

Programming Using C

week 05 practice session coding

Name:K.Kamaleshwaran

Department:AIML-'B'

Roll No.:242501079

Write a program that prints a simple chessboard.

Input format

The first line contains the number of squares N .

The lines after that contain a different value to control the chessboard.

Output format

Print a chessboard of dimensions $8 \times N$ size. Print a Field W for white squares and B for black squares.

Input

8
1
0

Output

WBWBWBWB
WBWBWBWB
WBWBWBWB
WBWBWBWB
WBWBWBWB
WBWBWBWB
WBWBWBWB
WBWBWBWB

Answer (currently requires 0 %)

```
def chessboard(n):  
    for i in range(1, n+1):  
        row = ''  
        for j in range(1, 8+1):  
            if (i+j)%2 == 1:  
                row += 'W'  
            else:  
                row += 'B'  
        print(row)
```

Input	Expected	Got
8	WBWBWBWB	WBWBWBWB
1	WBWBWBWB	WBWBWBWB
0	WBWBWBWB	WBWBWBWB
WBWBWBWB	WBWBWBWB	WBWBWBWB
WBWBWBWB	WBWBWBWB	WBWBWBWB
WBWBWBWB	WBWBWBWB	WBWBWBWB
WBWBWBWB	WBWBWBWB	WBWBWBWB
WBWBWBWB	WBWBWBWB	WBWBWBWB

Percent of tests: 100%

Report all incidents.

```

1  // Two Subsets of Size k
2  // per subset
3  int m = (int) Math.floor( (n + 1) / 2.0 );
4  int[] s1 = new int[m];
5  int[] s2 = new int[m];
6
7  // s1 contains the first m elements
8  for (int i = 0; i < m; i++)
9  {
10     s1[i] = i + 1;
11 }
12
13 // s2 contains the last m elements
14 for (int i = 0; i < m; i++)
15 {
16     s2[i] = i + m + 1;
17 }
18
19 // Print the two subsets
20 for (int i = 0; i < m; i++)
21 {
22     System.out.print(s1[i] + " ");
23 }
24
25 for (int i = 0; i < m; i++)
26 {
27     System.out.print(s2[i] + " ");
28 }
29
30 // Print the two subsets
31 for (int i = 0; i < m; i++)
32 {
33     System.out.print(s1[i] + " ");
34 }
35
36 for (int i = 0; i < m; i++)
37 {
38     System.out.print(s2[i] + " ");
39 }
40
41 // Print the two subsets
42 for (int i = 0; i < m; i++)
43 {
44     System.out.print(s1[i] + " ");
45 }
46
47 for (int i = 0; i < m; i++)
48 {
49     System.out.print(s2[i] + " ");
50 }
51
52 // Print the two subsets
53 for (int i = 0; i < m; i++)
54 {
55     System.out.print(s1[i] + " ");
56 }
57
58 for (int i = 0; i < m; i++)
59 {
60     System.out.print(s2[i] + " ");
61 }
62
63 // Print the two subsets
64 for (int i = 0; i < m; i++)
65 {
66     System.out.print(s1[i] + " ");
67 }
68
69 for (int i = 0; i < m; i++)
70 {
71     System.out.print(s2[i] + " ");
72 }
73
74 // Print the two subsets
75 for (int i = 0; i < m; i++)
76 {
77     System.out.print(s1[i] + " ");
78 }
79
80 for (int i = 0; i < m; i++)
81 {
82     System.out.print(s2[i] + " ");
83 }
84
85 // Print the two subsets
86 for (int i = 0; i < m; i++)
87 {
88     System.out.print(s1[i] + " ");
89 }
90
91 for (int i = 0; i < m; i++)
92 {
93     System.out.print(s2[i] + " ");
94 }
95
96 // Print the two subsets
97 for (int i = 0; i < m; i++)
98 {
99     System.out.print(s1[i] + " ");
100 }
101
102 for (int i = 0; i < m; i++)
103 {
104     System.out.print(s2[i] + " ");
105 }
106
107 // Print the two subsets
108 for (int i = 0; i < m; i++)
109 {
110     System.out.print(s1[i] + " ");
111 }
112
113 for (int i = 0; i < m; i++)
114 {
115     System.out.print(s2[i] + " ");
116 }
117
118 // Print the two subsets
119 for (int i = 0; i < m; i++)
120 {
121     System.out.print(s1[i] + " ");
122 }
123
124 for (int i = 0; i < m; i++)
125 {
126     System.out.print(s2[i] + " ");
127 }
128
129 // Print the two subsets
130 for (int i = 0; i < m; i++)
131 {
132     System.out.print(s1[i] + " ");
133 }
134
135 for (int i = 0; i < m; i++)
136 {
137     System.out.print(s2[i] + " ");
138 }
139
140 // Print the two subsets
141 for (int i = 0; i < m; i++)
142 {
143     System.out.print(s1[i] + " ");
144 }
145
146 for (int i = 0; i < m; i++)
147 {
148     System.out.print(s2[i] + " ");
149 }
150
151 // Print the two subsets
152 for (int i = 0; i < m; i++)
153 {
154     System.out.print(s1[i] + " ");
155 }
156
157 for (int i = 0; i < m; i++)
158 {
159     System.out.print(s2[i] + " ");
160 }
161
162 // Print the two subsets
163 for (int i = 0; i < m; i++)
164 {
165     System.out.print(s1[i] + " ");
166 }
167
168 for (int i = 0; i < m; i++)
169 {
170     System.out.print(s2[i] + " ");
171 }
172
173 // Print the two subsets
174 for (int i = 0; i < m; i++)
175 {
176     System.out.print(s1[i] + " ");
177 }
178
179 for (int i = 0; i < m; i++)
180 {
181     System.out.print(s2[i] + " ");
182 }
183
184 // Print the two subsets
185 for (int i = 0; i < m; i++)
186 {
187     System.out.print(s1[i] + " ");
188 }
189
190 for (int i = 0; i < m; i++)
191 {
192     System.out.print(s2[i] + " ");
193 }
194
195 // Print the two subsets
196 for (int i = 0; i < m; i++)
197 {
198     System.out.print(s1[i] + " ");
199 }
200
201 for (int i = 0; i < m; i++)
202 {
203     System.out.print(s2[i] + " ");
204 }
205
206 // Print the two subsets
207 for (int i = 0; i < m; i++)
208 {
209     System.out.print(s1[i] + " ");
210 }
211
212 for (int i = 0; i < m; i++)
213 {
214     System.out.print(s2[i] + " ");
215 }
216
217 // Print the two subsets
218 for (int i = 0; i < m; i++)
219 {
220     System.out.print(s1[i] + " ");
221 }
222
223 for (int i = 0; i < m; i++)
224 {
225     System.out.print(s2[i] + " ");
226 }
227
228 // Print the two subsets
229 for (int i = 0; i < m; i++)
230 {
231     System.out.print(s1[i] + " ");
232 }
233
234 for (int i = 0; i < m; i++)
235 {
236     System.out.print(s2[i] + " ");
237 }
238
239 // Print the two subsets
240 for (int i = 0; i < m; i++)
241 {
242     System.out.print(s1[i] + " ");
243 }
244
245 for (int i = 0; i < m; i++)
246 {
247     System.out.print(s2[i] + " ");
248 }
249
250 // Print the two subsets
251 for (int i = 0; i < m; i++)
252 {
253     System.out.print(s1[i] + " ");
254 }
255
256 for (int i = 0; i < m; i++)
257 {
258     System.out.print(s2[i] + " ");
259 }
260
261 // Print the two subsets
262 for (int i = 0; i < m; i++)
263 {
264     System.out.print(s1[i] + " ");
265 }
266
267 for (int i = 0; i < m; i++)
268 {
269     System.out.print(s2[i] + " ");
270 }
271
272 // Print the two subsets
273 for (int i = 0; i < m; i++)
274 {
275     System.out.print(s1[i] + " ");
276 }
277
278 for (int i = 0; i < m; i++)
279 {
280     System.out.print(s2[i] + " ");
281 }
282
283 // Print the two subsets
284 for (int i = 0; i < m; i++)
285 {
286     System.out.print(s1[i] + " ");
287 }
288
289 for (int i = 0; i < m; i++)
290 {
291     System.out.print(s2[i] + " ");
292 }
293
294 // Print the two subsets
295 for (int i = 0; i < m; i++)
296 {
297     System.out.print(s1[i] + " ");
298 }
299
300 for (int i = 0; i < m; i++)
301 {
302     System.out.print(s2[i] + " ");
303 }
304
305 // Print the two subsets
306 for (int i = 0; i < m; i++)
307 {
308     System.out.print(s1[i] + " ");
309 }
310
311 for (int i = 0; i < m; i++)
312 {
313     System.out.print(s2[i] + " ");
314 }
315
316 // Print the two subsets
317 for (int i = 0; i < m; i++)
318 {
319     System.out.print(s1[i] + " ");
320 }
321
322 for (int i = 0; i < m; i++)
323 {
324     System.out.print(s2[i] + " ");
325 }
326
327 // Print the two subsets
328 for (int i = 0; i < m; i++)
329 {
330     System.out.print(s1[i] + " ");
331 }
332
333 for (int i = 0; i < m; i++)
334 {
335     System.out.print(s2[i] + " ");
336 }
337
338 // Print the two subsets
339 for (int i = 0; i < m; i++)
340 {
341     System.out.print(s1[i] + " ");
342 }
343
344 for (int i = 0; i < m; i++)
345 {
346     System.out.print(s2[i] + " ");
347 }
348
349 // Print the two subsets
350 for (int i = 0; i < m; i++)
351 {
352     System.out.print(s1[i] + " ");
353 }
354
355 for (int i = 0; i < m; i++)
356 {
357     System.out.print(s2[i] + " ");
358 }
359
360 // Print the two subsets
361 for (int i = 0; i < m; i++)
362 {
363     System.out.print(s1[i] + " ");
364 }
365
366 for (int i = 0; i < m; i++)
367 {
368     System.out.print(s2[i] + " ");
369 }
370
371 // Print the two subsets
372 for (int i = 0; i < m; i++)
373 {
374     System.out.print(s1[i] + " ");
375 }
376
377 for (int i = 0; i < m; i++)
378 {
379     System.out.print(s2[i] + " ");
380 }
381
382 // Print the two subsets
383 for (int i = 0; i < m; i++)
384 {
385     System.out.print(s1[i] + " ");
386 }
387
388 for (int i = 0; i < m; i++)
389 {
390     System.out.print(s2[i] + " ");
391 }
392
393 // Print the two subsets
394 for (int i = 0; i < m; i++)
395 {
396     System.out.print(s1[i] + " ");
397 }
398
399 for (int i = 0; i < m; i++)
400 {
401     System.out.print(s2[i] + " ");
402 }
403
404 // Print the two subsets
405 for (int i = 0; i < m; i++)
406 {
407     System.out.print(s1[i] + " ");
408 }
409
410 for (int i = 0; i < m; i++)
411 {
412     System.out.print(s2[i] + " ");
413 }
414
415 // Print the two subsets
416 for (int i = 0; i < m; i++)
417 {
418     System.out.print(s1[i] + " ");
419 }
420
421 for (int i = 0; i < m; i++)
422 {
423     System.out.print(s2[i] + " ");
424 }
425
426 // Print the two subsets
427 for (int i = 0; i < m; i++)
428 {
429
```

4

Question 2

Correct

Marked out of 5.00

Flag question

Take a number, reverse it and add it to the original number until the obtained number is a palindrome. Constraints $1 \leq \text{num} \leq 99999999$ Sample Input 1 32 Sample Output 1 55 Sample Input 2 789 Sample Output 2 66066

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int rn,n,nt=0,i=0;
5     scanf("%d",&n);
6     do
7     {
8         nt=n;
9         rn=0;
10        while(n!=0)
11        {
12            rn=rn*10 + n%10;
13            n/=10;
14        }
15        n=nt+rn;
16        i++;
17    }
18    while(rn!=nt || i==1);
19    {
20        printf("%d",rn);
21    }
22    return 0;
23 }
```

	Input	Expected	Got	
✓	32	55	55	✓
✓	789	66066	66066	✓

Passed all tests! ✓

Thames 3
 Thames
 Medford
 100
 100
 100

A number is considered lucky if it contains either 7 or 4 or 2 and 4 both in it. Write a program to give the nth lucky number. Example, 1st lucky number is 2, and 2nd lucky number is 4 and 3rd lucky number is 22 and 4th lucky number is 34 and so on. Note that 13, 40 etc., are not lucky as they have other numbers in it.

The program should accept a number 'n' as input and display the nth lucky number as output.

Sample input 1:

3

Sample Output

22

Experiment 3.02

reversal. But looking at

2019/04/23 | Page 2

2000

... ..

Answer: (percentage increase = 0.5%)

	Input	Expected	Got	
✓	34	11144	11144	✓

Passed all tests! ✓