

Report

Course: System Programming

regular expression:

$(b \cup \epsilon)(ab)^* aa(ba)^*(b \cup \epsilon)$

$b(a \cup \epsilon)(ba)^* bb(ab)^*(a \cup \epsilon)$

Implementation:

```
public class RE {  
  
    public static void main(String[] args) {  
  
        Scanner sc=new Scanner(System.in);  
        System.out.println("Enter Regular Expression for First case ");  
        String m=sc.next();  
        System.out.println(Pattern.matches("[b]?(ab)*[a]{2}(ba)*[b]?", m));  
        System.out.println("Enter Regular Expression for Second case ");  
        String n=sc.next();  
        System.out.println(Pattern.matches("[a]?(ba)*[b]{2}(ab)*[a]?", n));  
  
    }  
  
}
```

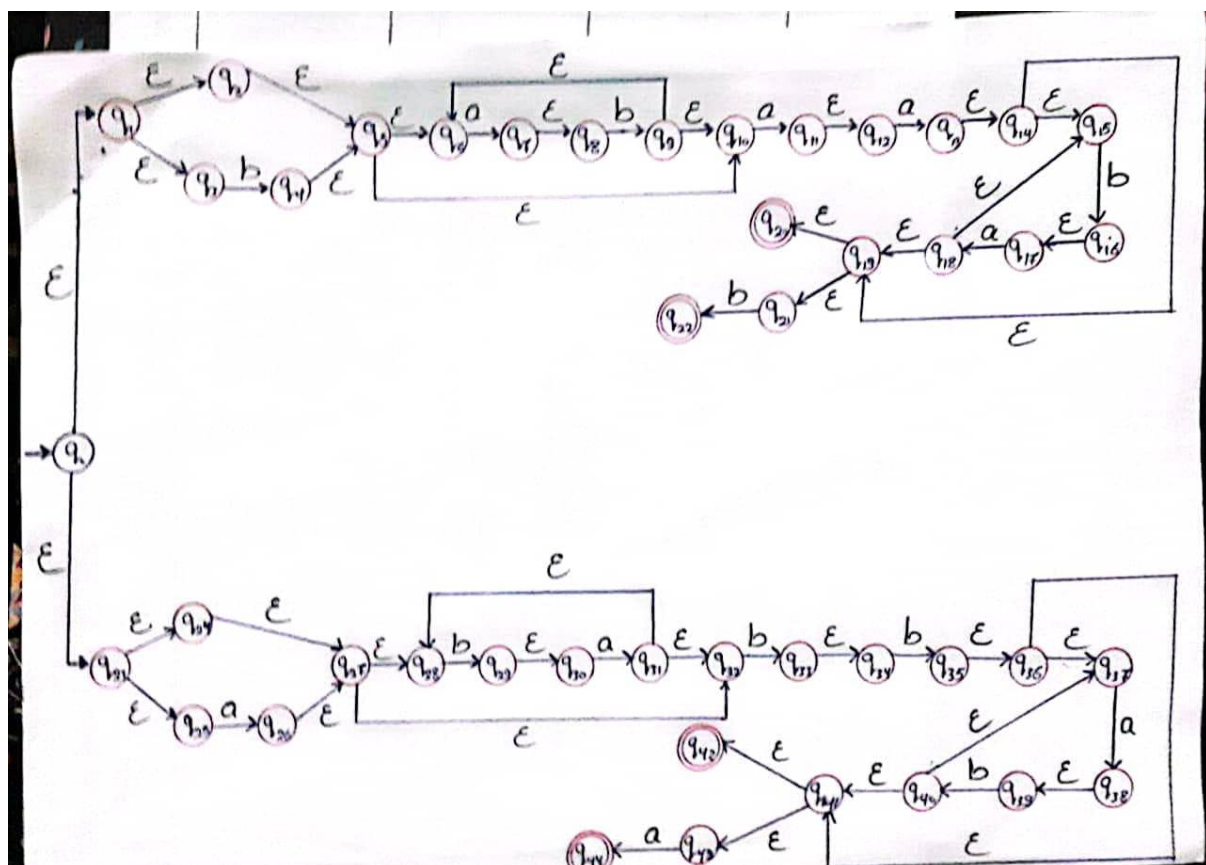
```

Output - RE (run) X
run:
Enter Regular Expression for First case
aa
true
Enter Regular Expression for Second case
bb
true
BUILD SUCCESSFUL (total time: 2 minutes 6 seconds)

Output - RE (run) X
run:
Enter Regular Expression for First case
bbbbbbbaa
false
Enter Regular Expression for Second case
aaaaaaaaabbb
false
BUILD SUCCESSFUL (total time: 13 seconds)

```

NFA:



Implementation:

```

13  */
14  public class NFAFINAL {
15      static int nfa = 0;
16      static char Epsilon = 'e';
17
18  static void q0(String x, int i) {
19      nfa = -1;
20      try {
21          if (x.charAt(i) == 'e' ) {
22              q1(x, i + 1);
23              q23(x, i + 1);
24          } else {
25              q0(x, i + 1);
26          }
27      } catch (StringIndexOutOfBoundsException e) {
28      }
29  }
30
31  static void q1(String x, int i) {
32      nfa = -1;
33      try {
34          if (x.charAt(i) == 'e' ) {
35              q2(x, i + 1);
36              q3(x, i + 1);
37          } else {

```

```
Source History
36         q3(x, i + 1);
37     } else {
38         q1(x, i + 1);
39     }
40 } catch (StringIndexOutOfBoundsException e) {
41 }
42 }
43
44 static void q2(String x, int i) {
45     nfa = -1;
46     try {
47         if (x.charAt(i) == 'e') {
48             q5(x, i + 1);
49         } else {
50             q2(x, i + 1);
51         }
52     } catch (StringIndexOutOfBoundsException e) {
53     }
54 }
55
56 static void q3(String x, int i) {
57     nfa = -1;
58     try {
59         if (x.charAt(i) == 'b') {
60             q4(x, i + 1);
```

```
Source History
60         q4(x, i + 1);
61     } else {
62         q3(x, i + 1);
63     }
64 } catch (StringIndexOutOfBoundsException e) {
65 }
66 }
67
68 static void q4(String x, int i) {
69     nfa = -1;
70     try {
71         if (x.charAt(i) == 'e') {
72             q5(x, i + 1);
73         } else {
74             q4(x, i + 1);
75         }
76     } catch (StringIndexOutOfBoundsException e) {
77     }
78 }
79
80 static void q5(String x, int i) {
81     nfa = -1;
82     try {
83         if (x.charAt(i) == 'e') {
84             q6(x, i + 1);
```

```

84         q6(x, i + 1);
85         q10(x, i + 1);
86     } else {
87         q5(x, i + 1);
88     }
89 } catch (StringIndexOutOfBoundsException e) {
90 }
91 }
92
93 static void q6(String x, int i) {
94     nfa = -1;
95     try {
96         if (x.charAt(i) == 'a') {
97             q7(x, i + 1);
98         } else {
99             q6(x, i + 1);
100         }
101     } catch (StringIndexOutOfBoundsException e) {
102     }
103 }
104
105 static void q7(String x, int i) {
106     nfa = -1;
107     try {
108         if (x.charAt(i) == 'e') {

```

```

108             if (x.charAt(i) == 'e') {
109                 q8(x, i + 1);
110             } else {
111                 q7(x, i + 1);
112             }
113         } catch (StringIndexOutOfBoundsException e) {
114         }
115     }
116 }
117 static void q8(String x, int i) {
118     nfa = -1;
119     try {
120         if (x.charAt(i) == 'b') {
121             q9(x, i + 1);
122         } else {
123             q8(x, i + 1);
124         }
125     } catch (StringIndexOutOfBoundsException e) {
126     }
127 }
128
129 static void q9(String x, int i) {
130     nfa = -1;
131     try {
132         if (x.charAt(i) == 'e') {

```

```
Source History
131
132     try {
133         if (x.charAt(i) == 'e') {
134             q6(x, i + 1);
135             q10(x, i + 1);
136         } else {
137             q9(x, i + 1);
138         }
139     } catch (StringIndexOutOfBoundsException e) {
140     }
141
142 static void q10(String x, int i) {
143     nfa = -1;
144     try {
145         if (x.charAt(i) == 'a') {
146             q11(x, i + 1);
147         } else {
148             q10(x, i + 1);
149         }
150     } catch (StringIndexOutOfBoundsException e) {
151     }
152 }
153
154 static void q11(String x, int i) {
155     nfa = -1;
```

```
Source History
155     nfa = -1;
156     try {
157         if (x.charAt(i) == 'e') {
158             q12(x, i + 1);
159         } else {
160             q11(x, i + 1);
161         }
162     } catch (StringIndexOutOfBoundsException e) {
163     }
164 }
165
166 static void q12(String x, int i) {
167     nfa = -1;
168     try {
169         if (x.charAt(i) == 'a') {
170             q13(x, i + 1);
171         } else {
172             q12(x, i + 1);
173         }
174     } catch (StringIndexOutOfBoundsException e) {
175     }
176 }
177
178 static void q13(String x, int i) {
```

```
Source History
177
178 static void q13(String x, int i) {
179     nfa = -1;
180     try {
181         if (x.charAt(i) == 'e') {
182             q14(x, i + 1);
183         } else {
184             q13(x, i + 1);
185         }
186     } catch (StringIndexOutOfBoundsException e) {
187     }
188 }
189
190 static void q14(String x, int i) {
191     nfa = -1;
192     try {
193         if (x.charAt(i) == 'e') {
194             q15(x, i + 1);
195             q19(x, i + 1);
196         } else {
197             q14(x, i + 1);
198         }
199     } catch (StringIndexOutOfBoundsException e) {
200     }
201 }
```

```
202
203 static void q15(String x, int i) {
204     nfa = -1;
205     try {
206         if (x.charAt(i) == 'b') {
207             q16(x, i + 1);
208         } else {
209             q15(x, i + 1);
210         }
211     } catch (StringIndexOutOfBoundsException e) {
212     }
213 }
214
215 static void q16(String x, int i) {
216     nfa = -1;
217     try {
218         if (x.charAt(i) == 'e') {
219             q17(x, i + 1);
220         } else {
221             q16(x, i + 1);
222         }
223     } catch (StringIndexOutOfBoundsException e) {
224     }
225 }
```

```
Source History
224     }
225 }
226
227 static void q17(String x, int i) {
228     nfa = -1;
229     try {
230         if (x.charAt(i) == 'a') {
231             q18(x, i + 1);
232         } else {
233             q17(x, i + 1);
234         }
235     } catch (StringIndexOutOfBoundsException e) {
236     }
237 }
238
239 static void q18(String x, int i) {
240     nfa = -1;
241     try {
242         if (x.charAt(i) == 'e') {
243             q15(x, i + 1);
244             q19(x, i + 1);
245         } else {
246             q18(x, i + 1);
247         }
248     } catch (StringIndexOutOfBoundsException e) {
249     }
250 }
251
252 static void q19(String x, int i) {
253     nfa = -1;
254     try {
255         if (x.charAt(i) == 'e') {
256             q20(x, i + 1);
257             q21(x, i + 1);
258         } else {
259             q19(x, i + 1);
260         }
261     } catch (StringIndexOutOfBoundsException e) {
262     }
263 }
264
265 static void q20(String x, int i) {
266     nfa = 1;
267     try {
268         q12(x, i + 1);
269     } catch (StringIndexOutOfBoundsException e) {
270     }
271 }
```

```
247     }
248     } catch (StringIndexOutOfBoundsException e) {
249     }
250 }
251
252 static void q19(String x, int i) {
253     nfa = -1;
254     try {
255         if (x.charAt(i) == 'e') {
256             q20(x, i + 1);
257             q21(x, i + 1);
258         } else {
259             q19(x, i + 1);
260         }
261     } catch (StringIndexOutOfBoundsException e) {
262     }
263 }
264
265 static void q20(String x, int i) {
266     nfa = 1;
267     try {
268         q12(x, i + 1);
269     } catch (StringIndexOutOfBoundsException e) {
270     }
271 }
```



```
270     }
271 }
272
273 static void q21(String x, int i) {
274     nfa = -1;
275     try {
276         if (x.charAt(i) == 'b') {
277             q22(x, i + 1);
278         } else {
279             q21(x, i + 1);
280         }
281     } catch (StringIndexOutOfBoundsException e) {
282     }
283 }
284
285 static void q22(String x, int i) {
286     nfa = 1;
287     try {
288         q22(x, i + 1);
289     } catch (StringIndexOutOfBoundsException e) {
290     }
291 }
292
293 static void q23(String x, int i) {
294     nfa = -1;
```

```

293 static void q23(String x, int i) {
294     nfa = -1;
295     try {
296         if (x.charAt(i) == 'e') {
297             q24(x, i + 1);
298             q25(x, i + 1);
299         } else {
300             q23(x, i + 1);
301         }
302     } catch (StringIndexOutOfBoundsException e) {
303     }
304 }
305
306 static void q24(String x, int i) {
307     nfa = -1;
308     try {
309         if (x.charAt(i) == 'e') {
310             q27(x, i + 1);
311         } else {
312             q24(x, i + 1);
313         }
314     } catch (StringIndexOutOfBoundsException e) {
315     }
316 }

```

```

315 }
316 }
317
318 static void q25(String x, int i) {
319     nfa = -1;
320     try {
321         if (x.charAt(i) == 'a') {
322             q26(x, i + 1);
323         } else {
324             q25(x, i + 1);
325         }
326     } catch (StringIndexOutOfBoundsException e) {
327     }
328 }
329
330 static void q26(String x, int i) {
331     nfa = -1;
332     try {
333         if (x.charAt(i) == 'e') {
334             q27(x, i + 1);
335         } else {
336             q26(x, i + 1);
337         }
338     } catch (StringIndexOutOfBoundsException e) {
339     }
340 }
341
342 static void q27(String x, int i) {
343     nfa = -1;
344     try {
345         if (x.charAt(i) == 'e') {
346             q28(x, i + 1);
347             q32(x, i + 1);
348         } else {
349             q27(x, i + 1);
350         }
351     } catch (StringIndexOutOfBoundsException e) {
352     }
353 }
354
355 static void q28(String x, int i) {
356     nfa = -1;
357     try {
358         if (x.charAt(i) == 'b') {
359             q29(x, i + 1);
360         } else {
361             q28(x, i + 1);
362         }
363     } catch (StringIndexOutOfBoundsException e) {
364     }
365 }

```

```

337 }
338 } catch (StringIndexOutOfBoundsException e) {
339 }
340 }
341
342 static void q27(String x, int i) {
343     nfa = -1;
344     try {
345         if (x.charAt(i) == 'e') {
346             q28(x, i + 1);
347             q32(x, i + 1);
348         } else {
349             q27(x, i + 1);
350         }
351     } catch (StringIndexOutOfBoundsException e) {
352     }
353 }
354
355 static void q28(String x, int i) {
356     nfa = -1;
357     try {
358         if (x.charAt(i) == 'b') {
359             q29(x, i + 1);
360         } else {
361             q28(x, i + 1);
362         }
363     } catch (StringIndexOutOfBoundsException e) {
364     }
365 }

```

```
Source History
358         if (x.charAt(i) == 'b') {
359             q29(x, i + 1);
360         } else {
361             q28(x, i + 1);
362         }
363     } catch (StringIndexOutOfBoundsException e) {
364     }
365 }
366
367 static void q29(String x, int i) {
368     nfa = -1;
369     try {
370         if (x.charAt(i) == 'e') {
371             q30(x, i + 1);
372         } else {
373             q29(x, i + 1);
374         }
375     } catch (StringIndexOutOfBoundsException e) {
376     }
377 }
378
379 static void q30(String x, int i) {
380     nfa = -1;
381     try {
382         if (x.charAt(i) == 'a') {
383             q31(x, i + 1);
384         } else {
385             q30(x, i + 1);
386         }
387     } catch (StringIndexOutOfBoundsException e) {
388     }
389 }
390
391 static void q31(String x, int i) {
392     nfa = -1;
393     try {
394         if (x.charAt(i) == 'e') {
395             q28(x, i + 1);
396             q32(x, i + 1);
397         } else {
398             q31(x, i + 1);
399         }
400     } catch (StringIndexOutOfBoundsException e) {
401     }
402 }
403
```

```
Source History
402 }
403
404 static void q32(String x, int i) {
405     nfa = -1;
406     try {
407         if (x.charAt(i) == 'b') {
408             q33(x, i + 1);
409         } else {
410             q32(x, i + 1);
411         }
412     } catch (StringIndexOutOfBoundsException e) {
413     }
414 }
415
416 static void q33(String x, int i) {
417     nfa = -1;
418     try {
419         if (x.charAt(i) == 'e') {
420             q34(x, i + 1);
421         } else {
422             q33(x, i + 1);
423         }
424     } catch (StringIndexOutOfBoundsException e) {
425     }
426 }
```

```
Source History
426 }
427
428 static void q34(String x, int i) {
429     nfa = -1;
430     try {
431         if (x.charAt(i) == 'b') {
432             q35(x, i + 1);
433         } else {
434             q34(x, i + 1);
435         }
436     } catch (StringIndexOutOfBoundsException e) {
437     }
438 }
439
440 static void q35(String x, int i) {
441     nfa = -1;
442     try {
443         if (x.charAt(i) == 'e') {
444             q36(x, i + 1);
445         } else {
446             q35(x, i + 1);
447         }
448     } catch (StringIndexOutOfBoundsException e) {
449     }
450 }
```

```
Source History
449 }
450
451 static void q36(String x, int i) {
452     nfa = -1;
453     try {
454         if (x.charAt(i) == 'e') {
455             q37(x, i + 1);
456             q41(x, i + 1);
457         } else {
458             q36(x, i + 1);
459         }
460     } catch (StringIndexOutOfBoundsException e) {
461     }
462 }
463
464 static void q37(String x, int i) {
465     nfa = -1;
466     try {
467         if (x.charAt(i) == 'a') {
468             q38(x, i + 1);
469         } else {
470             q37(x, i + 1);
471         }
472     } catch (StringIndexOutOfBoundsException e) {
473     }
```

```
471         q37(x, i + 1);
472     }
473     } catch (StringIndexOutOfBoundsException e) {
474     }
475 }
476
477 static void q38(String x, int i) {
478     nfa = -1;
479     try {
480         if (x.charAt(i) == 'e') {
481             q39(x, i + 1);
482         } else {
483             q38(x, i + 1);
484         }
485     } catch (StringIndexOutOfBoundsException e) {
486     }
487 }
488
489 static void q39(String x, int i) {
490     nfa = -1;
491     try {
492         if (x.charAt(i) == 'b') {
493             q40(x, i + 1);
494         } else {
495             q39(x, i + 1);
496         }
497     } catch (StringIndexOutOfBoundsException e) {
498     }
499 }
500
501 static void q40(String x, int i) {
502     nfa = -1;
503     try {
504         if (x.charAt(i) == 'e') {
505             q37(x, i + 1);
506             q41(x, i + 1);
507         } else {
508             q40(x, i + 1);
509         }
510     } catch (StringIndexOutOfBoundsException e) {
511     }
512 }
513
514 static void q41(String x, int i) {
515     nfa = -1;
516     try {
517         if (x.charAt(i) == 'e') {
518             q37(x, i + 1);
519             q41(x, i + 1);
520         } else {
521             q41(x, i + 1);
522         }
523     } catch (StringIndexOutOfBoundsException e) {
524     }
525 }
```

```

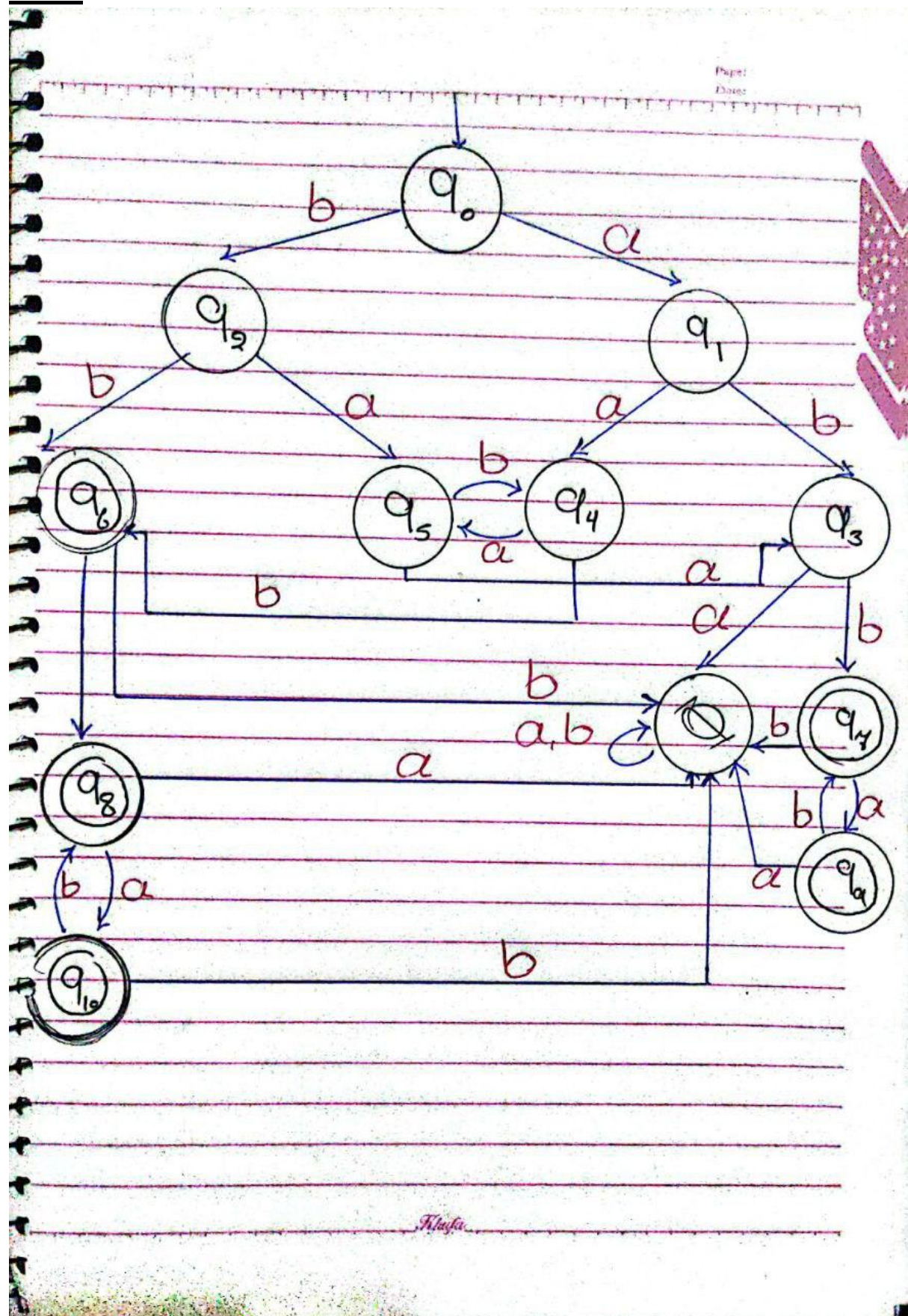
514 static void q41(String x, int i) {
515     nfa = -1;
516     try {
517         if (x.charAt(i) == 'e') {
518             q42(x, i + 1);
519             q43(x, i + 1);
520         } else {
521             q41(x, i + 1);
522         }
523     } catch (StringIndexOutOfBoundsException e) {
524     }
525 }
526
527 static void q42(String x, int i) {
528     nfa = 1;
529     try {
530         q42(x, i + 1);
531     } catch (StringIndexOutOfBoundsException e) {
532     }
533 }
534
535 static void q43(String x, int i) {
536     nfa = -1;
537     try {
538         if (x.charAt(i) == 'a') {
539             q44(x, i + 1);
540         } else {
541             q43(x, i + 1);
542         }
543     } catch (StringIndexOutOfBoundsException e) {
544     }
545 }
546
547 static void q44(String x, int i) {
548     nfa = 1;
549     try {
550         q44(x, i + 1);
551     } catch (StringIndexOutOfBoundsException e) {
552     }
553 }
554
555 public static void main(String[] args) {
556     Scanner sc = new Scanner(System.in);
557     System.out.println("Enter String: ");
558     String s = sc.next();
559     q0(s, 0);
560     if (nfa > 0) {
561         try {
562             q44(x, i + 1);
563         } catch (StringIndexOutOfBoundsException e) {
564         }
565     }
566 }
567
568 public static void main(String[] args) {
569     Scanner sc = new Scanner(System.in);
570     System.out.println("Enter String: ");
571     String s = sc.next();
572     q0(s, 0);
573     if (nfa > 0) {
574         System.out.println("Accepted");
575     } else {
576         System.out.println("Not Accepted");
577     }
578 }
579

```

```
Output - NFAFINAL (run) X
run:
Enter String:
abaaba
Accepted
BUILD SUCCESSFUL (total time: 16 seconds)

Output - NFAFINAL (run) ^
run:
Enter String:
baabb
not Accepted
BUILD SUCCESSFUL (total time: 3 seconds)
|
```

DFA



Implementation:

```
9 public class FINALDFA {
15
16     static int dfa = 0;
17
18     static void q0(String x, int i) {
19         dfa = -1;
20         try {
21             if (x.charAt(i) == 'a') {
22                 q1(x, i + 1);
23             } else if (x.charAt(i) == 'b') {
24                 q2(x, i + 1);
25             }
26
27         } catch (StringIndexOutOfBoundsException e) {
28
29         }
30
31     }
32
33     static void q1(String x, int i) {
34         dfa = -1;
35         try {
36             if (x.charAt(i) == 'a') {
37                 q4(x, i + 1);
38             } else if (x.charAt(i) == 'b') {
39                 q3(x, i + 1);
40             }
41
42         } catch (StringIndexOutOfBoundsException e) {
43
44         }
45
46     }
47
48     static void q2(String x, int i) {
49         dfa = -1;
50         try {
51             if (x.charAt(i) == 'a') {
52                 q5(x, i + 1);
53             } else if (x.charAt(i) == 'b') {
54                 q6(x, i + 1);
55             }
56
57         } catch (StringIndexOutOfBoundsException e) {
58
59         }
60
61     }
```

```

61 static void q3(String x, int i) {
62     dfa = 1;
63     try {
64         if (x.charAt(i) == 'a') {
65             Q(x, i + 1);
66         } else if (x.charAt(i) == 'b') {
67             q7(x, i + 1);
68         }
69     } catch (StringIndexOutOfBoundsException e) {
70     }
71 }
72
73
74
75 static void q4(String x, int i) {
76     dfa = -1;
77     try {
78         if (x.charAt(i) == 'a') {
79             q5(x, i + 1);
80         } else if (x.charAt(i) == 'b') {
81             q6(x, i + 1);
82         }
83     } catch (StringIndexOutOfBoundsException e) {
84     }
85 }

```

```

85
86
87
88
89 static void q5(String x, int i) {
90     dfa = -1;
91     try {
92         if (x.charAt(i) == 'a') {
93             q3(x, i + 1);
94         } else if (x.charAt(i) == 'b') {
95             q4(x, i + 1);
96         }
97     } catch (StringIndexOutOfBoundsException e) {
98     }
99 }
100
101
102
103 static void q6(String x, int i) {
104     dfa = 1;
105     try {
106         if (x.charAt(i) == 'a') {
107             q8(x, i + 1);
108         } else if (x.charAt(i) == 'b') {
109             Q(x, i + 1);
110         }
111     } catch (StringIndexOutOfBoundsException e) {
112     }
113 }
114
115
116
117 static void q7(String x, int i) {
118     dfa = 1;
119     try {
120         if (x.charAt(i) == 'a') {
121             q9(x, i + 1);
122         } else if (x.charAt(i) == 'b') {
123             Q(x, i + 1);
124         }
125     } catch (StringIndexOutOfBoundsException e) {
126     }
127 }
128
129
130
131 static void q8(String x, int i) {
132     dfa = 1;

```

```

131 static void q8(String x, int i) {
132     dfa = 1;
133     try {
134         if (x.charAt(i) == 'a') {
135             Q(x, i + 1);
136         } else if (x.charAt(i) == 'b') {
137             q10(x, i + 1);
138         }
139     } catch (StringIndexOutOfBoundsException e) {
140     }
141 }
142
143
144

```

```

145 static void q9(String x, int i) {
146     dfa = 1;
147     try {
148         if (x.charAt(i) == 'a') {
149             Q(x, i + 1);
150         } else if (x.charAt(i) == 'b') {
151             q7(x, i + 1);
152         }
153     } catch (StringIndexOutOfBoundsException e) {
154     }
155 }
156
157
158

```

```

154     } catch (StringIndexOutOfBoundsException e) {
155     }
156 }
157
158

```

```

159 static void q10(String x, int i) {
160     dfa = 1;
161     try {
162         if (x.charAt(i) == 'a') {
163             q8(x, i + 1);
164         } else if (x.charAt(i) == 'b') {
165             Q(x, i + 1);
166         }
167     } catch (StringIndexOutOfBoundsException e) {
168     }
169 }
170
171
172

```

```

173 static void Q(String x, int i) {
174     dfa = -1;
175     try {
176         if (x.charAt(i) == 'a') {
177             Q(x, i + 1);

```

```

177             Q(x, i + 1);
178         } else if (x.charAt(i) == 'b') {
179             Q(x, i + 1);
180         }
181     } catch (StringIndexOutOfBoundsException e) {
182     }
183 }
184
185

```

```

186 public static void main(String[] args) {
187     Scanner sc = new Scanner(System.in);
188     System.out.println("Enter String: ");
189     String s = sc.next();
190     q0(s, 0);
191     if (dfa > 0) {
192         System.out.println("Accepted");
193     } else {
194         System.out.println("Not Accepted");
195     }
196 }
197
198
199
200

```

```
Output - FINALNFA (run) x
run:
Enter String:
baaba
Accepted
BUILD SUCCESSFUL (total time: 11 seconds)
|

Output - FINALNFA (run) x
run:
Enter String:
baaaba
Not Accepted
BUILD SUCCESSFUL (total time: 3 seconds)
```

CFG:

$$S \rightarrow S \mid S'$$
$$S \rightarrow abS \mid aA \mid b \mid \epsilon$$
$$A \rightarrow aB$$
$$B \rightarrow baB \mid b \mid \epsilon$$
$$S' \rightarrow baS' \mid bC \mid a \mid \epsilon$$
$$C \rightarrow bD$$
$$D \rightarrow abD \mid a \mid \epsilon$$

Implementation:

```
15 public class FinallycontextFREEG {
16
17     /**
18      * @param args the command line arguments
19      */
20     static int np = 0;
21     //Insert grammar here
22     static String grammar[][] = {{"S", "S", "S'"}, {"S", "abS", "aA", "b", ""},
23     {"A", "aB"}, {"B", "baB", "b", ""}, {"S'", "baS'", "bC", "a", " "},
24     {"C", "bD"}, {"D", "abd", "a", " "}};
25
26     //Checks if the passed string can be achieved for the grammar
27     static String check(String a) {
28
29         String to_ret = "";
30
31         int count = 0;
32         for (int i = 0; i < np; i++) {
33             for (count = 0; count < grammar[i].length; count++) {
34
35                 for (count = 0; count < grammar[i].length; count++) {
36                     if (grammar[i][count].equals(a)) {
37                         to_ret += grammar[i][0];
38                     }
39                 }
40             }
41             return to_ret;
42         }
43
44         //Makes all possible combinations out of the two string passed
45         static String combinat(String a, String b) {
46             String to_ret = "", temp = "";
47
48             for (int i = 0; i < a.length(); i++) {
49                 for (int j = 0; j < b.length(); j++) {
50                     temp = "";
51                     temp += a.charAt(i) + " " + b.charAt(j);
52                     to_ret += check(temp);
53                 }
54             }
55             return to_ret;
56         }
57
58         public static void main(String[] args) {
59             // TODO code application logic here
60             String start;
61             int n = 0;
62             Scanner in = new Scanner(System.in);
63             //Start symbol is generally "S"
64             start = "S";
65             //np = no of productions
66             np = grammar.length;
67             String temp;
68             System.out.println("Enter the string to be checked >>");
69             String str = in.nextLine(), st = "", r = "";
```

```

67     np = grammar.length;
68     String temp;
69     System.out.println("Enter the string to be checked >>");
70     String str = in.nextLine(), st = "", r = "";
71     int count;
72     String ans_mat[][] = new String[10][10];
73
74     //Fill the diagonl of the matrix (first iteration of algorithm)
75     for (int i = 0; i < str.length(); i++) {
76         r = "";
77         st = "" + str.charAt(i);
78         for (int j = 0; j < np; j++) {
79             for (count = 1; count < grammar[j].length; count++) {
80                 if (grammar[j][count].equals(st)) {
81                     r += grammar[j][0];
82                 }
83             }
84         }
85         ans_mat[i][i] = r;

```

```

84     }
85     ans_mat[i][i] = r;
86 }
87
88 //Fill the rest of the matrix
89 for (int i = 1; i < str.length(); i++) {
90     for (int j = i; j < str.length(); j++) {
91         r = "";
92         for (int k = j - i; k < j; k++) {
93             r += combinat(ans_mat[j - i][k], ans_mat[k + 1][j]);
94         }
95         ans_mat[j - i][j] = r;
96     }
97 }
98
99 //The last column of first row should have the start symbol
100 /* if (ans_mat[0][str.length() - 1].indexOf(start) >= 0) {
101     accept();
102 } else {

```

```

Source History
102     } else {
103         reject();*/
104     System.out.println(Pattern.matches("(ab)+[a]{2}(ba)+[b]", str)); //true
105     System.out.println(Pattern.matches("(ab)+[a]{2}(ba)+[b]", "ababaaabab")); //false
106     System.out.println(Pattern.matches("(ba)+[b]{2}(ab)+[a]?", str)); //true
107
108 }
109
110 }
111
112 /* public static void accept() {
113     System.out.println("String is accepted");
114     System.exit(0);
115 }
116
117 public static void reject() {
118     System.out.println("String is rejected");
119     System.exit(0);
120 }*/

```

```
Output - finallycontextFREEG (run) x
FUN:
Enter the string to be checked >>
abaabab
true
false
false
BUILD SUCCESSFUL (total time: 7 seconds)
```

```
put - finallycontextFREEG (run) x
FUN:
Enter the string to be checked >>
bababbaba
false
false
true
BUILD SUCCESSFUL (total time: 20 seconds)
```