

[ED201] Playlist

Neste problema deverá submeter uma classe **ED201** contendo um programa completo para resolver o problema (ou seja, com o método main).

O Aniceto vai viajar de comboio e pretende ouvir música durante a sua viagem. Como é muito organizado, deseja ter uma *playlist* de músicas já escolhida. Sabendo a duração da viagem, ele deseja escolher **um conjunto de músicas que não ultrapasse a duração (ele não gosta de ficar com uma música a meio) mas que ao mesmo tempo tenha uma duração total o mais próxima possível da duração (para não ficar muito tempo sem ouvir nada)**. Além disso, ele não quer que a mesma música apareça mais do que uma vez na *playlist*.

Imagina por exemplo que ele tem as seguintes 9 músicas disponíveis:

| # | Intérprete | Música | Duração (segundos) |
|----|-----------------------|-------------------------------|--------------------|
| 01 | Yann Tiersen | Porz Goret | 243 |
| 02 | Ludovico Einaudi | Fly | 202 |
| 03 | God is an Astronaut | All Is Violent, All Is Bright | 254 |
| 04 | Explosions In The Sky | Your Hand In Mine | 502 |
| 05 | Opeth | Harvest | 385 |
| 06 | Green Carnation | 9-25-045 | 942 |
| 07 | Porcupine Tree | Lazarus | 237 |
| 08 | Metallica | Fade to Black | 721 |
| 09 | Tuna Javardemica | Ciencito Aluno | 192 |

Considere que o Aniceto sabe que **a viagem demora 1300 segundos**. Se experimentar ir adicionando as músicas enquanto não ultrapassarem a duração, ele chega à seguinte *playlist*:

- Hipótese 1:** #01, #02, #03, #04 | Duração: 243+202+254+502 = **1201 segundos**

O Aniceto resolveu começar a adiconar as músicas começando pela mais pequena, e com isso chegou à seguinte *playlist*, que é pior que a anterior:

- Hipótese 2:** #09, #02, #07, #01, #3 | Duração: 192+202+237+243+254 = **1128 segundos**

O Aniceto tentou várias outras estratégias (como começar pela maior música) mas percebeu que nenhuma dava sempre a melhor solução possível. Em particular, para o seu caso, a melhor *playlist* era a seguinte, com apenas menos 2 segundos que a duração da viagem:

- Hipótese 3:** #05, #08, #09 | Duração: 385+721+192 = **1298 segundos**

Será que podes ajudar o Aniceto a encontrar qual a melhor *playlist* para futuras viagens?

O Problema

Dada a duração de uma viagem e um conjunto de músicas (especificadas pela sua duração), tens de calcular qual a duração da *playlist* ótima, ou seja qual o subconjunto de músicas tal que a soma das suas durações seja a maior possível e ao mesmo tempo essa soma seja menor ou igual à duração da viagem.

Input

Na primeira linha do input vem dois números inteiros **D** ($1 \leq D \leq 10\,000$) e **N** ($1 \leq N \leq 20$) indicando a duração da viagem e o número de músicas disponíveis.

Seguem-se exactamente **N** linhas, cada uma com um número **M_i** indicando a duração da *i*-ésima música (as durações das músicas são inteiros entre 1 e 10 000). É garantido que pelo menos uma música tem duração igual ou inferior a **D**.

Output

O output é constituído por uma linha contendo um único número indicando a duração total da melhor *playlist* possível, ou seja, a escolha de um subconjunto de músicas que maximize $\sum M_i$ tal que $\sum M_i \leq D$

Exemplo de Input

```
1300 9
243
202
254
502
385
942
237
721
192
```

(Nota que o exemplo de input corresponde ao exemplo do enunciado)

Exemplo de Output

```
1298
```

