Lavanya's Party Cap

HackerRank

`Time Limit: 2 seconds`

`Memory Limit: 512MB`

The tea on Lavanya *di* is legendary. She isn't just "mad for parties"—she's a party *vortex*. Her social calendar is a blur, and she's left a trail of IOUs and unpaid bills across the college. She has successfully partied with seniors, juniors, professors, and (allegedly) a homeless guy she met at 3 AM.

Now, the bill is due. She's staring at a massive, terrifying list of N party expenses (decor, damages, "emotional support" snacks...).

But Lavanya doesn't *pay* bills. She *delegates* them.

She has successfully cornered D "sponsors"—unwitting victims she charmed or blackmailed into helping. She now needs to "distribute the opportunity" (i.e., dump her debt) onto them.

Being methodically lazy, she won't reorder her expense list. It's too much work. She'll just go down the line:

- The first sponsor (Victim 1) gets a contiguous block of expenses from the start (e.g., expenses 1-3).
- The second sponsor (Victim 2) gets the *next* contiguous block (e.g., expenses 4-6).
- ...and so on, until all N expenses are covered by her D sponsors.

She has one critical problem: if any single sponsor's bill is too high, they might actually realize they've been scammed and start a revolution. She needs to find the "sweet spot" to keep everyone just *calm* enough.

She needs you to find the **minimum possible value for the *maximum* bill** any single sponsor gets stuck with. If she can make this "maximum bill" (let's call it C) as small as possible, no single person will feel *too* betrayed, and she can live to party another day.

Input Format

- The first line contains a single integer T — the number of test cases.
- Each test case consists of two lines:
  - The first line contains two integers N (the number of expenses) and D (the number of sponsors).
  - The second line contains N space-separated integers, $e_1, e_2, \ldots, e_N$, representing the cost of each party expense.

Constraints

- $1 \leq T \leq 10$
- $1 \leq D \leq N \leq 10^5$

- $1 \leq \text{expense}_i \leq 10^6$
- The sum of N over all test cases will not exceed 10^6.
- **Note:** Her party bills are... astronomical. The total bill for a sponsor might overflow a 32-bit integer.

Output Format

- For each test case, output a single integer: the **minimum possible value for the largest bill** any sponsor has to pay.

Sample Input 0

```
2
5 2
7 2 5 10 8
6 3
1 2 3 4 5 6
```

Sample Output 0

```
18
9
```

Explanation 0

Test Case 1

- **Input:** $N = 5$ expenses, $D = 2$ sponsors.
- **Expenses:** `[7, 2, 5, 10, 8]`

We need to split the list into **2 contiguous subarrays** such that the largest sum is minimized. The optimal split is:

1. **Sponsor 1:** `[7, 2, 5]` $\rightarrow$ Sum = $7 + 2 + 5 = 14$
2. **Sponsor 2:** `[10, 8]` $\rightarrow$ Sum = $10 + 8 = 18$

The bills are 14 and 18. The maximum bill is 18. *(Note: If we tried splitting it as* `[7, 2]` *and* `[5, 10, 8]`*, the bills would be* 9 *and* 23*. Since* $23 > 18$*, that arrangement is worse.)*

Test Case 2

- **Input:** $N = 6$ expenses, $D = 3$ sponsors.
- **Expenses:** `[1, 2, 3, 4, 5, 6]`

We need to split the list into **3 contiguous subarrays**. The optimal split is:

1. **Sponsor 1:** `[1, 2, 3]` $\rightarrow$ Sum = 6
2. **Sponsor 2:** `[4, 5]` $\rightarrow$ Sum = 9
3. **Sponsor 3:** `[6]` $\rightarrow$ Sum = 6

The bills are 6, 9, and 6. The maximum bill is 9.