

---



12/13

I/O

# 1. BufferedReader

1) 1 2 3 4 — Integers

→ StringTokenizer st =

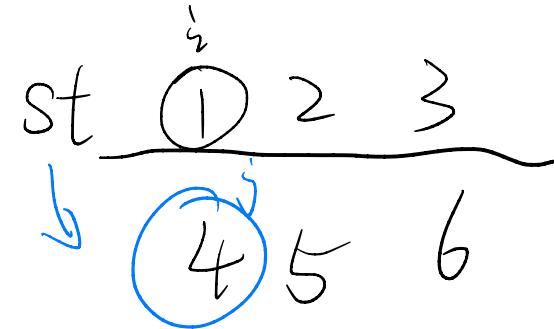
new StringTokenizer(f.readLine());

int i = Integer.parseInt(st.nextToken());

st =

→ new StringTokenizer(f.readLine());

int j = ...;



y

exe 8

→ 2) String { S = ?  
 S → abc de \n S1 = ?  
 S1 → ij ii \n

S.length() → 6 charAt()

S1.length() → 5

3) char

a b c  
 st = new S.T.(f.readLines())

char c = Character. Parsechar(st.)

## 2. PrintWriter.

BufferedWriter bw = new BufferedWriter  
(new FileWriter(" "));

PrintWriter out = new  
PrintWriter( bw );

out.print(" "); → to file.

out.close();

# ASCII

A - Z

$$\alpha = \overline{z}$$

— 9

1 2

ABC IT

char A > int

(int) A → 65

int  $A - i = \underline{(\text{int})}^{\text{of}} A$   
↓  
- 4.

0, 1 2 3 4 5  
COMET Q.

65 - 90

A - Z

↓

① char At.      1 - 26.

② (int) char - 64      1 - 26.

③ Calc.

1.

1 2 3 4 | 5 6 7 8 9 10

G G H G H G H H G G

(C) (H) (G) (H) (G) (H) (H) (G) (G)

↓ ↓ ↓ ↓ ↓ ↓

1 2 3 4 5 6 7 8 9 10

H H G H G G F G G H

(H) (G) (H) (G) (G) (F) (G) (G) (H)

X

~~20000~~

1 2 3 4 5 6 > 7 9 10 11 12 13 14

→ G G G H G H H H G A A H G H G

G H G H H H G H G H G H G

~~A~~

j=4. 6. 8. 10. 12. 14.

max 24 (6) 3. 3. 2. 1.

O<sub>1</sub> for (i=0; i < len; i++)

2 for(j=0; j < len; j+=2)

reverse

Umax matory even G.

)

)

print l'')

GGG H G H G A T G A H H G G G

G H | G G H | G

even not G  $\rightarrow$  G (max)

2)

ans = 0

ans = 2

for(j = n-2; j >= 0; j -= 2)

(if

str[j] == str[j+1] &

(str[j] == 'G') == ==

(ans / 2) == 0

ans++ ; )