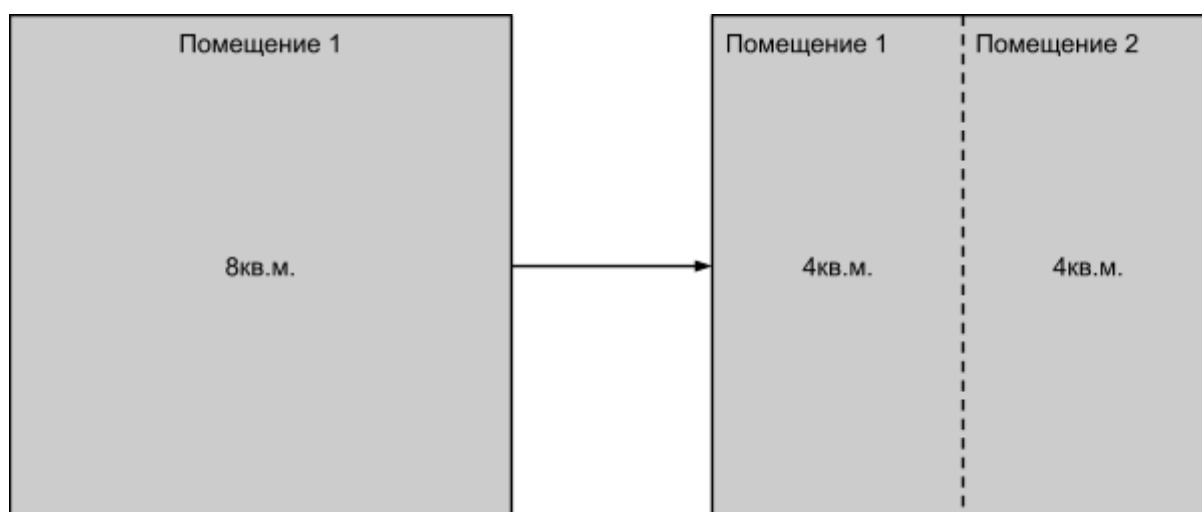


## Указатели, динамична памет

**Задача 1.** Въведете две числа N и M, заделете динамично памет за матрица с N реда и M стълба и прочетете елементите ѝ. Подредете матрицата, така че първи да бъде редът с най-малка сума, след това този със следваща по големина сума и т.н. Изведете получената матрица на екрана.

**Задача 2.** Император Палпатин има сериозен проблем. След две седмици е битката при Хот, а в спалните помещения на Звездните Разрушители няма място заради новите попълнения от Татуин. След съвещание с Вейдър, се разбрали че комфорта на щурмоваците му не е от особено значение и се сетили, че след последната победа на борда на корабите има *“безкрайно”* количество завеси. И измислили решение - да разделят помещенията на две като провисват завеси в средата на свободното пространство и така да настанят новодошлите.

Например, ако има едно спално помещение с големина 8 кв.м., могат да опънат завеса през средата и да направят 2 помещения с големина 4 кв.м. всяко.



Разбира се, след това могат отново да спуснат завеса през средата на едното от по-малките спални помещения и да получат 3 спални с размери: [4 кв.м, 2 кв.м, 2 кв.м].

Вашата задача е да улесните императора с разделението на стаите.

В началото на програмата, ще получите броя спални помещения на разрушителя и големината на всяко помещение като моделите разрушители варират от такива с 4 до такива с 1024 спални помещения, а големината на всяко помещение е число, степен на двойката между 2 и 256.

### Пример:

```
>> 4
>> 64 8 16 4
```

Приемаме, че всички спални помещения са подредени в редица едно до друго.

Тъй като Империята не знае колко войници ще се разположат на корабите, а и за удобството на Вейдър, вашата програма ще трябва да се справя със серия от команди.

#### **A <number>**

Команда за добавяне на <number> щурмоваци в кораба.

Всеки нов войник трябва да има самостоятелно помещение, което да се създаде, чрез завеси като разделянето е следното:

Най-голямото помещение се разделя на две и в новото свободно пространство се добавя един войник. Ако има две най-големи помещения избираме първото.

Подредбата е от значение, за това най-голямото помещение остава на мястото си, намаля големината си и след него се добавя новото.

#### **Пример:**

```
>> 4
>> 8 64 16 2
>> A 1
// [8, 64, 16, 2] -> [8, 32, 32, 16, 2]
```

Най-малкото възможно помещение е с размер 2 кв.м. Ако на борда на кораба няма помещения по-големи от 2 кв.м на командата трябва да се отговори с "Try again after Hoth."

Ако при команда за добавяне стигнем до момент при който в кораба няма повече места, принтираме "Try again after Hoth.", но все пак разполагаме на кораба всички войници, които можем.

#### **Пример:**

```
>> 4
>> 2 2 2 2
>> A 1
<< Try again after Hoth
```

#### **P**

Команда за принтиране на вида на стаите във формата

[<големина стая 1>, <големина стая 2>, ..., <големина стая n>]

#### **Пример:**

```
>> 4
>> 2 4 8 16
>> P
<< [2, 4, 8, 16]
```

#### **E**

Команда за приключване на програмата.

След нея всички ресурси трябва да бъдат освободени.

Тъй като ресурсите на терминалите на разрушителите са лимитирани от вашата програма се изисква да използва най-много **2 \* броя стаи** памет за стаите при изпълнение.

**Примерно изпълнение:**

```
>> 4
>> 2 8 2 8
>> A 2
// [2, 4, 4, 2, 8]
// [2, 4, 4, 2, 4, 4]
>> P
<< [2, 4, 4, 2, 4, 4]
>> A 5
// [2, 2, 2, 4, 2, 4, 4]
// [2, 2, 2, 2, 2, 2, 4, 4]
// [2, 2, 2, 2, 2, 2, 2, 2, 4]
// [2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
// -
<< Try again after Hoth.
>> P
<< [2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

*Навсякъде в примерите с >> е маркирано началото на данни, въведени от потребителя, с << началото на данни изведени от програмата, а с // коментари, които не са част от изпълнението, но са сложени за по-добро разбиране на примера.*