando um RNA e determine se esse RNA é uma estrutura *hairpin*. A função deve retornar **`verdadeiro`** se o RNA é uma estrutura *hairpin* e **`falso`** em caso contrário. Na string de entrada, o caractere **`'G'`** representa a base Guanina, o caractere **`'A'`** uma Adenina, o caractere **`'U'`** uma Uracila e o caractere **`'C'`** uma Citosina.

Em sua solução, você pode considerar a existência das funções **`charAt()`** e **`tamanho()`**, cuja documentação é mostrada a seguir:

`função charAt(entradas:` `str, pos)`

Retorna uma string contendo o caractere na posição **`pos`** da cadeia de caracteres **`str`** passada como parâmetro.

Exemplo:

```
imprima charAt('CEDERJ', 3)        // imprimiria 'D'
```

função tamanho(**entradas:** str)

Retorna o número de caracteres na string **str** passada como parâmetro.

Exemplos:

```
imprima tamanho('Dilma')     # imprimiria 5
```

Exemplos de uso da função.

```
imprima ehHairpin('GUGCCACGCGUGGCAC')    # imprime verdadeiro
imprima ehHairpin('GUGCCACGACGUGGCAC')   # imprime falso (número ímpar de bases)
imprima ehHairpin('GC')                  # imprime falso (menos de quatro bases)
imprima ehHairpin('CUCGCCAUCAAUAUUGAUGGCGAG')    # imprime verdadeiro
```

2ª questão (valor 5.0)

Uma famosa doceira usa um chocolate muito caro nas suas receitas. Este chocolate vem em barras retangulares de tamanhos variáveis. Para obter a quantidade exata para uma determinada receita, esta doceira realiza divisões sucessivas em uma barra até obter a quantidade requerida. Ela faz isto porque só aceita que existam barras retangulares na sua cozinha.

Por exemplo, para obter uma barra de 100 g a partir de uma barra de 3 kg, primeiro ela divide a barra ao meio, ficando com duas barras de 1,5 kg. Em seguida uma das metades é dividida em cinco partes iguais ficando com cinco barras de 300 g. Por fim, um desses pedaços de 300 g é dividido em 3 pedaços de 100 g, e a doceira conseguiu o seu pedaço de 100 g e todas as barras que sobraram são retangulares.

A sua tarefa é a partir de uma sequência de divisões realizada pela doceira, determinar quantos pedaços sobrarão no estoque para uso futuro.

A entrada do algoritmo contem as informações de uma sequência de divisões. A primeira linha da entrada contém um inteiro N que indica o número de divisões feitas na barra de chocolate original (1 <= N <= 1000). A linha seguinte contém N inteiros M (2 <= M <= 10) representando o número de pedaços em que o pedaço atual foi dividido. Sempre que uma divisão é realizada, um pedaço é utilizado para a próxima divisão e os restantes são separados para serem armazenados.

O seu programa deve imprimir o número de pedaços de chocolate que serão armazenados em estoque.

A seguir mostramos alguns exemplos de execução do algoritmo.

1º exemplo de execução (veja a Figura 1):

```
Quantos cortes?
3
Quantos pedaços?
2 3 5
Pedaços armazenados: 7
```

2º exemplo de execução:

```
Quantos cortes?
7
Quantos pedaços?
2 3 4 5 6 7 8
Pedaços armazenados: 28
```

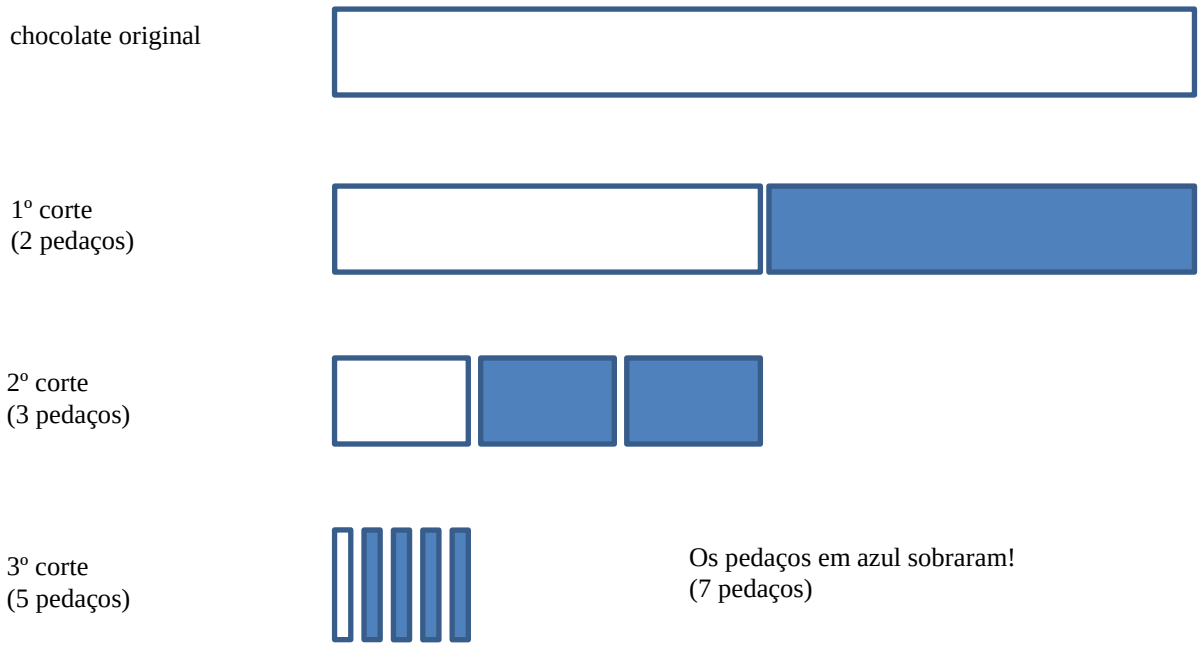


Figura 1 - Ilustração do 1º exemplo de execução