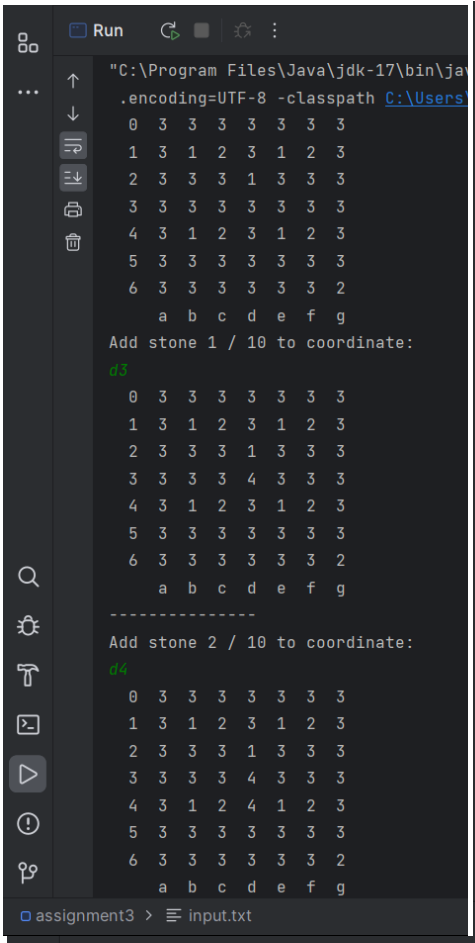
Main Class

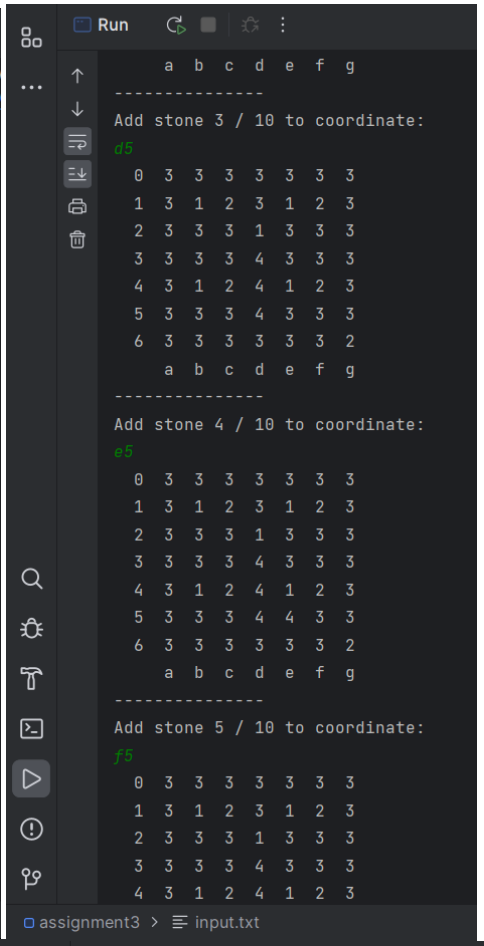
After opening the file in the main method, the rows of the text file are iterated through. After determining the matrice size, the height values of the given terrain are stored in an arraylist called “terrain” that contains integer arrays. After collecting all necessary data from the text file, there is a while loop to get 10 inputs from the user. Until 10 moves are made the loop continues working. Within it the correction of the user input is checked and if incorrect input is detected the necessary message is printed out. Right after exiting the while loop, a 2d arraylist with the same size of the terrain matrice is initialized and the moneyFlow method is called to detect the puddles that can contain money. After that the terrain arraylist is deep copied. So that in the namingLakes method when the values of the corresponding elements are changed, the original values would not be lost. The aim to change the corresponding elements is to tag them if they are money lakes. Finally the last version of the terrain with the name of the lakes and the final score are printed out.

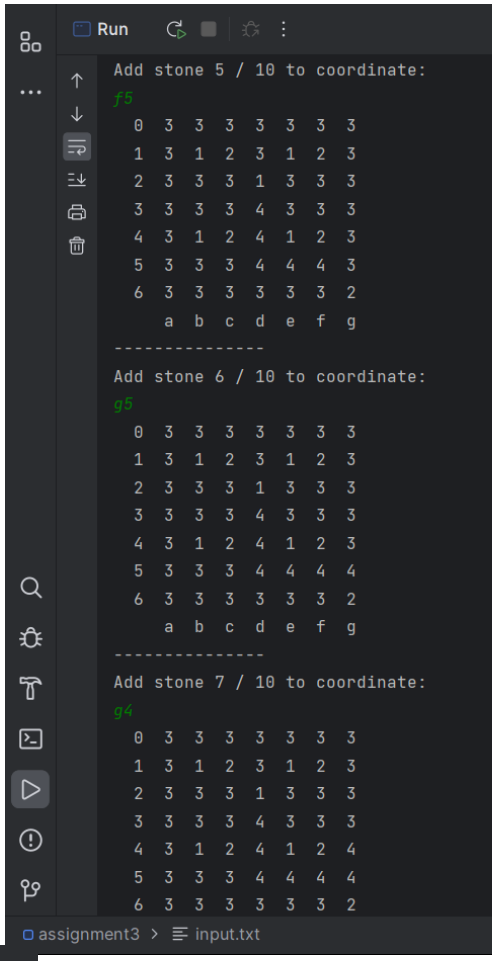
Terrain Class

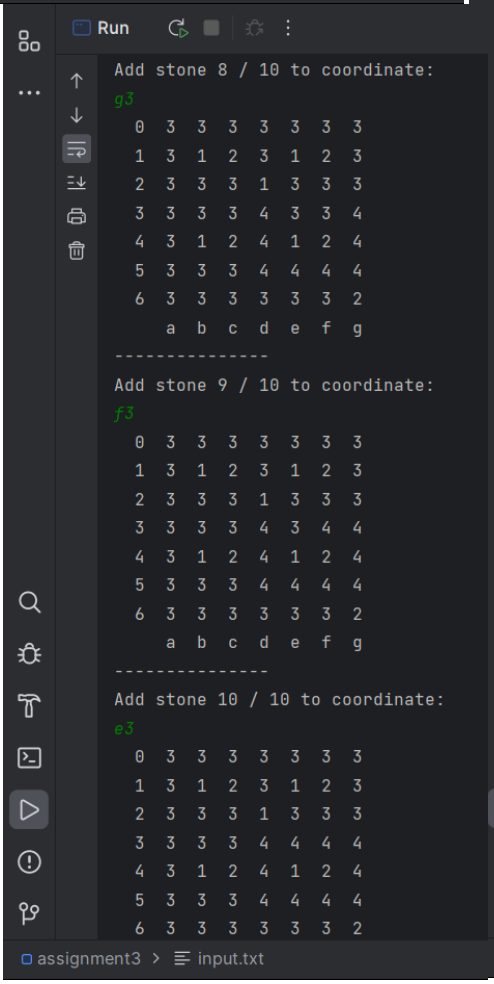
Firstly the moneyFlow method traverses through the matrix, deciding for each element individually whether it is a lake or not and stores this information in a 2d boolean array by changing the corresponding value in the boolean array to true if it is a lake. For traversing the matrix there are two for loops and a while loop within them. Before the while an arraylist called “flow” is initialized and the current cell is added to it. There are two conditions for the while loop to finish functioning: if a boundary point is found while checking the neighbors of a cell or if the flow is completely emptied. It should be noted that a neighbor of the current cell is added to flow if it is a possible lake and then the indexes that are controlled in the while loop (n1, m1) are updated to its index values. Lastly it is removed from the flow.
In the namingLakes method a similar algorithm is used. Once again there is an arraylist called flow if the current cell is a lake, then in the while loop its neighbors are checked if they are also lakes they are added to the flow and their neighbors are also checked. This time only way for entering the while loop for a cell was being a lake. In the while loop all the neighbor cells that are also lakes are marked with the same integer value that starts from 0. If the while loop has not worked for the current cell meaning the boolean flag “in” has never been set to true, there is no need to increment the integer value. Otherwise it would not indicate the number of the lakes.
In the volume method first of all each lake’s integer value in the copyTerrain which identifies its name is stored in an arraylist called lakeCount. lakeCount’s size is actually shows the number of the lakes. Secondly for each lake that is stored in the lakeCount the smallest height around it is stored in the smallestBorders. That way for a specific lake its name’s index in the lakeCount and its smallest border’s index in smallestBorders are equal. For one last time the matrix is traversed and if its value in copyTerrain is same as the integer in lakeCount representing its name, then for its smallest border value the money volume in the current cell is calculated and added to the sum. Right after calculating the volume for each cell in a lake the final sum’s square root is added to the volume.
To print out the terrain with the names of the lakes I created a method called printTerrainLast. In this method the matrix is once again traversed and the cell is simply printed out if it is not a lake. In case it is a lake and its value in the copyTerrain is less than 25, its value is simply increased by 65 and it is printed out as a character. If it is greater than 25, for its name two letters should be used.

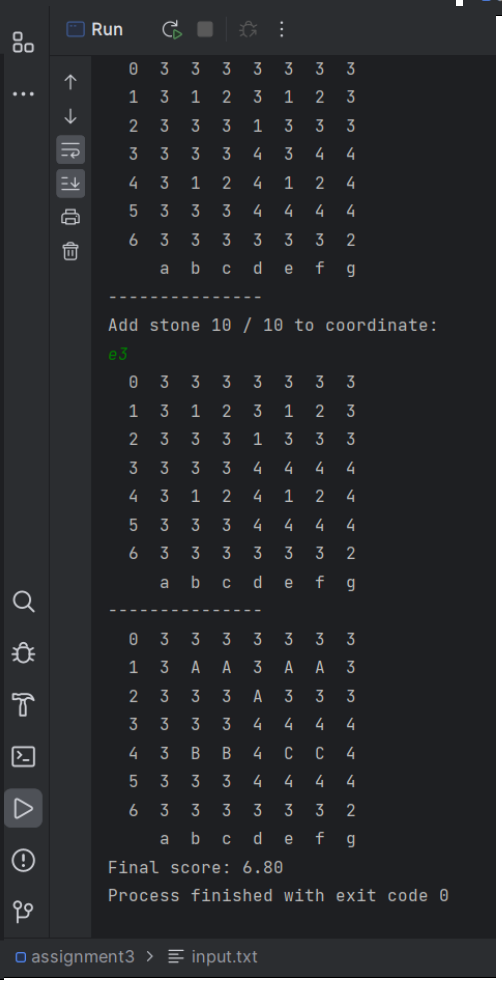
Input 1


Run
"C:\Program Files\Java\jdk-17\bin\ja
.encoding=UTF-8 -classpath C:\Users
0 3 3 3 3 3 3 3
1 3 1 2 3 1 2 3
2 3 3 3 1 3 3 3
3 3 3 3 3 3 3 3
4 3 1 2 3 1 2 3
5 3 3 3 3 3 3 3
6 3 3 3 3 3 3 2
a b c d e f g
Add stone 1 / 10 to coordinate:
d3
0 3 3 3 3 3 3 3
1 3 1 2 3 1 2 3
2 3 3 3 1 3 3 3
3 3 3 3 4 3 3 3
4 3 1 2 3 1 2 3
5 3 3 3 3 3 3 3
6 3 3 3 3 3 3 2
a b c d e f g
Add stone 2 / 10 to coordinate:
d4
0 3 3 3 3 3 3 3
1 3 1 2 3 1 2 3
2 3 3 3 1 3 3 3
3 3 3 3 4 3 3 3
4 3 1 2 4 1 2 3
5 3 3 3 3 3 3 3
6 3 3 3 3 3 3 2
a b c d e f g
assignment3 > input.txt

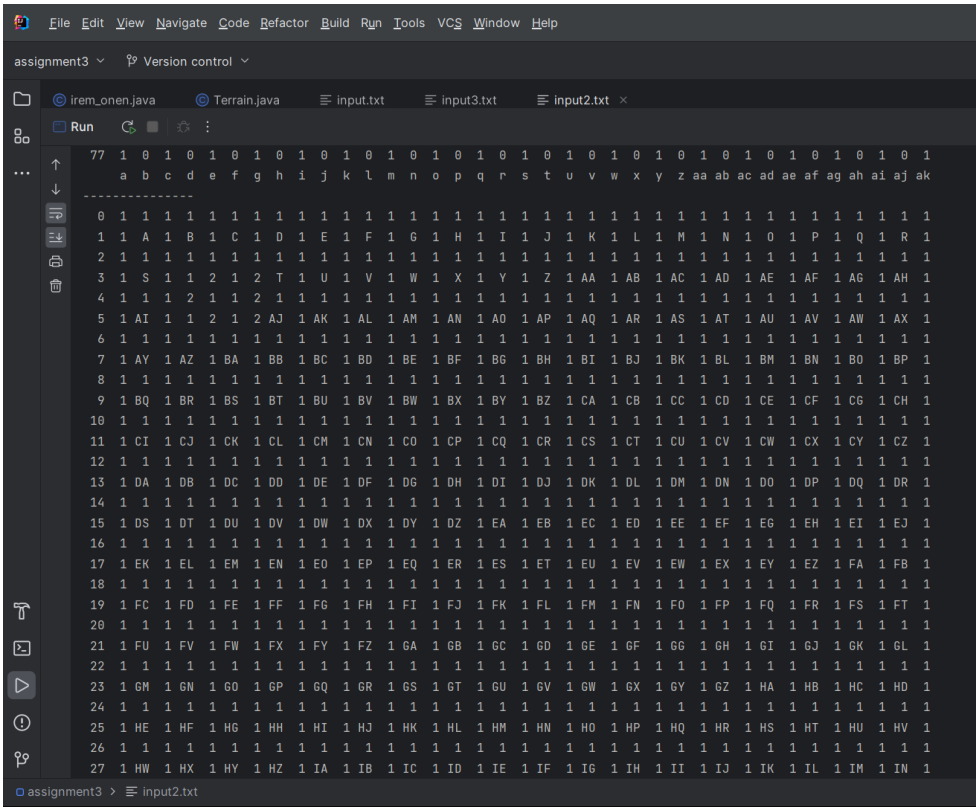

Run
a b c d e f g
Add stone 3 / 10 to coordinate:
d5
0 3 3 3 3 3 3 3
1 3 1 2 3 1 2 3
2 3 3 3 1 3 3 3
3 3 3 3 4 3 3 3
4 3 1 2 4 1 2 3
5 3 3 3 4 3 3 3
6 3 3 3 3 3 3 2
a b c d e f g
Add stone 4 / 10 to coordinate:
e5
0 3 3 3 3 3 3 3
1 3 1 2 3 1 2 3
2 3 3 3 1 3 3 3
3 3 3 3 4 3 3 3
4 3 1 2 4 1 2 3
5 3 3 3 4 4 3 3
6 3 3 3 3 3 3 2
a b c d e f g
Add stone 5 / 10 to coordinate:
f5
0 3 3 3 3 3 3 3
1 3 1 2 3 1 2 3
2 3 3 3 1 3 3 3
3 3 3 3 4 3 3 3
4 3 1 2 4 1 2 3
5 3 3 3 4 4 3 3
6 3 3 3 3 3 3 2
a b c d e f g
assignment3 > input.txt

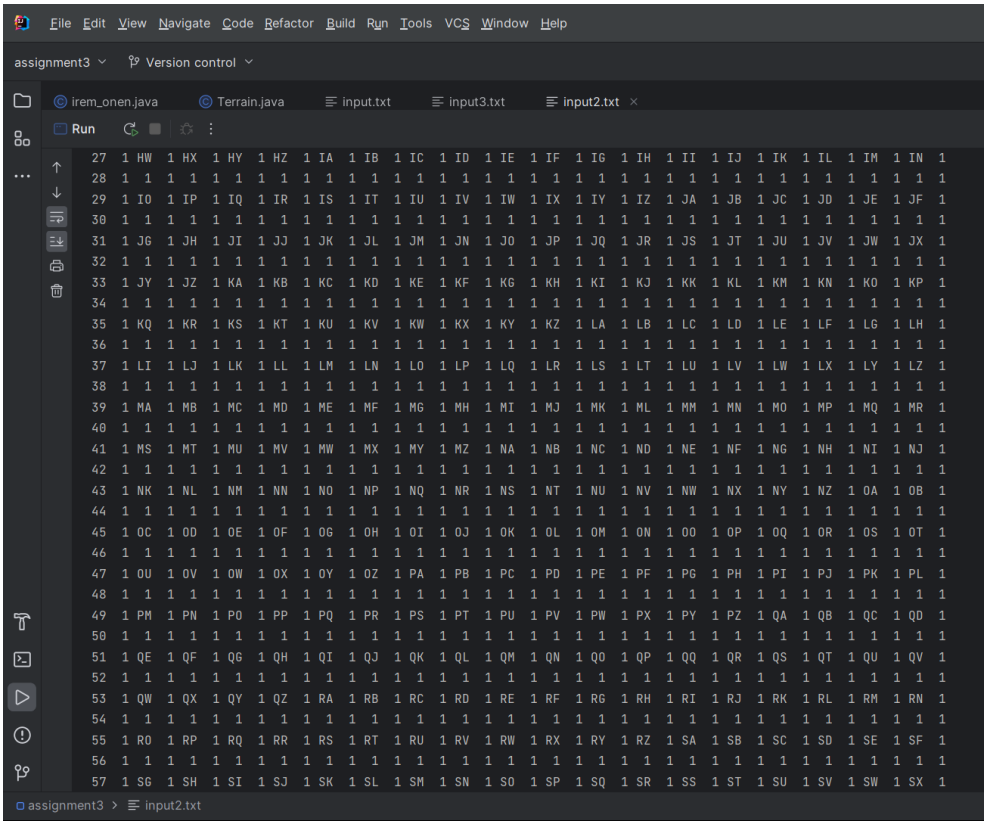

Run
Add stone 5 / 10 to coordinate:
f5
0 3 3 3 3 3 3 3
1 3 1 2 3 1 2 3
2 3 3 3 1 3 3 3
3 3 3 3 4 3 3 3
4 3 1 2 4 1 2 3
5 3 3 3 4 4 4 3
6 3 3 3 3 3 3 2
a b c d e f g
Add stone 6 / 10 to coordinate:
g5
0 3 3 3 3 3 3 3
1 3 1 2 3 1 2 3
2 3 3 3 1 3 3 3
3 3 3 3 4 3 3 3
4 3 1 2 4 1 2 3
5 3 3 3 4 4 4 4
6 3 3 3 3 3 3 2
a b c d e f g
Add stone 7 / 10 to coordinate:
g4
0 3 3 3 3 3 3 3
1 3 1 2 3 1 2 3
2 3 3 3 1 3 3 3
3 3 3 3 4 3 3 3
4 3 1 2 4 1 2 4
5 3 3 3 4 4 4 4
6 3 3 3 3 3 3 2
assignment3 > input.txt


Run
Add stone 8 / 10 to coordinate:
g3
0 3 3 3 3 3 3 3
1 3 1 2 3 1 2 3
2 3 3 3 1 3 3 3
3 3 3 3 4 3 3 4
4 3 1 2 4 1 2 4
5 3 3 3 4 4 4 4
6 3 3 3 3 3 3 2
a b c d e f g
Add stone 9 / 10 to coordinate:
f3
0 3 3 3 3 3 3 3
1 3 1 2 3 1 2 3
2 3 3 3 1 3 3 3
3 3 3 3 4 3 4 4
4 3 1 2 4 1 2 4
5 3 3 3 4 4 4 4
6 3 3 3 3 3 3 2
a b c d e f g
Add stone 10 / 10 to coordinate:
e3
0 3 3 3 3 3 3 3
1 3 1 2 3 1 2 3
2 3 3 3 1 3 3 3
3 3 3 3 4 4 4 4
4 3 1 2 4 1 2 4
5 3 3 3 4 4 4 4
6 3 3 3 3 3 3 2
assignment3 > input.txt

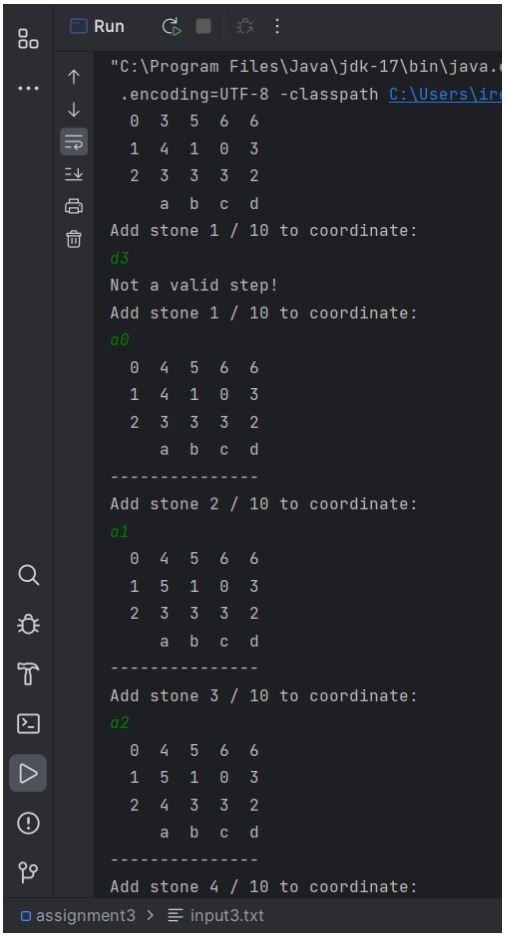

Run
0 3 3 3 3 3 3 3
1 3 1 2 3 1 2 3
2 3 3 3 1 3 3 3
3 3 3 3 4 3 4 4
4 3 1 2 4 1 2 4
5 3 3 3 4 4 4 4
6 3 3 3 3 3 3 2
a b c d e f g
Add stone 10 / 10 to coordinate:
e3
0 3 3 3 3 3 3 3
1 3 1 2 3 1 2 3
2 3 3 3 1 3 3 3
3 3 3 3 4 4 4 4
4 3 1 2 4 1 2 4
5 3 3 3 4 4 4 4
6 3 3 3 3 3 3 2
a b c d e f g
0 3 3 3 3 3 3 3
1 3 A A 3 A A 3
2 3 3 3 A 3 3 3
3 3 3 3 4 4 4 4
4 3 B B 4 C C 4
5 3 3 3 4 4 4 4
6 3 3 3 3 3 3 2
a b c d e f g
Final score: 6.80
Process finished with exit code 0
assignment3 > input.txt

Input 2


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
assignment3
Version control
irem_onen.java
Terrain.java
input.txt
input3.txt
input2.txt
Run
77 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
a b c d e f g h i j k l m n o p q r s t u v w x y z aa ab ac ad ae af ag ah ai aj ak
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 A 1 B 1 C 1 D 1 E 1 F 1 G 1 H 1 I 1 J 1 K 1 L 1 M 1 N 1 O 1 P 1 Q 1 R 1
2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
3 1 S 1 1 2 1 2 T 1 U 1 V 1 W 1 X 1 Y 1 Z 1 AA 1 AB 1 AC 1 AD 1 AE 1 AF 1 AG 1 AH 1
4 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
5 1 AI 1 1 2 1 2 AJ 1 AK 1 AL 1 AM 1 AN 1 AO 1 AP 1 AQ 1 AR 1 AS 1 AT 1 AU 1 AV 1 AW 1 AX 1
6 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
7 1 AY 1 AZ 1 BA 1 BB 1 BC 1 BD 1 BE 1 BF 1 BG 1 BH 1 BI 1 BJ 1 BK 1 BL 1 BM 1 BN 1 BO 1 BP 1
8 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
9 1 BQ 1 BR 1 BS 1 BT 1 BU 1 BV 1 BW 1 BX 1 BY 1 BZ 1 CA 1 CB 1 CC 1 CD 1 CE 1 CF 1 CG 1 CH 1
10 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
11 1 CI 1 CJ 1 CK 1 CL 1 CM 1 CN 1 CO 1 CP 1 CQ 1 CR 1 CS 1 CT 1 CU 1 CV 1 CW 1 CX 1 CY 1 CZ 1
12 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
13 1 DA 1 DB 1 DC 1 DD 1 DE 1 DF 1 DG 1 DH 1 DI 1 DJ 1 DK 1 DL 1 DM 1 DN 1 DO 1 DP 1 DQ 1 DR 1
14 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
15 1 DS 1 DT 1 DU 1 DV 1 DW 1 DX 1 DY 1 DZ 1 EA 1 EB 1 EC 1 ED 1 EE 1 EF 1 EG 1 EH 1 EI 1 EJ 1
16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
17 1 EK 1 EL 1 EM 1 EN 1 EO 1 EP 1 EQ 1 ER 1 ES 1 ET 1 EU 1 EV 1 EW 1 EX 1 EY 1 EZ 1 FA 1 FB 1
18 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
19 1 FC 1 FD 1 FE 1 FF 1 FG 1 FH 1 FI 1 FJ 1 FK 1 FL 1 FM 1 FN 1 FO 1 FP 1 FQ 1 FR 1 FS 1 FT 1
20 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
21 1 FU 1 FV 1 FW 1 FX 1 FY 1 FZ 1 GA 1 GB 1 GC 1 GD 1 GE 1 GF 1 GG 1 GH 1 GI 1 GJ 1 GK 1 GL 1
22 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
23 1 GM 1 GN 1 GO 1 GP 1 GQ 1 GR 1 GS 1 GT 1 GU 1 GV 1 GW 1 GX 1 GY 1 GZ 1 HA 1 HB 1 HC 1 HD 1
24 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
25 1 HE 1 HF 1 HG 1 HH 1 HI 1 HJ 1 HK 1 HL 1 HM 1 HN 1 HO 1 HP 1 HQ 1 HR 1 HS 1 HT 1 HU 1 HV 1
26 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
27 1 HW 1 HX 1 HY 1 HZ 1 IA 1 IB 1 IC 1 ID 1 IE 1 IF 1 IG 1 IH 1 II 1 IJ 1 IK 1 IL 1 IM 1 IN 1
assignment3 > input2.txt


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
assignment3
Version control
irem_onen.java
Terrain.java
input.txt
input3.txt
input2.txt
Run
27 1 HW 1 HX 1 HY 1 HZ 1 IA 1 IB 1 IC 1 ID 1 IE 1 IF 1 IG 1 IH 1 II 1 IJ 1 IK 1 IL 1 IM 1 IN 1
28 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
29 1 IO 1 IP 1 IQ 1 IR 1 IS 1 IT 1 IU 1 IV 1 IW 1 IX 1 IY 1 IZ 1 JA 1 JB 1 JC 1 JD 1 JE 1 JF 1
30 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
31 1 JG 1 JH 1 JI 1 JJ 1 JK 1 JL 1 JM 1 JN 1 JO 1 JP 1 JQ 1 JR 1 JS 1 JT 1 JU 1 JV 1 JW 1 JX 1
32 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
33 1 JY 1 JZ 1 KA 1 KB 1 KC 1 KD 1 KE 1 KF 1 KG 1 KH 1 KI 1 KJ 1 KK 1 KL 1 KM 1 KN 1 KO 1 KP 1
34 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
35 1 KQ 1 KR 1 KS 1 KT 1 KU 1 KV 1 KW 1 KX 1 KY 1 KZ 1 LA 1 LB 1 LC 1 LD 1 LE 1 LF 1 LG 1 LH 1
36 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
37 1 LI 1 LJ 1 LK 1 LL 1 LM 1 LN 1 LO 1 LP 1 LQ 1 LR 1 LS 1 LT 1 LU 1 LV 1 LW 1 LX 1 LY 1 LZ 1
38 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
39 1 MA 1 MB 1 MC 1 MD 1 ME 1 MF 1 MG 1 MH 1 MI 1 MJ 1 MK 1 ML 1 MM 1 MN 1 MO 1 MP 1 MQ 1 MR 1
40 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
41 1 MS 1 MT 1 MU 1 MV 1 MW 1 MX 1 MY 1 MZ 1 NA 1 NB 1 NC 1 ND 1 NE 1 NF 1 NG 1 NH 1 NI 1 NJ 1
42 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
43 1 NK 1 NL 1 NM 1 NN 1 NO 1 NP 1 NQ 1 NR 1 NS 1 NT 1 NU 1 NV 1 NW 1 NX 1 NY 1 NZ 1 OA 1 OB 1
44 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
45 1 OC 1 OD 1 OE 1 OF 1 OG 1 OH 1 OI 1 OJ 1 OK 1 OL 1 OM 1 ON 1 OO 1 OP 1 OQ 1 OR 1 OS 1 OT 1
46 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
47 1 OU 1 OV 1 OW 1 OX 1 OY 1 OZ 1 PA 1 PB 1 PC 1 PD 1 PE 1 PF 1 PG 1 PH 1 PI 1 PJ 1 PK 1 PL 1
48 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
49 1 PM 1 PN 1 PO 1 PP 1 PQ 1 PR 1 PS 1 PT 1 PU 1 PV 1 PW 1 PX 1 PY 1 PZ 1 QA 1 QB 1 QC 1 QD 1
50 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
51 1 QE 1 QF 1 QG 1 QH 1 QI 1 QJ 1 QK 1 QL 1 QM 1 QN 1 QO 1 QP 1 QQ 1 QR 1 QS 1 QT 1 QU 1 QV 1
52 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
53 1 QW 1 QX 1 QY 1 QZ 1 RA 1 RB 1 RC 1 RD 1 RE 1 RF 1 RG 1 RH 1 RI 1 RJ 1 RK 1 RL 1 RM 1 RN 1
54 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
55 1 RO 1 RP 1 RQ 1 RR 1 RS 1 RT 1 RU 1 RV 1 RW 1 RX 1 RY 1 RZ 1 SA 1 SB 1 SC 1 SD 1 SE 1 SF 1
56 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
57 1 SG 1 SH 1 SI 1 SJ 1 SK 1 SL 1 SM 1 SN 1 SO 1 SP 1 SQ 1 SR 1 SS 1 ST 1 SU 1 SV 1 SW 1 SX 1
assignment3 > input2.txt

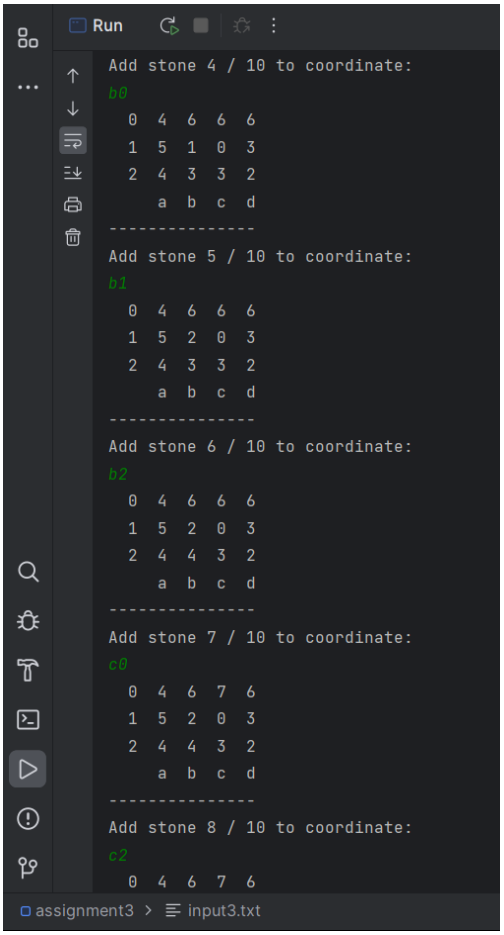
Input 3


Run
"C:\Program Files\Java\jdk-17\bin\java.
.encoding=UTF-8 -classpath C:\Users\ir
0 3 5 6 6
1 4 1 0 3
2 3 3 3 2
a b c d
Add stone 1 / 10 to coordinate:
d3
Not a valid step!
Add stone 1 / 10 to coordinate:
a0
0 4 5 6 6
1 4 1 0 3
2 3 3 3 2
a b c d

Add stone 2 / 10 to coordinate:
a1
0 4 5 6 6
1 5 1 0 3
2 3 3 3 2
a b c d

Add stone 3 / 10 to coordinate:
a2
0 4 5 6 6
1 5 1 0 3
2 4 3 3 2
a b c d

Add stone 4 / 10 to coordinate:
assignment3 > input3.txt

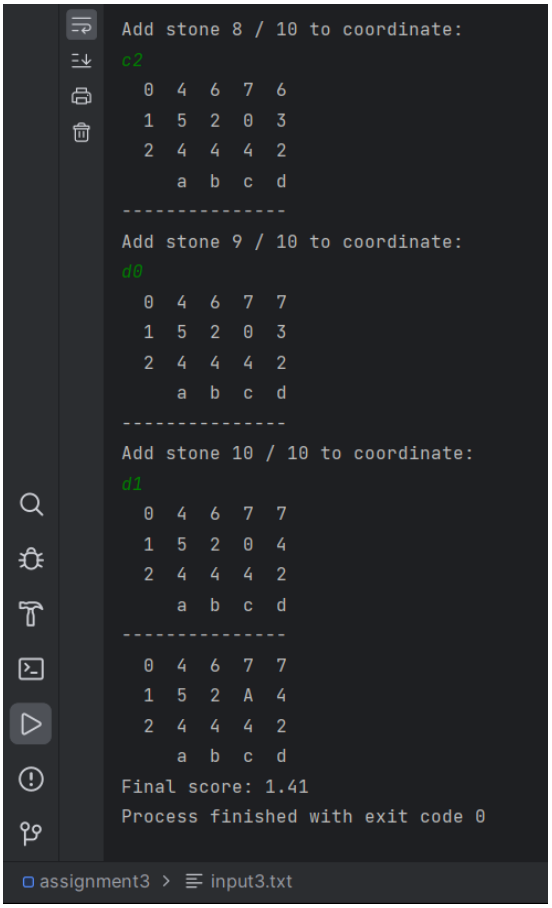

Run
Add stone 4 / 10 to coordinate:
b0
0 4 6 6 6
1 5 1 0 3
2 4 3 3 2
a b c d

Add stone 5 / 10 to coordinate:
b1
0 4 6 6 6
1 5 2 0 3
2 4 3 3 2
a b c d

Add stone 6 / 10 to coordinate:
b2
0 4 6 6 6
1 5 2 0 3
2 4 4 3 2
a b c d

Add stone 7 / 10 to coordinate:
c0
0 4 6 7 6
1 5 2 0 3
2 4 4 3 2
a b c d

Add stone 8 / 10 to coordinate:
c2
0 4 6 7 6
assignment3 > input3.txt


Add stone 8 / 10 to coordinate:
c2
0 4 6 7 6
1 5 2 0 3
2 4 4 4 2
a b c d

Add stone 9 / 10 to coordinate:
d0
0 4 6 7 7
1 5 2 0 3
2 4 4 4 2
a b c d

Add stone 10 / 10 to coordinate:
d1
0 4 6 7 7
1 5 2 0 4
2 4 4 4 2
a b c d
---
0 4 6 7 7
1 5 2 A 4
2 4 4 4 2
a b c d
Final score: 1.41
Process finished with exit code 0
assignment3 > input3.txt