Parabéns! Bem vindo a segunda fase do processo seletivo da **IdeenBox**.

Nessa fase você irá fazer algumas atividades relacionadas às competências necessárias para a vaga. Avançar para a próxima fase dependerá do resultado final destas atividades.

Para conhecimento, as etapas dessa seleção são:

1. Triagem de currículos
2. Atividades **(estamos aqui!)**
3. Entrevista (caso necessário, podemos dividir essa fase em duas entrevistas);
4. Seleção final

Para entrega das atividades dessa fase, você tem até o dia **04 de outubro de 2015 às 23:59**.

Como todas as atividades são de programação, você deve disponibilizar todos os códigos fontes em um único repositório no **GitHub** (<https://github.com/>) e nos enviar o link através do formulário a seguir.

**Formulário para envio de resposta:** [**http://goo.gl/forms/jUQ2MGDcIR**](http://goo.gl/forms/jUQ2MGDcIR)

As atividades dessa etapa estão descritas nas próximas páginas.

Boa sorte!

**Equipe IdeenBox**

**Bruce e o Array**

Bruce dá a Steve um vetor A de tamanho N. Então ele pede para que Steve determine se existe um elemento no vetor tal que a soma dos elementos da sua esquerda seja igual a soma dos elementos da sua direita. Se não existir elementos à esquerda ou direita, então a soma é considerada zero.

Formalmente, ele quer que Steve encontre um *i* tal que A1+A2...Ai-1 = Ai+1+Ai+2 … AN

**Formato da Entrada**

A primeira linha contém T, que representa o número de casos de teste. Para cada caso de teste, a primeira linha contém N, que representa o número de elementos do vetor A do caso de teste. A segunda linha para cada caso de teste contém N inteiros positivos separados por espaço, denotando o vetor A.

**Formato da Saída**

Para cada caso de teste imprima "SIM" na tela se existir um elemento no vetor com as características descritas acima. Caso contrário, imprima "NÃO".

**Restrições**

1 ≤ T ≤ 10

1 ≤ N ≤ 105

1 ≤ Ai ≤ 2 × 104

1 ≤ i ≤N

**Exemplo de Entrada**

2

3

1 2 3

4

1 2 3 3

**Exemplo de Saída**

NÃO

SIM

**Explicação**

No primeiro caso de testes, nenhum elemento com as características desejadas existe.

No segundo caso de testes, A[1]+A[2]=A[4], portanto o índice 3 satisfaz as condições exigidas.

**Dicas**

- Forneça a solução mais eficiente que você encontrar

- Use **Python** para resolver o problema

- Considere que as entradas são digitadas na entrada padrão, não em arquivos texto

**Marco e os Brinquedos**

Marco e Janine estão muito felizes por ter seu primeiro filho. O filho deles adora brinquedos, então Marco quer comprar alguns. Existem N tipos de brinquedos na loja que ele deseja comprar. Todos estão com a etiqueta que corresponde ao seu preço. Marco só pode gastar K reais com brinquedos sem comprometer seu orçamento mensal. Ele quer maximizar o número de brinquedos que ele pode comprar com esse dinheiro.

Você é o melhor amigo de Marco e quer o ajudar a comprar o maior número possível de brinquedos com K reais.

**Formato da Entrada**

A primeira linha contém dois inteiros, N e K, respectivamente. A segunda linha contem N inteiros separados por espaço indicando o preço de cada um dos N tipos de brinquedos da loja.

**Formato da Saída**

Um inteiro que denota o número máximo de brinquedos que Marco pode comprar para o seu filho.

Restrições

1 <= N <= 105

1 <= K <= 109

1 <= preço de qualquer brinquedo <= 109

Um brinquedo não pode ser comprado múltiplas vezes

**Exemplo de Entrada**

7 50

1 12 5 111 200 1000 10

**Exemplo de Saída**

4

**Explicação**

Ele pode comprar, no máximo, 4 brinquedos. Esses brinquedos são os de preço 1, 12, 5 e 10.

**Dicas**

- Forneça a solução mais eficiente que você encontrar

- Use Python para resolver o problema

- Considere que as entradas são digitadas na entrada padrão, não em arquivos texto

**Nome de Usuário Válido**

Em um sistema com desenvolvedores de gosto duvidoso, os nomes dos usuários seguem um formato diferente do que usamos costumeiramente. Um nome de usuário válido nesse sistema deve obedecer aos seguintes critérios:

* Deve começar ou com underscore '\_' (código ascii 95) ou com um ponto '.' (código ascii 46)
* Deve ser seguido imediatamente por uma ou mais ocorrências de dígitos (qualquer número entre 0 e 9, inclusive)
* Os números podem ser seguido de letras (maiúsculas ou minúsculas). Zero ou mais ocorrências.
* O nome de usuário pode terminar com um underscore opcional

Sua tarefa é validar se um dado nome de usuário é válido ou não

**Formato de Entrada**

A primeira linha contém um número N. As próximas N linhas contêm um nome de usuário cada.

Restrição

1 <= N <= 100

**Formato de Saída**

Para cada nome de usuário, imprima

VALIDO

se o nome do usuário atender aos critérios acima

INVALIDO

caso contrário.

**Exemplo de Entrada**

3

\_0898989811abced\_

\_abce

\_09090909abcD0

**Exemplo de Saída**

VALIDO

INVALIDO

INVALIDO

**Explicação**

O primeiro nome de usuário é válido, visto que ele atende aos 3 primeiros critérios mencionados acima

O segundo nome de usuário é inválido, visto que '\_' não é imediatamente seguido por um número

O terceiro nome de usuário é também inválido, visto que as letras são seguidas de números. O que não é permitido pelos critérios de validade.

**Dicas**

- Forneça a solução mais elegante que você encontrar

- Use Python para resolver o problema

- Considere que as entradas são digitadas na entrada padrão, não em arquivos texto

**Cardápio Online**

Um restaurante necessita do desenvolvimento de uma aplicação que disponibilize seu cardápio online e nos aplicativos móveis de terceiros a partir de uma API Rest.

Os principais requisitos da aplicação são:

* A aplicação deverá possuir uma versão Web e uma API Rest para acesso aos dados.
* Os clientes **não** precisarão se autenticar para acessar as informações do cardápio.
* As mesmas funcionalidades disponíveis na versão Web deverão estar disponíveis na API Rest.
* O cardápio deverá ser separado em categorias (entradas, principal, sobremesa, bebidas…).
* Cada item do cardápio deverá possuir as seguintes informações: nome; ingredientes; valor; tempo de preparo.
* Os clientes poderão filtrar por uma categoria.
* Os clientes poderão pesquisar pelo nome do item do cardápio.
* Os clientes poderão ordenar pelo valor do item do cardápio.

**Dicas**

- Utilize o framework Django para resolver esse problema

- Utilize o Django REST framework

- Para interface web, utilize somente o bootstratp

- Você pode utilizar o SQLite

Na página a seguir é apresentado um exemplo de retorno da API Rest.

**Exemplo de formato de saída da API Rest:**

{

"count": 4,

"next": null,

"previous": null,

"results": [

{

"id": 1683,

"nome": "Salmão",

"valor": 29,

"categoria": {

"id": 14,

"name": "Pratos Principais"

},

"ingredientes": [

{

"id": 14,

"name": "Salmão"

},

{

"id": 15,

"name": "Azeite"

},

{

"id": 16,

"name": "Pimenta-do-reino Branca"

}

]

}

]

}