



Salvar página completa



Salvar área visível

Clique e arraste, ou aponte e clique,
para selecionar uma região. Tecle ESC
para cancelar.

Cancelar

Atendimento

Questão 1

Parcialmente
corretoAtingiu 8,33 de
10,00

Santa Concorrência!

A nova nave do império, a temida BAT², está quase pronta para aniquilar a ameaça rebelde. Faltam apenas o núcleo de controle de processos concorrentes para que ela funcione a plena capacidade. O sistema operacional se encarrega de enviar as solicitações de processos, e você foi designado para implementar o controle da [ordem](#) de execução de partes dos processos utilizando a abordagem [round-robin](#). O sistema operacional já determina a fatia de tempo de cada solicitação: a execução de um comando especial.

Basta implementar as seguintes funcionalidades:

- **add** X adiciona uma solicitação com $X > 0$ processos à [fila](#) de execução. Este comando é seguido de X instâncias de um dos 3 comandos especiais listados a seguir, que devem ser executados na [ordem](#) em que são fornecidos para completar o processo.
 - **crypto** S define a chave criptográfica para envio de mensagens aos heróis do Império, visando garantir sua segurança ante espiões rebeldes. A partir da sequência S de não mais que 8 símbolos $+$ ou $-$ que indicam, respectivamente, que a [ordem](#) dos dígitos adjacentes deve ser crescente ou decrescente. Este comando gera o menor número possível com dígitos distintos.
 - **deYoda** W decodifica as [instruções](#) do [Mestre Yoda](#), fornecidas como uma sequência W de palavras (e pontuação), invertendo sua [ordem](#) para que façam sentido e, portanto, sirvam para as decisões táticas da BAT².
 - **merge** I facilita a perseguição aos rebeldes simplificando o espaço de busca: apresenta uma sequência de intervalos sem sobreposição, em [ordem](#), criada a partir dos I intervalos [dados](#) (que indicam as coordenadas onde se sabe que a paz do Império reina absoluta). Cada intervalo é definido por valores inteiros no formato $[x, y]$, separados por espaço, tais que $x \leq y$ (há um espaço após a vírgula entre x e y).
- **halt** interrompe a execução e mostra quantas solicitações ficaram órfãs (não foram finalizadas) e a quantidade total de comandos que não foram executados.
- **process** executa o próximo comando especial disponível conforme o [algoritmo](#) round-robin.

A comunicação com o sistema é simples, as funcionalidades são apresentadas como descritas acima pelo sistema operacional, e seu [programa](#) deve organizá-las e executá-las conforme solicitado pelas [instâncias superiores](#).

Entrada

A entrada consiste de uma série de comandos como os descritos. É garantido que há pelo menos um comando **halt**.

Saída

A saída deve ser a [informação](#) resultante da execução de um comando especial ou **halt**, conforme especificados.

Observações

No quarto exemplo, o primeiro **process** lida com o primeiro comando do primeiro pedido (que vai para o final da [fila](#) conforme o round-robin). Então insere-se um novo pedido com outros 2 comandos. Como antes, o **process** lida com o primeiro comando (do segundo pedido - cujo processo é finalizado), e aplica-se o round-robin. Um novo **process** executa o primeiro comando do terceiro pedido de processo (novamente o round-robin) e o último **process** executa o comando que termina o primeiro pedido de processo. Resta, então, um comando do terceiro pedido.

Atendimento

For example:

Input	Result
add 1 crypto --- process halt	4321 0 processo(s) e 0 comando(s) órfão(s).
add 2 crypto -- merge [1, 3] [3, 4] [5, 8] [6, 10] process process halt	2143 [1, 4] [5, 10] 0 processo(s) e 0 comando(s) órfão(s).
add 1 deYodafy contra-atacar deve Império! add 1 merge [1, 3] [6, 10] [3, 4] [5, 8] process process halt	Império deve contra-atacar! [1, 4] [5, 10] 0 processo(s) e 0 comando(s) órfão(s).
add 2 crypto +++ merge [1, 3] [6, 10] [3, 4] [5, 8] add 1 deYodafy faça não ou faça. process add 2 crypto -- merge [1, 7] [2, 5] [5, 10] [8, 9] process process process halt	13254 faça ou não faça. [1, 4] [5, 10] 2143 1 processo(s) e 1 comando(s) órfão(s).

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 ▼ def crypto(chave):
2     lista_sequencia = [1]
3     numero = [] # inicializando as listas que vamos armazenar cada variável
4 ▼     for i in range(len(chave)):
5         numero.append(i + 2) # adicionando na lista numero ignorando o 1
6         contador = 0

```

Atendimento

Verificar

	Input	Expected	Got	
✓	add 1 crypto --- process halt	4321 0 processo(s) e 0 comando(s) órfão(s).	4321 0 processo(s) e 0 comando(s) órfão(s).	✓
✓	add 2 crypto --+ merge [1, 3] [3, 4] [5, 8] [6, 10] process process halt	2143 [1, 4] [5, 10] 0 processo(s) e 0 comando(s) órfão(s).	2143 [1, 4] [5, 10] 0 processo(s) e 0 comando(s) órfão(s).	✓
✓	add 1 deYodafy contra-atacar deve Império! add 1 merge [1, 3] [6, 10] [3, 4] [5, 8] process process halt	Império deve contra-atacar! [1, 4] [5, 10] 0 processo(s) e 0 comando(s) órfão(s).	Império deve contra-atacar! [1, 4] [5, 10] 0 processo(s) e 0 comando(s) órfão(s).	✓

Atendimento

	Input	Expected	Got	
✓	add 2 crypto ++- merge [1, 3] [6, 10] [3, 4] [5, 8] add 1 deYodafy faça não ou faça. process add 2 crypto +- merge [1, 7] [2, 5] [5, 10] [8, 9] process process process halt	13254 faça ou não faça. [1, 4] [5, 10] 2143 1 processo(s) e 1 comando(s) órfão(s).	13254 faça ou não faça. [1, 4] [5, 10] 2143 1 processo(s) e 1 comando(s) órfão(s).	✓
✓	process halt	0 processo(s) e 0 comando(s) órfão(s).	0 processo(s) e 0 comando(s) órfão(s).	✓
✓	add 1 crypto - halt add 1 crypto + halt	1 processo(s) e 1 comando(s) órfão(s).	1 processo(s) e 1 comando(s) órfão(s).	✓
✓	add 1 merge [1, 7] [2, 5] [5, 10] [8, 9] add 1 merge [1, 7] [2, 5] [5, 10] [8, 9] halt	2 processo(s) e 2 comando(s) órfão(s).	2 processo(s) e 2 comando(s) órfão(s).	✓
✓	add 2 deYodafy usar_a_Força você_deve! deYodafy você_deve usar_a_Força! halt	1 processo(s) e 2 comando(s) órfão(s).	1 processo(s) e 2 comando(s) órfão(s).	✓
✓	add 4 deYodafy usar a Força você deve! deYodafy usar_a_Força você_deve! deYodafy usar_a_Força você_deve! deYodafy usar_a_Força_você_deve! process process process process halt	deve você Força a usar! você_deve a_Força usar! você_deve usar_a_Força! usar_a_Força_você_deve! 0 processo(s) e 0 comando(s) órfão(s).	deve você Força a usar! você_deve a_Força usar! você_deve usar_a_Força! usar_a_Força_você_deve! 0 processo(s) e 0 comando(s) órfão(s).	✓

Atendimento

	Input	Expected	Got	
✓	<pre> add 2 deYodafy lado_sombrio_da_Força cuidado tenha. deYodafy Hmm? process add 1 deYodafy Luke seu_tio sou não. process halt </pre>	<pre> tenha cuidado lado_sombrio_da_Força. Hmm? 1 processo(s) e 1 comando(s) órfão(s). </pre>	<pre> tenha cuidado lado_sombrio_da_Força. Hmm? 1 processo(s) e 1 comando(s) órfão(s). </pre>	✓
✓	<pre> add 3 crypto + crypto +-+--+ crypto --+-- process process process halt </pre>	<pre> 12 132547698 3214765 0 processo(s) e 0 comando(s) órfão(s). </pre>	<pre> 12 132547698 3214765 0 processo(s) e 0 comando(s) órfão(s). </pre>	✓
✓	<pre> add 1 merge [-5, 5] [13, 42] [10, 100] process add 2 merge [1, 2] [2, 3] [3, 4] [4, 5] merge [4, 5] [2, 3] [1, 2] [3, 4] process halt </pre>	<pre> [-5, 5] [10, 100] [1, 5] 1 processo(s) e 1 comando(s) órfão(s). </pre>	<pre> [-5, 5] [10, 100] [1, 5] 1 processo(s) e 1 comando(s) órfão(s). </pre>	✓
✓	<pre> add 3 crypto +--+ deYodafy nossa bossa? merge [1, 2] [2, 2] [2, 3] [3, 4] process process process halt </pre>	<pre> 14325 bossa nossa? [1, 4] 0 processo(s) e 0 comando(s) órfão(s). </pre>	<pre> 14325 bossa nossa? [1, 4] 0 processo(s) e 0 comando(s) órfão(s). </pre>	✓

Your code failed one or more hidden tests.

Parcialmente correto

Notas para este envio: 8,33/10,00.

Atendimento



ATIVIDADE ANTERIOR
[Questionário] Recursão

PRÓXIMA ATIVIDADE
[Estudo] Ordenação e Busca



Atendimento