Encoded String

Problem Code: **DECODEIT**

Submit

Read problem statements in Hindi and Mandarin Chinese.

An encoder encodes the **first 16 lowercase English letters** using 4 bits each. The first bit (from the left) of the code is 0 if the letter lies among the first 8 letters, else it is 1, signifying that it lies among the last 8 letters. The second bit of the code is 0 if the letter lies among the first 4 letters of those 8 letters found in the previous step, else it's 1, signifying that it lies among the last 4 letters of those 8 letters. Similarly, the third and the fourth bit each **signify the half in which the letter lies**.

For example, the letter j would be encoded as :

- Among $(a, b, c, d, e, f, g, h \mid i, j, k, l, m, n, o, p)$, j appears in the second half. So the first bit of its encoding is 1.
- Now, among $(i, j, k, l \mid m, n, o, p)$, j appears in the first half. So the second bit of its encoding is 0.
- Now, among $(i, j \mid k, l)$, j appears in the first half. So the third bit of its encoding is 0.
- Now, among $(i \mid j)$, j appears in the second half. So the fourth and last bit of its encoding is 1.

So j's encoding is 1001,

Given a binary encoded string S, of length at most 10^5, decode the string. That is, the first 4 bits are the encoding of the first letter of the secret message, the next 4 bits encode the second letter, and so on. It is guaranteed that the string's length is a multiple of 4.

Input:

- The first line of the input contains an integer T, denoting the number of test cases.
- The first line of each test case contains an integer N, the length of the encoded string.
- The second line of each test case contains the encoded string S.

Output:

For each test case, print the decoded string, in a separate line.

Constraints

- $1 \leq T \leq 10$
- $4 \leq N \leq 10^5$
- The length of the encoded string is a multiple of 4.
- $0 \leq S_i \leq 1$

Subtasks

- 100 **points :** Original constraints.

Sample Input:

```
3
4
0000
8
00001111
4
1001
```

Sample Output:

```
a
ap
j
```

Explanation:

- **Sample Case 1 :**

The first bit is 0, so the letter lies among the first 8 letters, i.e., among a, b, c, d, e, f, g, h. The second bit is 0, so it lies among the first four of these, i.e., among a, b, c, d.

The third bit is 0, so it again lies in the first half, i.e., it's either a or b. Finally, the fourth bit is also 0, so we know that the letter is a.

- **Sample Case 2 :**

Each four bits correspond to a character. Just like in sample case 1, 0000 is equivalent to a. Similarly, 1111 is equivalent to p. So, the decoded string is ap.

- **Sample Case 3 :**

The first bit is 1, so the letter lies among the last 8 letters, i.e., among i, j, k, l, m, n, o, p. The second bit is 0, so it lies among the first four of these, i.e., among i, j, k, l.

The third bit is 0, so it again lies in the first half, i.e., it's either i or j. Finally, the fourth bit is 1, so we know that the letter is j.

Author:	daanish_adm
Date Added:	31-12-2020
Time Limit:	1 secs
Source Limit:	50000 Bytes
Languages:	CPP14, C, JAVA, PYTH 3.6, PYTH, CS2, ADA, PYPY, PYP3, TEXT, CPP17, PAS fpc, RUBY, PHP, NODEJS, GO, TCL, HASK, PERL, SCALA, kotlin, BASH, JS, PAS gpc, BF, LISP sbcl, CLOJ, LUA, D, R, CAML, rust, ASM, FORT, FS, LISP clisp, SQL, swift, SCM guile, PERL6, CLPS, WSPC, ERL, ICK, NICE, PRLG, ICON, PIKE, COB, SCM chicken, SCM qobi, ST, NEM, SQLQ

Submit

Comments