Team Notebook

ntTas

April 14, 2019

Contents

1 Setup **2**

1.1 C++Template 2

1.2 FastScanner 2

1 Setup

1.1 C++Template

```
#pragma GCC optimize ("O3")
#pragma GCC target ("sse4")

#include <bits/stdc++.h>
using namespace std;

#define fi first
#define se second
#define pb push_back

typedef long long LL;
typedef vector<int> vi;
typedef pair<int,int> ii;

const int MOD = 1e9 + 7;
const LL INF = 1e18;

void fastscan(int &number) {
    //variable to indicate sign of input number
    bool negative = false;
    register int c;

    number = 0;

    // extract current character from buffer
    c = getchar();
    if (c=='-')
    {
        // number is negative
        negative = true;

        // extract the next character from the buffer
        c = getchar();
    }

    // Keep on extracting characters if they are integers
    // i.e ASCII Value lies from '0'(48) to '9' (57)
    for (; (c>47 && c<58); c=getchar())
        number = number *10 + c - 48;

    // if scanned input has a negative sign, negate the
    // value of the input number
    if (negative)
        number *= -1;
}

int main(){
    //cin / cout user
    //ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0)


    return 0;
}
```

1.2 FastScanner

```
class FastScanner {
    private InputStream stream;
    private byte[] buf = new byte[1024];
    private int curChar;
    private int numChars;

    public FastScanner(InputStream stream) {
        this.stream = stream;
    }

    int read() {
        if (numChars == -1)
            throw new InputMismatchException();
        if (curChar >= numChars) {
            curChar = 0;
            try {
                numChars = stream.read(buf);
            } catch (IOException e) {
                throw new InputMismatchException();
            }
            if (numChars <= 0) return -1;
        }
        return buf[curChar++];
    }

    boolean isSpaceChar(int c) {
        return c ==       || c ==   \ n   || c ==   \ r   || c
             ==   \ t   || c == -1;
    }

    public int nextInt() {
        return Integer.parseInt(next());
    }

    public long nextLong() {
        return Long.parseLong(next());
    }

    public double nextDouble() {
        return Double.parseDouble(next());
    }

    public String next() {
        int c = read();
        while (isSpaceChar(c)) c = read();
        StringBuilder res = new StringBuilder();
        do {
            res.appendCodePoint(c);
            c = read();
        } while (!isSpaceChar(c));
        return res.toString();
    }

    public String nextLine() {
        int c = read();
        while (isEndline(c))
            c = read();
        StringBuilder res = new StringBuilder();
        do {
            res.appendCodePoint(c);
            c = read();
        } while (!isEndline(c));
        return res.toString();
    }
}
```