

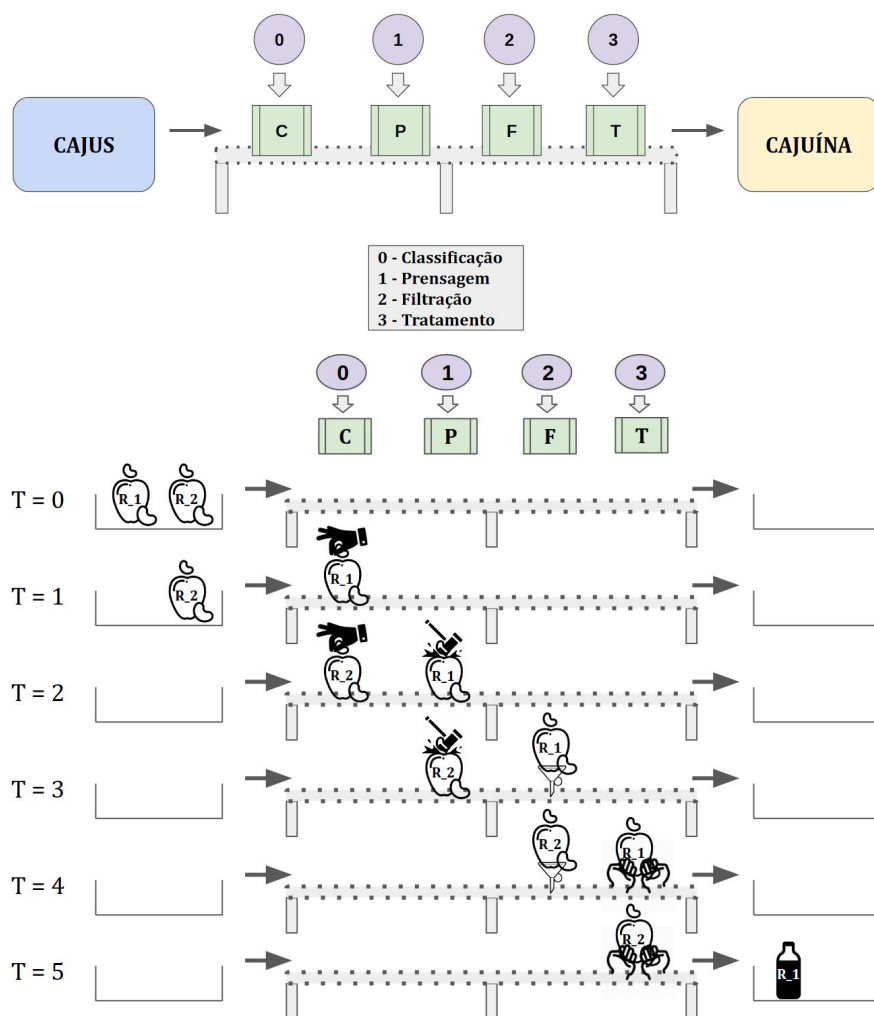
# MC102W - Algoritmos e Programação de Computadores

## Lab06: Fábrica de Cajuína

**Prazo:** 05 de Maio de 2020.

**Peso da Atividade:** 2

A Cajuína é uma bebida típica do nordeste, em especial do Piauí. Ela foi inventada por um farmacêutico chamado Rodolfo Teófilo para combater o alcoolismo. A bebida ficou famosa também por conta da canção feita pelo músico Caetano Veloso.



A Cajuína é preparada a partir do suco de caju, sem álcool, clarificada e esterilizada, apresenta uma cor amarelo-âmbar resultante da caramelização dos açúcares naturais do suco. Pode ser preparada de maneira artesanal ou industrializada. A bebida tem um método de fabricação que consiste em **4 etapas em sequência: Classificação, Prensagem, Filtração e Tratamento térmico**. O processo consiste na transformação de uma remessa de caju em cajuína. Veja a imagem acima.

# Tarefa

Nessa tarefa, você deverá implementar uma representação de uma fábrica de cajuína utilizando laços, listas e funções.

- Cada etapa deverá ter a sua própria **função**.

Ordem	Etapa	Processamento
0	Classificação	Envolve descartar caju de baixa qualidade, diminuirá em 1/3 da quantidade da remessa.
1	Prensagem	Converte o caju em suco de caju. O rendimento obtido é de 2x o valor da remessa antes da Prensagem se houver pelo menos 10 caju. Se houver menos do que 10 caju, então polpa de caju é adicionada a mistura, de forma que o rendimento é de 5x.
2	Filtragem	Diminui a quantidade do suco obtido e é ineficiente quando há muito suco. Assim, haverá a perda de 9/10 do suco se a quantidade de suco for maior do que 45 ( $> 45$ ) e diminuirá 1/9 caso contrário.
3	Tratamento	Por envolver adição de água, multiplicará por 2 o valor obtido na filtração.

- OBS.: Utilize conversão para inteiro em todas as operações que envolvem divisão.  
>> int (operação)
- A quantidade de caju de cada remessa será sempre **números inteiros maiores do que 1, pois é necessário pelo menos 2 caju para a produção de cajuína**.
- O funcionamento da máquina ocorre sequencialmente nas remessas, isto é (veja a figura na página anterior):
  - Tempo T=0: a máquina está vazia.
  - Tempo T=1: a 1ª remessa é enviada para a Classificação.
  - Tempo T=2: a 2ª remessa é enviada para a Classificação e a 1ª remessa é enviada para Prensagem.
  - Tempo T=3: a 3ª remessa é enviada para a Classificação, a 2ª remessa é enviada para Prensagem e a 1ª remessa é enviada para a Filtração.
  - Tempo T=4: a 4ª remessa é enviada para a Classificação, a 3ª remessa é enviada para Prensagem e a 2ª remessa é enviada para a Filtração e a 1ª remessa é enviada para o Tratamento.
  - Tempo T=5: a 5ª remessa é enviada para a Classificação, a 4ª remessa é enviada para Prensagem e a 3ª remessa é enviada para a Filtração e a 2ª remessa é enviada para o Tratamento e a 1ª está pronta.
  - E assim por diante, enquanto houverem remessas a serem processadas.

O programa deve ser em python e deve abranger os seguintes tópicos:

- ❑ Variáveis e tipos básicos;
- ❑ Expressões Lógicas e Relacionais;
- ❑ Comandos Condicionais;
- ❑ Listas;
- ❑ Funções;
- ❑ Comandos de Repetição;
- ❑ Entrada e saídas de dados.

## Entrada

O seu programa deve ter o **número de remessas e a quantidade cada remessa como entrada**. O número de remessas e as quantidades deverão **ser números inteiros maiores do que 1**.

## Saída

O seu programa deve gerar como saída **cada etapa de produção**, imprimindo a lista de remessas, a linha de produção e lista de cajuínas produzidas.

- As etapas de produção devem ser impressas após o processamento dos caju.
- Caso alguma as remessas **tenha menos que 2 caju**, imprima: “É necessário pelo menos dois caju para produção de cajuína!”

## Exemplos

A grafia da saída abaixo deve ser seguida rigorosamente por seu programa.

Exemplo 1:

### Entrada

```
4
2
3
4
5
```

### Saída

```
T=0 | [2, 3, 4, 5] -> [0, 0, 0, 0] -> []
T=1 | [0, 3, 4, 5] -> [1, 0, 0, 0] -> []
T=2 | [0, 0, 4, 5] -> [2, 5, 0, 0] -> []
T=3 | [0, 0, 0, 5] -> [2, 10, 4, 0] -> []
T=4 | [0, 0, 0, 0] -> [3, 10, 8, 8] -> []
T=5 | [0, 0, 0, 0] -> [0, 15, 8, 16] -> [8]
T=6 | [0, 0, 0, 0] -> [0, 0, 13, 16] -> [8, 16]
T=7 | [0, 0, 0, 0] -> [0, 0, 0, 26] -> [8, 16, 16]
```

```
T=8 | [0, 0, 0, 0] -> [0, 0, 0, 0] -> [8, 16, 16, 26]
```

## Exemplo 2:

### Entrada

```
5
10
20
30
40
50
```

### Saída

```
T=0 | [10, 20, 30, 40, 50] -> [0, 0, 0, 0] -> []
T=1 | [0, 20, 30, 40, 50] -> [6, 0, 0, 0] -> []
T=2 | [0, 0, 30, 40, 50] -> [13, 30, 0, 0] -> []
T=3 | [0, 0, 0, 40, 50] -> [20, 26, 26, 0] -> []
T=4 | [0, 0, 0, 0, 50] -> [26, 40, 23, 52] -> []
T=5 | [0, 0, 0, 0, 0] -> [33, 52, 35, 46] -> [52]
T=6 | [0, 0, 0, 0, 0] -> [0, 66, 5, 70] -> [52, 46]
T=7 | [0, 0, 0, 0, 0] -> [0, 0, 6, 10] -> [52, 46, 70]
T=8 | [0, 0, 0, 0, 0] -> [0, 0, 0, 12] -> [52, 46, 70, 10]
T=9 | [0, 0, 0, 0, 0] -> [0, 0, 0, 0] -> [52, 46, 70, 10, 12]
```

## Exemplo 3:

### Entrada

```
2
808
1
```

### Saída

```
É necessário pelo menos dois cajus para produção de cajuína!
```

## Exemplo 4:

### Entrada

```
3
25
20
30
```

### Saída

```
T=0 | [25, 20, 30] -> [0, 0, 0, 0] -> []
T=1 | [0, 20, 30] -> [16, 0, 0, 0] -> []
T=2 | [0, 0, 30] -> [13, 32, 0, 0] -> []
T=3 | [0, 0, 0] -> [20, 26, 28, 0] -> []
T=4 | [0, 0, 0] -> [0, 40, 23, 56] -> []
T=5 | [0, 0, 0] -> [0, 0, 35, 46] -> [56]
T=6 | [0, 0, 0] -> [0, 0, 0, 70] -> [56, 46]
T=7 | [0, 0, 0] -> [0, 0, 0, 0] -> [56, 46, 70]
```

## Exemplo 5:

### Entrada

```
1
9
```

### Saída

```
T=0 | [9] -> [0, 0, 0, 0] -> []
T=1 | [0] -> [6, 0, 0, 0] -> []
T=2 | [0] -> [0, 30, 0, 0] -> []
T=3 | [0] -> [0, 0, 26, 0] -> []
T=4 | [0] -> [0, 0, 0, 52] -> []
T=5 | [0] -> [0, 0, 0, 0] -> [52]
```

## Critérios específicos

Os seguintes critérios específicos sobre o envio e implementação devem ser satisfeitos.

i. Submeter no SuSy os arquivos:

⇒ `lab06.py`: Arquivo onde deverá ser implementada a tarefa.

## Observações gerais

No decorrer do semestre haverá 3 tipos de tarefas no SuSy (descritas logo abaixo). As tarefas possuirão os mesmos casos de testes abertos e fechados, no entanto o número de submissões permitidas e prazos são diferentes. As seguintes tarefas estão disponíveis no SuSy:

- ❑ **lab06-AmbienteDeTeste:** Esta tarefa serve para testar seu programa no SuSy antes de submeter a versão final. Nessa tarefa, tanto o prazo quanto o número de submissões são ilimitados, porém os arquivos submetidos aqui **não serão corrigidos**.
- ❑ **lab06-Entrega:** Esta tarefa tem limite de uma **única** submissão e serve para entregar a **versão final** dentro do prazo estabelecido para o laboratório. Não use essa tarefa para testar o seu programa e submeta aqui apenas quando não for mais fazer alterações no seu programa.
- ❑ **lab06-ForaDoPrazo:** Esta tarefa tem limite de uma **única** submissão e serve para entregar a versão final fora prazo estabelecido para o laboratório. Esta tarefa irá substituir a nota obtida na tarefa **lab06-Entrega** apenas se o aluno tiver realizado as correções sugeridas no *feedback* ou caso não tenha enviado anteriormente na tarefa **lab06-Entrega**.

### Avaliação

Este laboratório será avaliado da seguinte maneira: se o seu programa apresentar resposta correta para todos os casos de teste do SuSy, código de qualidade e comentários no código explicando o que está sendo feito, então é nota 10; caso contrário pontos serão descontados da nota final.

### Testando seu programa

Para testar se a solução do seu programa está correta, basta seguir o exemplo abaixo.

```
python lab06.py < arq01.in > arq01.out
diff arq01.out arq01.res
```

O `arq01.in` é a entrada e `arq01.res` é a saída esperada, ambos disponíveis no SuSy. O `arq01.out` é a saída gerada pelo seu programa. Após o prazo, os casos de teste fechados serão liberados e podem ser baixados e testados da mesma forma que os testes abertos.