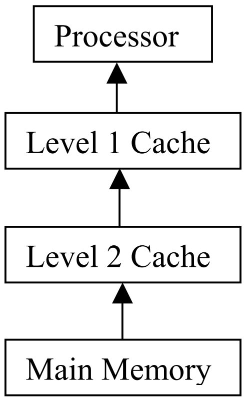
Problem 2: Cache Simulation

Most modern computers use one or more levels of cache memory between the processor and the main memory to minimize the time the processor has to wait for information from main memory. Each cache level is characterized as having some number of memory blocks, each of which has a fixed size (measured in bytes, and always a power of 2); the total size of a cache level is just the number of blocks in that cache level times the size of a block. The address of the lowest-numbered byte in each block is always a integral multiple of the block size, and the bytes in a block have contiguous addresses. For example, with a block size of 16, the bytes in a block might possibly be numbered 16 though 31, or 32 through 47, or 160 through 175.



The processor in this problem only reads single bytes, and it does so by issuing a request that specifies the address of the desired byte. If the byte is in the cache closest to the processor (known as the "level 1 cache"), then that cache delivers the byte to the processor; the length of time required for this operation is called the "level 1 access time." If the byte desired by the processor is not in the level 1 cache, but is in the level 2 cache (which is just below the level 1 cache), then the (level 1 size) block containing that byte is delivered from the level 2 cache to the level 1 cache, and then the desired byte is delivered from the level 1 cache to the processor. The total time required in this case is the time required by the level 2 cache to deliver the block to the level 1 cache (naturally called the "level 2 access time"), plus the level 1 access time for the single byte. This pattern continues through all lower cache levels (if present) to the main memory, if necessary. Thus, if a byte requested by the processor isn't in any of the cache levels, the total access time required is the sum of the access times of each cache level plus that of the main memory. The figure to the left illustrates the flow of information in a system with two cache levels.

Each cache is initially empty. When a block is retrieved from a lower-level cache or main memory, it is placed in an empty block in the cache. When no empty blocks are available, and a new block is requested, it will replace an existing block. The particular block it replaces is that block that has been least recently used.

In this problem you will be given the number of caches in a system (between 1 and 3), the block size and total size of each cache, and the access time for each cache and the main memory. Times will be in integral numbers of nanoseconds (nsec). You will then be given a list of the addresses of bytes requested by the processor, and are to compute the time the processor must wait for all of the bytes to be delivered.

As a simple (but unreaslistic) example, suppose the system has two caches. The level 1 cache has 16 byte blocks, a total size of 32 bytes (that is, 2 blocks), and an access time of 4 nanoseconds. The level 2 cache has 32 byte blocks, a total size of 64 bytes (2 blocks), and an access time of 10 nanoseconds. Main memory has an access time of 50 nanoseconds. Suppose the processor requests, in order, bytes from locations 10, 20, 30, 40, and 50. (Cache blocks are numbered here for reference.)

- Since both levels are initially empty, 32 bytes from main memory locations 0 through 31 will be placed in level 2 block 0 (50 nsec), then 16 bytes from that block (addresses 0 through 15) will be placed in level 1 block 0 (10 nsec). Finally, the byte with address 10 will be delivered to the processor (4 nsec). Total time to access the first byte is 64 nsec.
- The byte with address 20 isn't found in level 1, but is found in level 2 in block 0. The 16 bytes containing address 20 (16 to 31) are placed in level 1 block 1 (10 nsec), and the byte with address 20 is delivered to the processor (4 nsec), for a total time of 14 nsec. Note that both blocks in the level 1 cache are now in used.

- Next, the byte with address 30 is sought. Since it is found in the level 1 cache, that byte is simply delivered to the processor (4 nsec).
- Now address 40 is issued by the processor. The byte at this location is not in either cache level, so the corresponding 32 byte block (addresses 32 to 63) is delivered to the level 2 cache and placed there in block 1 (which was previously unused), taking 50 nsec. Then the 16 bytes containing address 40 (32 to 47) are delivered to the level 1 cache (10 nsec more). These 16 bytes are placed in block 0 of the level 1 cache, since it is the least recently used (block 1 of the level 1 cache was used to satisfy the processor's request for address 30). Finally, the byte is delivered to the processor, for a total time of 64 nsec.
- Finally address 50 is requested. Found in the level 2 cache in block 1, the appropriate 16 bytes (48 to 63) are delivered to the level 1 cache (10 nsec). These bytes are placed in block 1 of the level 1 cache, since block 0 was just used. The selected byte is delivered to the processor (4 nsec), for a total time requirement of 14 nsec.

The total time for the bytes at this sequence of addresses to be delivered to the processor is 64 + 14 + 4 + 64 + 14 = 160 nsec.

Input
There will be multiple input cases. For each case, the input begins with an integer that specifies the number of cache levels (between 1 and 3). For each cache level, starting with level 1, the input then contains integers giving the block size, total size, and access time for the cache level. Each cache level has mo more than 100 blocks, and a block size that is no larger than the next (lower level) cache. Next there appears an integer giving the access time for the main memory. Finally, there appears an integer specifying the number of addresses requested by the processor (no more than 1,000) followed by those addresses in the order they were requested; each address is in the range 0 to 65535. A single 0 follows the input for the last case.

Output
For each case, display a single line containing the case number (1, 2, …) and the total time required for all of the bytes requested by the processor to be delivered.

Sample Input

```
2
  16 32 4
  32 64 10
  50
  5 10 20 30 40 50

2
  8 48 4
  32 64 10
  50
  5 10 20 30 40 50

0
```

Expected Output

```
Case 1: total time = 160 nanoseconds
Case 2: total time = 170 nanoseconds
```