## Fast I/O in C++

* Use scanf() and printf() instead of cin and cout (OR) use the following Fast I/O method:

Add these two lines at the starting of main() function.

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

→ Now you can use cin and cout, this is faster than the regular cin and cout.

## Fast I/O in Python(also for PyPy)

* Better submit in PyPy as it is faster than Python.

Instead of normal print() function, use the following:

Example:

Initial code:

n = input()

print(n)

Instead of the above code, use the following Fast I/O method:

import sys

input = sys.stdin.readline

n = input()

sys.stdout.write(str(n) + "\n")

# Fast I/O in Java

* Instead of the Scanner class for i/o, define a class for faster i/o.
* You may create an inner class such as FastReader to define the fast reader and its relevant methods:

static class FastReader {

BufferedReader br;

StringTokenizer st;

public FastReader() {

br = new BufferedReader(new

InputStreamReader(System.in));

}

String next() {

while (st == null || !st.hasMoreElements()) {

try {

st = new StringTokenizer(br.readLine());

}

catch (IOException e) {

e.printStackTrace();

}

}

return st.nextToken();

}

int nextInt() {

return Integer.parseInt(next());

}

long nextLong() {

return Long.parseLong(next());

}

double nextDouble() {

return Double.parseDouble(next());

}

String nextLine() {

String str = "";

try {

str = br.readLine();

}

catch (IOException e) {

e.printStackTrace();

}

return str;

}

int [] readintarray(int n) {

int res [] = new int [n];

for(int i = 0; i<n; i++)res[i] = nextInt();

return res;

}

long [] readlongarray(int n) {

long res [] = new long [n];

for(int i = 0; i<n; i++)res[i] = nextLong();

return res;

}

}