

Trong ngôn ngữ Pascal có các hàm **eof**, **seekeof**, **eoln**, **seekeoln** để kiểm tra việc đọc đến hết file và đọc đến hết dòng. Tuy nhiên C++ không trang bị các hàm này. Do vậy chúng ta phải sử dụng một số kỹ thuật (chủ yếu là vận dụng khéo léo các hàm đọc của C++)

a. Cho một file chứa các số nguyên cách nhau bởi dấu trống, ký tự xuống dòng hoặc dòng trống. Hãy tính tổng các số nguyên này.

```
int t=0;
while (scanf("%d",&x)==1) t+=x;
cout <<t
```

b. Cho một file chứa các số nguyên cách nhau bởi dấu trống, ký tự xuống dòng hoặc dòng trống. Với mỗi dòng có ít nhất một số nguyên, hãy in ra tổng các số của từng dòng.

```
#include <bits/stdc++.h>
#define maxs 1000001
#define maxn 1001
using namespace std;
char s[maxs];
int n, a[maxn];

int main()
{
    freopen("inp.txt", "r", stdin);
    n=0;
    while (gets(s)!=NULL) if (s[0]) {
        char* pch=strtok(s, " ");
        if (pch!=NULL) {
            int t=0;
            while (pch!=NULL) {
                int x; sscanf(pch, "%d", &x);
                t+=x;
                pch=strtok(NULL, " ");
            }
            a[++n]=t;
        }
    }
    for(int i=1;i<=n;i++) printf("%d\n",a[i]);
}
```

Ví dụ: Bài TAPCON

```
char s[100000], ss[100000];
scanf("%d\n",&n);
// Tính số lượng tập con
while (gets(s)!=NULL) if (s[0]) { // chừng nào còn đọc được xâu s khác rỗng
    char* pch=strtok(s," "); // Các số được cách nhau bởi dấu trống
    int loai; sscanf(pch,"%d",&loai);
    if (loai==1) {
        p=0;
        while (pch!=NULL) {
            sscanf(pch,"%d", &x[++p]);
            pch=strtok(NULL," ");
        }
        // Tính thứ tự của tập hợp x[1], x[2], ..., x[p]
    } else {
        sscanf(pch,"%s",ss);
        // Đổi xâu ss thành số nguyên lớn m
        // Tìm hoán vị tương ứng
    }
}
```