**5. Base64 Decoder**

**Description**

**Base64** is an encoding technique that represents binary data in a plain-text(human readable, displayable) string format by translating it into a 64-radix representation. It works by dividing every three bytes of data into six-bit units.

Base64 can be used for transmitting binary data (e.g. image, executable), when the transmitting channel does not allow binary data (such as email). But in this task, we will Base64 encode a plain-text, for simplicity. In other words, original string and base64 encoded string are both plain-text.

For example, let’s Base64 encode the word ‘**Man**’.

1. Represent the original string in binary bit pattern.

|  |  |  |
| --- | --- | --- |
| **Original character** | **ASCII (decimal)** | **Bit pattern** |
| M | 77 | 01001101 |
| a | 97 | 01100001 |
| n | 110 | 01101110 |

‘Man’ = 010011010110000101101110.

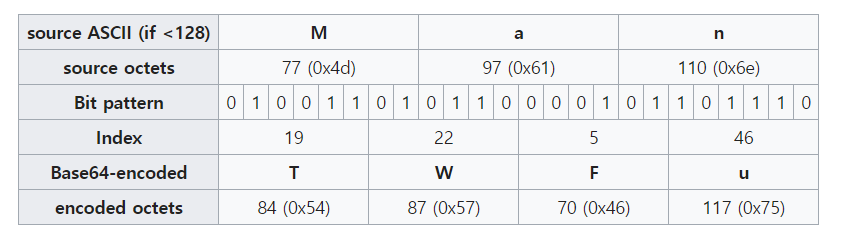
1. Divide binary into six-bit units

010011010110000101101110 = 010011 010110 000101 101110

1. Convert each units into a character using Base64 table ( see Appendix )

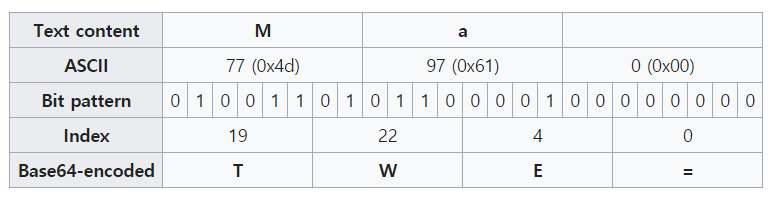
|  |  |  |
| --- | --- | --- |
| **Six-bit unit** | **Decimal** | **Base64 encoded** |
| 010011 | 19 | T |
| 010110 | 22 | W |
| 000101 | 5 | F |
| 101110 | 46 | u |

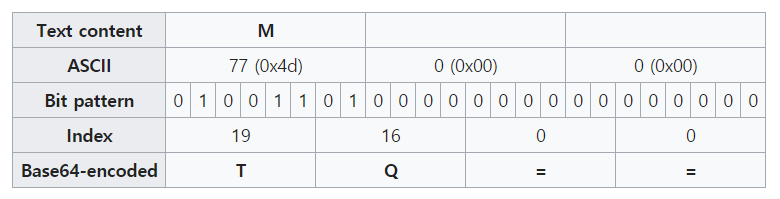
So, Base64 encoded value of the word ‘**Man**’ is ‘**TWFu**’.



However, if a length (number of characters) of the original string is not a multiple of 3, it cannot be divided into sit-bit units (8\*3 = 6\*4). In that case, padding bytes 0(00000000) are appended until the length of the string becomes a multiple of 3. When Base64 encoded, the padding unit is encoded as a character ‘**=**’. Therefore, a length of the Base64 encoded string is always a multiple of 4.

For example, ‘**Ma**’ is encoded into ‘**TWE=**’ and ‘**M**’ is encoded into ‘**TQ==**’.





You can decode the Base64 encoded string, by applying these steps in reverse order.

In this task, you need to implement a **Base64** decoder (NOT an encoder!).

**Input**

On the first line, the number of Base64 encoded string N is given.

On the next N lines, Base64 encoded string is given.

**Output**

For each line, print Base64 decoded value of each input string. Output string is always a plain text.

**Constraints**

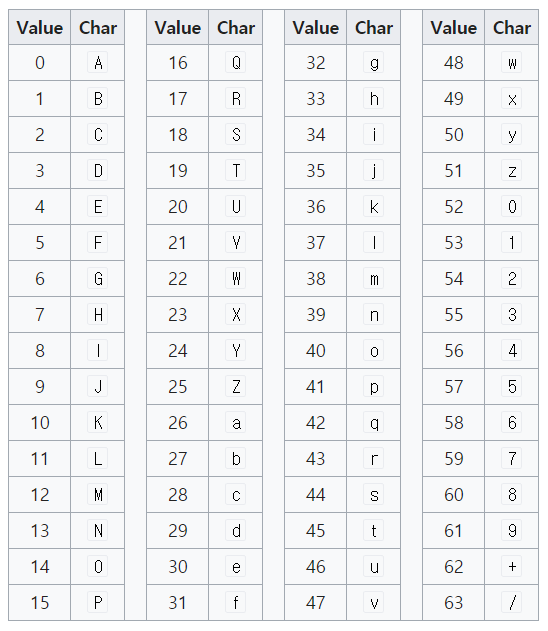
1 <= N <= 1000

Length of each word is smaller than or equal to 1000.

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  TWFu  TWE=  TQ== | Man  Ma  M |
| 3  QXJzZW5hbA==  V2VuZ2Vy  RW1pcmF0ZXM= | Arsenal  Wenger  Emirates |
| 5  My4xNDE1OTI=  Mi43MTgyODE4  U3VwM3JfUzNjUmU3fk1lc1NAZ2Uh  YWJjZGVmZ2hpamtsbW5vcHFyc3R1dnd4eXo=  IUAjJCVeJiooKV8rfg== | 3.141592  2.7182818  Sup3r\_S3cRe7~MesS@ge!  abcdefghijklmnopqrstuvwxyz  !@#$%^&\*()\_+~ |

**Appendix – Base64 Table**

****

|  |  |
| --- | --- |
| **Value** | **Char** |
| 0 | = ( only for padding ) |