題目 560

✔560. 三十六進制

Submit | Status | Discussion / Solution | Ranklist | Back to Problems List

TopCoder	User's AC Ratio	Submission's AC Ratio	Tags
 abcabcabc *有人要寫p6 嗎> <*	90.9% (10/11)	44.2% (19/43)	Show solution-related tags: apcsc, apcsc-string

Description

三十六進制，是一種利用 `0` 到 `9` 與 `A` 到 `Z` 組成的進位方式，其中 `A` 代表 10、`B` 代表 11，其他依此類推。相較於十進位與十六進位，三十六進制能夠以更短的字串來表示數字，進而省下更多的儲存空間。

現在就來實作一個程式，能夠用它把三十六進制與十進制互相轉換。

Input Format

輸入有 $N+1$ 行。第一行有一個數字 N，代表接下來有 N 個指令，每個指令為一行。

每個指令有以下兩種：

1. `Base10 n` 代表一個十進制的整數 n，需要把它轉為三十六進制，輸出後換行。
2. `Base36 s` 代表一個三十六進制的整數 s，需要把它轉為十進制的 m，輸出後換行。

- $1 \leq N \leq 10^5$
- $0 \leq m, n \leq 2^{64}$
- 三十六進位中的英文字母皆為大寫。

Output Format

輸出共有 N 行，對於每一個指令，請輸出對應的轉換結果。

Sample Input 1

```
4
Base10 12345
Base10 123456789
Base36 123456
Base36 ABCDEF
```

Sample Output 1

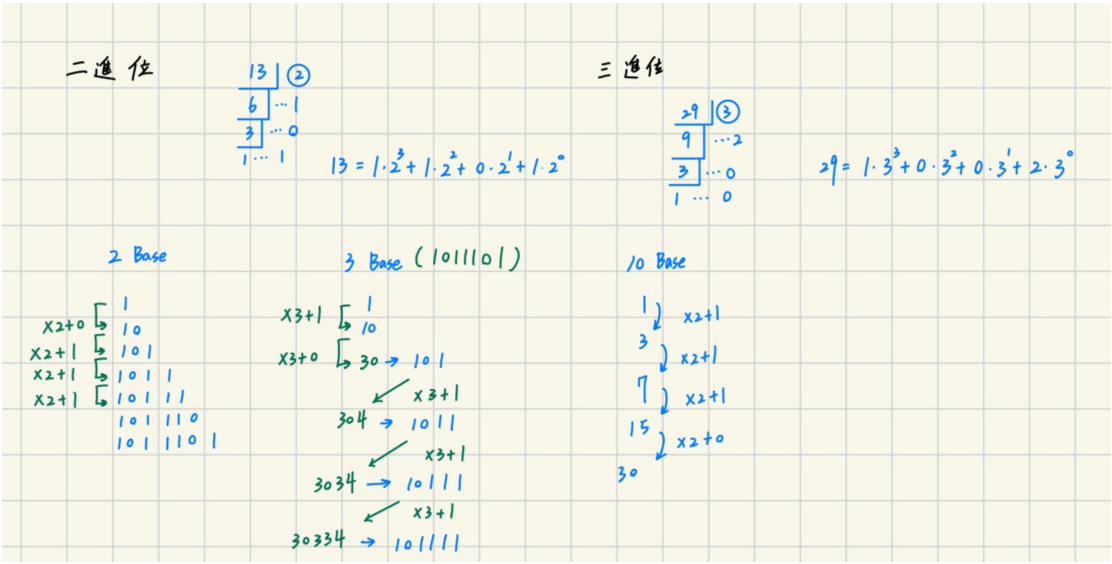
```
9IX
21I3V9
63970746
623714775
```

Sample Input 2

```
6
Base10 5334221234555
Base10 11556654433221
Base36 APCSCAMP
Base36 NTUCSIE
Base36 TAIWAN
Base36 COVID19
```

Sample Output 2

```
1W2I9LIEZ
43H1XNYKL
838834396369
51870498134
1771196927
27625501053
```



Problem	Total Time (ms)	Max Memory (RSS, KiB)	Verdict	Score
560. 三十六進制	469	27648	Accepted	100

Subtask Results

Subtask no.	Testdata Range	Constraints	Score
1	0–1	範例測資	0 / 0
2	0–12	無額外限制	100 / 100

Testdata Results

Testdata no.	Subtasks	Time (ms)	Memory (VSS, KiB)	Memory (RSS, KiB)	Verdict	Score
0	1 2	3.1	6356	3456	Accepted	100
1	1 2	4.0	6356	3456	Accepted	100
2	2	3.8	6356	3456	Accepted	100
3	2	35.5	8752	5760	Accepted	100
4	2	204.9	30500	27648	Accepted	100
5	2	106.7	18364	14976	Accepted	100
6	2	5.6	6488	3456	Accepted	100
7	2	8.6	6620	3456	Accepted	100
8	2	24.1	7544	4608	Accepted	100
9	2	28.9	7980	4608	Accepted	100
10	2	32.3	8492	5760	Accepted	100
11	2	7.2	6488	3456	Accepted	100
12	2	3.8	6356	3456	Accepted	100

Submitter: joseph **Compiler:** c++17 **Code Length:** 2.3 KB

```cpp
#include <iostream>
#include <cmath>
using namespace std;
int main(){
    int n;
    cin >> n;
    cin.ignore();
    int base[n];
    int num[n];
    for(int i = 0; i < n; i++) num[i] = 0;
    char **input = new char*[n];
    for (int i = 0; i < n; i++)
    {
        input[i] = new char[100];
        cin.getline(input[i], 100);
        for (int j = 7;input[i][j] != '\0' ; j++)
        {
            num[i]++;
        }
        if (input[i][4] == '1')
        {
            base[i] = 1;
        }else{
            base[i] = 3;
        }
    }
    char **out = new char *[n];
    for (int i = 0; i < n; i++)
    {
        out[i] = new char[100];
        if (base[i] == 1)
        {
            unsigned long long sum = 0;
            for (int j = 7; input[i][j] != '\0'; j++)
            {
                int a = input[i][j] - '0';
                sum = sum * 10 + a;
            }
            int idx = num[i] - 1;
            while (sum / 36 != 0)
            {
                if (sum % 36 < 10)
                {
                    out[i][idx] = (sum % 36) + '0';
                }else{
                    out[i][idx] = (sum % 36) + 'A' - 10;
                }
                sum /= 36;
                idx--;
            }
```

```
            if (sum % 36 < 10){
                out[i][idx] = (sum % 36) + '0';
            }else{
                out[i][idx] = (sum % 36) + 'A' - 10;
            }
            for (int k = idx; k < num[i]; k++)
            {
                cout << out[i][k];
            }
            cout << endl;
        }else{
                unsigned long long sum = 0;
            for (int j = 7; input[i][j] != '\0'; j++)
            {
                char c = input[i][j];
                int a;
                if (c >= '0' && c <= '9') {
                    a = c - '0';
                }
                else if (c >= 'A' && c <= 'Z'){
                     a = c - 'A' + 10;
                }
                sum = sum * 36 + a;
            }
            int idx = 100 - 1;
            while (sum / 10 != 0)
            {
                out[i][idx] = sum % 10 + '0';
                sum /= 10;
                idx--;
            }
            out[i][idx] = sum % 10 + '0';
            for (int k = idx; k < 100; k++) cout << out[i][k];
            cout << endl;
            }
        }

  return 0;
}
```

這題的解法類似於 CLASS EXAM 01 裡的第六題，要先了解進位的原理

cin.ignore();　　//避免 ’\n’ 被吃掉

int base[n];

int num[n];

for(int i = 0; i < n; i++) num[i] = 0;　　//配置每行的長度

char **input = new char*[n];

for (int i = 0; i < n; i++){

```
        input[i] = new char[100];
        cin.getline(input[i], 100);       //因為中間有空白，所以用 getline
        for (int j = 7;input[i][j] != '\0' ; j++){
            num[i]++;
        }
        if (input[i][4] == '1'){
            base[i] = 1;                  //判斷每行是 36 進位還是 10 進位
        }else{
            base[i] = 3;
        }
    }
    char **out = new char *[n];
    for (int i = 0; i < n; i++){
        out[i] = new char[100];
        if (base[i] == 1){
            unsigned long long sum = 0;
            for (int j = 7; input[i][j] != '\0'; j++){
                int a = input[i][j] - '0';
                sum = sum * 10 + a;
            }
            int idx = num[i] - 1;
            while (sum / 36 != 0){
```

```
            if (sum % 36 < 10){
                out[i][idx] = (sum % 36) + '0';
            }else{
                out[i][idx] = (sum % 36) + 'A' - 10;}
        sum /= 36;      //while 的終止條件
        idx--;
        }
        if (sum % 36 < 10){
                out[i][idx] = (sum % 36) + '0';
        }else{
                out[i][idx] = (sum % 36) + 'A' - 10;}        //補上最後一位
        for (int k = idx; k < num[i]; k++){
            cout << out[i][k];
        }
        cout << endl;
    }else{
        unsigned long long sum = 0;
        for (int j = 7; input[i][j] != '\0'; j++){
            char c = input[i][j];
            int a;
                if (c >= '0' && c <= '9') {   //留意 36 進位在大於 10 是英文字母所以要另外處理
```

```
                    a = c - '0';}
                else if (c >= 'A' && c <= 'Z'){
                    a = c - 'A' + 10;
                }
                sum = sum * 36 + a;
            }
            int idx = 100 - 1;
            while (sum / 10 != 0){
                out[i][idx] = sum % 10 + '0';
                sum /= 10;
                idx--;
            }
            out[i][idx] = sum % 10 + '0';    //補位
            for (int k = idx; k < 100; k++) cout << out[i][k];
            cout << endl;
        }
    }
    return 0;
}
```