



UIL Computer Science Competition

Invitational B 2020

JUDGES PACKET - CONFIDENTIAL

I. Instructions

1. The attached printouts of the judge test data are provided for the reference of the contest director and programming judges. Additional copies may be made if needed for this purpose.
2. This packet must remain CONFIDENTIAL. Additional copies may be made and returned to schools when other confidential contest material is returned.

II. Table of Contents

Number	Name
Problem 1	Aran
Problem 2	Brianna
Problem 3	Deepa
Problem 4	Emilio
Problem 5	Guoliang
Problem 6	Irene
Problem 7	Josey
Problem 8	Kelly
Problem 9	Maciej
Problem 10	Nitin
Problem 11	Teresa
Problem 12	Willie

Problem #1
60 Points

1. Aran

Program Name: Aran.java

Input File: None

Test Output To Screen:

Day 1: Aa
Day 2: AaBb
Day 3: AaBbCc
Day 4: AaBbCcDd
Day 5: AaBbCcDdEe
Day 6: AaBbCcDdEeFf
Day 7: AaBbCcDdEeFfGg
Day 8: AaBbCcDdEeFfGgHh
Day 9: AaBbCcDdEeFfGgHhIi
Day 10: AaBbCcDdEeFfGgHhIiJj
Day 11: AaBbCcDdEeFfGgHhIiJjKk
Day 12: AaBbCcDdEeFfGgHhIiJjKkLl
Day 13: AaBbCcDdEeFfGgHhIiJjKkLlMm
Day 14: AaBbCcDdEeFfGgHhIiJjKkLlMmNn
Day 15: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOo
Day 16: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPp
Day 17: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQq
Day 18: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRr
Day 19: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSs
Day 20: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTt
Day 21: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUu
Day 22: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVv
Day 23: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWw
Day 24: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXx
Day 25: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYy
Day 26: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

Problem #2
60 Points

2. Brianna

Program Name: Brianna.java

Input File: brianna.dat

Test Input File:

92 29 97 71 38 73 62 64 74 21 77 45 42 81 9 44 57 55 23 1 40 54 36 4 91 59 41
95 47 16 33 6 72 27 14 63

Test Output To Screen:

1 97 96

Problem #3
60 Points

3. Deepa

Program Name: Deepa.java

Input File: deepa.dat

Test Input File:

```
5
3 40000
0 10
9525 12
38700 22
4 9823
0 10
9525 12
38700 22
82500 24
7 1000000
0 10
9525 12
38700 22
82500 24
157500 32
200000 35
500000 37
20 14285
0 10
500 12
1000 14
1500 16
2000 18
2500 20
3000 22
3500 24
4000 26
4500 28
5000 30
5500 32
6000 34
6500 36
7000 38
7500 40
8000 42
8500 44
9000 46
9500 48
1 1
0 1
```

Test Output To Screen:

```
Case #1: $4,739.50 11.849%
Case #2: $988.26 10.061%
Case #3: $335,689.50 33.569%
Case #4: $4,956.80 34.699%
Case #5: $0.01 1.000%
```

Problem #4
60 Points

4. Emilio

Program Name: Emilio.java

Input File: emilio.dat

Test Input File:

```

5
25
5.6 3.3 3.5 6.2 4.1 6.3 6.4
6.1 7.6 4.4 3.0 7.8 4.2
4.6 3.4 8.5 4.2 6.6 9.5 9.7 9.3 3.5 7.2
5.5 3.5
35
8.5
5.9 8.5 9.9 9.8 8.3 7.2 6.9
3.8 9.3 6.2 7.7
4.2 7.1 3.4 7.2 4.5 3.2 5.9
5.2 6.8 7.7
9.6 5.5 8.1 10.0
6.2 3.7 5.2 5.8 3.7 4.3 9.1 9.1
3.9
27
7.7 6.2 9.3 9.7 8.5 8.2 8.2 3.4
3.3 9.8 9.2 6.8 3.1 9.7 8.6 7.5
6.1 8.5 5.6 6.1 9.9 5.7 3.7 5.2
4.5 10.0
8.3
20
8.2 8.1
5.4 6.7 4.1 7.3 6.2 6.5 6.9
7.6 5.8 3.7 8.5 3.3
5.9
8.2 9.9 3.2 3.9 4.7
100
5.7
9.9 7.5
8.9 7.0 3.6
5.4 7.3 5.6 6.4 5.5 9.9 10.0 5.8 3.8 7.4
6.7 3.6 3.0 3.6 7.3 6.5 6.3 5.5 5.0 7.3 3.5 3.5 9.6 4.6 4.6 3.9 8.6 6.0 8.6 3.5 5.1
5.8
3.1 8.1 8.6 5.4 6.1 9.4 6.6 9.7 8.1 7.1 8.1 4.8 4.2 6.3 3.4 6.4 6.1 4.6 8.0 4.8 4.4
7.1
9.4 7.9 8.7 8.6 6.1 7.8 5.2 6.1 3.8 7.2 8.5 9.6 5.3 3.6 9.8 4.8 5.9 3.7 8.7 8.1 9.8
9.0
3.9 8.6 3.9 7.3 8.1 8.3 9.2 9.4 7.1 4.9 8.5 8.7 9.5 8.2 8.2
4.4 7.3 9.8

```

Test Output To Screen:

```

1: 7.9
1: 3.7
-----
2: 9.2
2: 4.0
-----
3: 9.3
3: 4.7
-----
4: 7.8
4: 4.6
-----
5: 9.8
5: 3.4
-----

```

Problem #5
60 Points

5. Guoliang

Program Name: Guoliang.java

Input File: guoliang.dat

Test Input File:

```
position tickle tub truculent
erratic chance
equal string bathe
butter need
irate
go cooing repeat look stick
pleasant insurance glorious
lively fabulous grandiose
form kind campgrounds sparkle
tense beef face team salt abiding
bottle flower
typical stiff request
```

Test Output To Screen:

```
tickle
tub
equal
string
bathe
irate
go
stick
grandiose
form
kind
campgrounds
sparkle
face
team
salt
flower
typical
```

Problem #6
60 Points

6. Irene

Program Name: Irene.java

Input File: irene.dat

Test Input File:

```
22
15
16
49
35
72
55
65
700
121
1397
259081
20000
2
74
573049
82
1690499
10098
27473
11
2147483646
1334367841
```

Test Output To Screen:

```
15 DISCRETE SEMIPRIME
16 NOT SEMIPRIME
49 SEMIPRIME
35 DISCRETE SEMIPRIME
72 NOT SEMIPRIME
55 DISCRETE SEMIPRIME
65 DISCRETE SEMIPRIME
700 NOT SEMIPRIME
121 SEMIPRIME
1397 DISCRETE SEMIPRIME
259081 SEMIPRIME
20000 NOT SEMIPRIME
2 NOT SEMIPRIME
74 DISCRETE SEMIPRIME
573049 SEMIPRIME
82 DISCRETE SEMIPRIME
1690499 DISCRETE SEMIPRIME
10098 NOT SEMIPRIME
27473 DISCRETE SEMIPRIME
11 NOT SEMIPRIME
2147483646 NOT SEMIPRIME
1334367841 SEMIPRIME
```

Problem #7

60 Points

7. Josey

Program Name: Josey.java

Input File: josey.dat

Test Input File:

```
8
13
LIECTENSTEIN SWITZERLAND
SWITZERLAND AUSTRIA
AUSTRIA LIECTENSTEIN
ITALY SWITZERLAND
SWITZERLAND FRANCE
FRANCE ITALY
ITALY OCEAN
OCEAN FRANCE
AUSTRIA SLOVENIA
SLOVENIA OCEAN
GERMANY AUSTRIA
GERMANY SWITZERLAND
OCEAN GERMANY
7
OCEAN ONE
ONE TWO
TWO THREE
THREE FOUR
FOUR FIVE
FIVE SIX
SIX OCEAN
13
VALE CROWNLANDS
WESTERLAND REACH
REACH STORMLANDS
OCEAN DORNE
STORMLANDS DORNE
RIVERLANDS WESTERLAND
RIVERLANDS NORTH
OCEAN VALE
OCEAN CROWNLANDS
WESTERLAND OCEAN
CROWNLANDS RIVERLANDS
RIVERLANDS OCEAN
REACH OCEAN
7
EARTH WIND
WATER FIRE
FIRE WIND
OCEAN WATER
FIRE EARTH
WATER EARTH
WIND WATER
12
UWOIDXTMBNHLGKY EWBRFYVSLXHCPZO
PATXESCLNRDZKOH POFUTSKRCGEAWVX
PWAYIKQGSLOCHXF OCEAN
JOSHFDZWGMNCAIT PATXESCLNRDZKOH
HUFDAYCSXNTGOVL IBTOLSZQXGWYEMC
JOSHFDZWGMNCAIT PWAYIKQGSLOCHXF
HUFDAYCSXNTGOVL ECBDGIVNLQPMWYF
JOSHFDZWGMNCAIT OCEAN
UWOIDXTMBNHLGKY POFUTSKRCGEAWVX
IBTOLSZQXGWYEMC DOXNAZGCLWBERTS
EWBRFYVSLXHCPZO ECBDGIVNLQPMWYF
OCEAN DOXNAZGCLWBERTS
102
JIBGFHNZTMSXWPL OYRPDJULQHWTVCI
MBKFWQOGJINSC TH FPVAJNEHMSYUTLO
TSZMOFEHPYINXJL KLSMIBWEHJTGFU
LOBMPESNRAVZDWI FTOMNUKCDRPLYBI
```

```
AREWUTJVKIYOFHL ANWQEDSRKVOMLIG
VDRJKOYLXQENSWG CGODWUVBTRNHLMF
LCDOEHFXAVQRIGS RNYDXLUPEIBCQVM
CSTIQPD LNFZBGHY BFQLIJVTNAZWYKE
OWUHMDXLCYTSEAN TSZMOFEHPYINXJL
RUM LWOSJTYKQEBI JSXELHANVWROTDY
WNIAXPOVUZEHYQF TFMRKLVBCNGQOHX
IBSQJVFCPOAHWTE PVFWZJMGKCILROX
SFUHMVITKWJLCZP SJWGRQUNLKEZHXF
JSXELHANVWROTDY YNAFKBLPRWQZXVD
VJHLFUKTDIMZAYB MSEQB VWOZIUFND
AYIULJDOWPKBRCH EHJKPVMXWUZDNQO
JBYGPRENMZWCVD GKBVIQFYLRNHCZE
WFTQAI OCEZSUNKV RLJFWTAIGZVSDYK
LRJGCKUHXENMDTA HYDGRFILEOXWUTP
REPTJQABNMYKSZV GWSBAKFMOVZTINR
EMLXQKJZPONYUSB OCEAN
JNMZXSBDKAEHLFR NTYSFOKDLIEVCRM
YJWIOGEHLXSAQFV FVRBLGZCIWYMHXE
(see file for remainder of input – 2180 lines)
```

Test Output To Screen:

```
Case #1: 1
LIECTENSTEIN
Case #2: 2
FOUR
THREE
Case #3: 0
Case #4: 0
Case #5: 5
ECBDGIVNLQPMWYF
EWBRFYVSLXHCPZO
HUFDAYCSXNTGOVL
POFUTSKRCGEAWVX
UWOIDXTMBNHLGKY
Case #6: 91
ANWQEDSRKVOMLIG
AREWUTJVKIYOFHL
AXFPCIUVHWJZKDT
AYIULJDOWPKBRCH
BAOWXFDNIEUMJVR
BFQLIJVTNAZWYKE
BNOVJLCMSKIFRWH
BWUHYAFNXOKPRTE
BZRG SUELFVHIXOK
CGODWUVBTRNHLMF
CSTIQPD LNFZBGHY
DBHXXJPVUGZTQIR
DCQANEVRXKIGYJU
DEHYBIMZNVSCXU
EHJKPVMXWUZDNQO
EHMBTGVND F CJRLA
FPVAJNEHMSYUTLO
FTOMNUKCDRPLYBI
FVCQD TKEUSNHZXY
FVRBLGZCIWYMHXE
GKBVIQFYLRNHCZE
GQOIFVNHJSJKMTRX
GWSBAKFMOVZTINR
GYBAJLKVNESWORM
GYVQLUFHSXCPAZJ
HGUCVSDYLQFIXZJ
HYDGRFILEOXWUTP
(see file for remainder of output – 2053 lines)
```


Problem #8
60 Points

8. Kelly

Program Name: Kelly.java

Input File: kelly.dat

Test Input File: (the first ten digits are bolded for the last four data sets listed below)

```

4 5 2 4 3 1 0 0 4 2 3 4 1 3 1 3 4 3
4 4 1 1 3 5 0 0 1 3 3 4 1 4 1 5 0 0 5 2 3 4 4 2 1 3 4 3
3 5 4 2 3 4 2 2 4 2 1 1 3 2 3 2 2 4 3 3 2 2 0 0 2 4 4 3 0 0 2 1 4 5 3 3
3 2 2 4 4 3 4 3 2 4 3 3 2 2 0 0 3 1 1 5 4 4 4 4 1 5 4 2 4 3 0 0 2 1 3 4 3 1 3
1 3 4 5 2
1 2
2 3
2 5
4 1
5 1
5 3
5 4
5 5
4 4 2 3 2 4 4 3 0 0 2 4 4 3 0 0 1 3 4 2 1 1 5 5 5 4
4 4 1 5 5 3 1 1 4 3 0 0 2 3 2 4 2 2 2 3 0 0 4 3 1 3 2 3 3 4 3 4 3 1 0 0 4 5 2
4 3 1 0 0 3 5 4 2 3 4 2 2 4 2 1 1 3 2 3 2 1 5 4 2 4 3 0 0 4 2 4 5 3 1 1 5 0 0
3 4 4 5 4 2 0 0 5 2 3 4 4 2 3 1 1 4
2 5 1 1 5 1 1 1 0 0 4 1 4 5 2 4 1 3 1 3 3 1 5 4 0 0 1 2 1 5 1 3 1 1 3 2 1 5 0
0 3 5 3 4 3 5 4 5 3 1 1 1 4 2 0 0 4 3 3 4 3 4 3 3 0 0 1 1 2 1 4 4 1 5 4 2 0 0
2 4 4 4 0 0 5 2 1 1 4 3 0 0 2 4 3 3 4 4 4 2 3 4 1 4 4 5 1 3 1 5 1 4
1 1 1 4 1 1 0 0 3 1 3 4 5 1 1 5 3 1 1 1 1 3 1 5 0 0 5 2 1 1 4 3 0 0 5 2 3 4 4
2 3 1 1 4 4 3 0 0 2 1 2 4 4 2 4 3 4 4 0 0 3 5 4 2 3 4 2 2 4 2 1 1 3 2 3 2 1 5
4 2 0 0 5 2 4 2 2 4 4 4 2 4 3 3 2 2 0 0 2 1 2 4 4 2 4 3 4 4 0 0 1 1 3 1 2 2 3
4 4 2 2 4 4 4 2 3 3 2
2 1 3 4 4 2 0 0 1 1 0 0 3 2 1 1 1 3 2 3 2 4 3 3 1 5 0 0 5 2 2 4 4 4 2 3 0 0 1
3 2 3 1 1 4 2 3 1 1 5 4 3 0 0 1 2 1 1 1 2 1 2 1 1 2 2 1 5

```

Test Output To Screen:

```

UIL ROCCS
TAP CODE WORCS
PROGRAMMING IS FUN
MISSING LETTERS FOLLOW
B
H
J
Q
V
X
Y
Z
THIS IS CRAZY
TEXAS HIGH SCHOOL UIL PROGRAMMERS RULE OUR WORLD
JAVA QUICCLY BECAME POPULAR SOON AFTER IT WAS INTRODUCED
ADA LOVELACE WAS WORLDS FIRST PROGRAMMER WRITING FIRST ALGORITHM
FOR A MACHINE WITH CHARLES BABBAGE

```

Problem #9
60 Points

9. Maciej

Program Name: maciej.java

Input File: maciej.dat

Test Input File:

```
10
#000000 #FFFFFF
#BF5700 #430030
#00FF00 #FF00FF
#123456 #EFDCEB
#000000 #000000
#000000 #000001
#FFFFFF #000000
#9AC13F #840C3B
#FFFFFF #FFFFFF
#010101 #101010
```

Test Output To Screen:

```
Case #1: #7F7F7F
Case #2: #812B18
Case #3: #7F7F7F
Case #4: #808888
Case #5: #000000
Case #6: #000000
Case #7: #7F7F7F
Case #8: #8F663D
Case #9: #FFFFFF
Case #10: #080808
```

Problem #10
60 Points

10. Nitin

Program Name: Nitin.java

Input File: nitin.dat

Test Input File:

15
Willie D
Ralph D
Rover D
Tiger C
Daisy D
Max C
Felix C
Lucy D
Bubba C
Bella C
Cooper D
Bailey D
Simba C
Luna D
Charlie D
18
Noah C

Tereza D
Bruce D
Olivia D
Terry C
Irene D
Joseph D
Estaban C
Rashi D
Emily C
Igor C
Jason C
Keith D
Tatiana D
Ambrocio D
Jose C
Niti C
Samantha D

Test Output To Screen:

Noah CAT Tiger
Tereza DOG Willie
Bruce DOG Ralph
Olivia DOG Rover
Terry CAT Max
Irene DOG Daisy
Joseph DOG Lucy
Estaban CAT Felix
Rashi DOG Cooper
Emily CAT Bubba
Igor CAT Bella
Jason CAT Simba
Keith DOG Bailey
Tatiana DOG Luna
Ambrocio DOG Charlie
Jose WAIT LIST CAT
Niti WAIT LIST CAT
Samantha WAIT LIST DOG

Problem #11
60 Points

11. Teresa

Program Name: Teresa.java

Input File: teresa.dat

Test Input File:

```

4
5 4
194 819 449 405
560 410 914 534
302 670 856 448
933 239 259 477
591 455 665 652
3 5
558      777      773      761      995
252      632      580      639      818
590      828      372      386      489
2 2
987      707
635      409
25 25
242      146      842      531      923      733      126      740      258      917      535      630      794
328      130      675      715      913      608      149      170      558      798      226      964
164      643      495      345      176      327      739      292      751      514      898      552      921
506      137      206      847      429      706      106      928      784      328      683      858
570      506      948      291      360      124      549      419      757      598      833      554      613
312      564      652      554      113      876      748      979      633      447      849      448
642      265      262      916      634      609      746      907      300      181      274      938      177
866      525      581      253      367      948      869      465      991      619      141      853
687      130      518      198      680      506      243      514      818      552      443      478      553
846      174      157      398      262      913      818      519      149      348      375      367
435      672      243      786      731      359      384      699      150      141      742      628      896
964      526      548      483      292      944      290      851      707      297      184      205
534      432      214      127      958      947      852      875      976      725      367      279      735
239      309      460      276      447      854      119      701      725      621      164      536
196      265      957      997      253      257      821      923      727      240      559      422      575
971      594      515      830      872      763      865      815      152      526      636      769
522      701      425      455      159      929      597      182      818      892      602      910      266
879      407      734      882      329      918      272      375      540      596      441      179
624      230      205      630      747      318      818      376      185      406      802      567      276
502      864      294      913      575      746      468      290      745      275      302      740
701      174      117      171      695      265      720      651      475      429      775      902      590
707      799      291      610      333      771      703      917      875      955      934      113
858      974      246      448      227      235      964      463      444      782      938      420      276
910      762      999      599      328      648      345      513      621      456      129      282
565      858      435      255      828      893      942      490      125      247      755      574      520
121      222      482      119      653      826      343      456      761      870      608      627
556      432      143      508      346      290      107      278      871      796      701      701      331
281      269      179      966      692      546      400      815      620      818      587      240
942      559      202      976      841      358      775      220      254      801      447      174      270
986      657      665      728      418      451      353      431      402      884      410      571
242      687      786      128      668      182      118      942      949      663      229      261      836
983      130      463      251      809      683      917      920      625      101      978      491
526      805      466      830      724      294      634      175      645      788      417      825      561
999      638      695      398      802      129      666      876      145      953      606      288
650      400      947      619      603      659      126      930      820      867      162      438      421
366      858      470      185      454      418      502      854      712      600      439      997
723      168      746      936      744      255      422      433      736      213      847      197      628
643      615      434      147      331      469      952      139      753      146      998      677
965      727      209      117      996      991      333      566      911      321      729      826      697
933      362      115      162      838      836      972      177      500      119      890      231
378      239      821      145      605      433      489      938      859      880      595      951      328
305      355      648      927      459      253      227      747      198      777      781      581
444      785      851      998      236      590      105      102      457      521      260      346      800
914      707      887      367      146      575      240      791      584      260      250      175
388      360      229      955      358      238      837      846      154      928      814      748      382
197      172      793      915      139      506      741      282      860      125      989      776
136      449      304      613      832      274      670      513      806      713      655      118      998
481      711      396      973      320      928      793      227      436      491      360      742
258      967      332      585      273      665      132      612      427      201      281      112      988
429      201      822      547      341      703      384      868      770      752      318      345

```

(Test Output next page)

UIL – Computer Science Programming Judge Packet – Invitational B 2020

Teresa - Test Output To Screen: (Note: each output line for the last data set below wraps around)

```

1:
194      239      259      405
302      410      448      449
455      477      560      665
534      591      670      856
652      819      914      933
+++++++
2:
252      372      386      489      632
558      580      639      777      818
590      761      773      828      995
+++++++
3:
409      635
707      987
+++++++
4:
101      102      105      106      112      113      113      115      117      117      119      121      124
      125      129      129      130      130      136      139      141      145      159      162      177
107      118      119      119      127      130      139      141      143      146      147      149      150
      157      164      170      175      175      176      181      182      197      229      253      266
118      125      128      137      146      149      152      154      162      164      168      174      177
      179      179      182      196      201      202      227      235      239      260      270      418
126      145      146      172      174      184      185      185      197      198      205      205      206
      213      220      227      227      236      238      242      262      274      276      281      451
126      171      174      209      226      230      231      240      240      242      251      253      254
      255      260      261      276      278      281      282      291      292      305      402      469
132      198      201      214      239      240      247      250      253      257      258      258      272
      273      275      279      290      292      294      312      320      328      367      436      506
222      229      243      255      265      265      276      282      290      297      300      302      321
      328      328      331      346      346      375      376      378      396      417      500      546
243      265      269      274      288      291      318      328      329      333      343      358      366
      367      382      388      406      419      420      427      431      434      443      540      575
246      290      309      331      345      345      345      348      360      360      360      384      398
      421      422      429      429      433      435      444      447      456      460      558      608
262      294      333      353      355      367      375      398      400      422      438      444      447
      448      457      463      463      465      481      514      520      535      584      603      648
304      318      341      362      384      410      432      448      455      456      470      475      478
      483      489      490      502      513      521      522      553      559      605      620      683
327      332      367      400      407      432      439      482      491      491      506      508      513
      519      526      547      549      552      552      561      595      621      634      645      703
358      418      425      441      449      468      514      515      525      526      531      534      536
      554      554      575      597      598      602      625      643      668      701      706      727
359      429      435      502      506      526      548      556      566      567      571      585      587
      590      596      599      633      634      655      663      680      707      736      746      747
433      447      466      559      564      565      574      581      581      600      606      610      612
      613      613      666      670      695      701      707      713      751      763      791      846
454      495      506      570      594      608      619      619      627      628      628      643      648
      651      703      712      715      720      724      725      729      757      771      815      866
459      518      590      615      621      624      630      630      636      652      672      677      697
      699      725      728      731      739      741      742      782      806      815      826      879
575      609      638      642      665      683      687      701      735      740      740      744      745
      746      746      748      752      755      786      788      818      830      836      851      910
650      653      657      659      675      701      742      747      748      753      775      775      777
      781      786      793      794      796      818      830      846      847      854      854      914
665      687      692      695      707      727      761      769      798      800      801      802      818
      818      820      821      825      828      849      868      875      876      882      916      933
701      711      733      734      770      776      785      802      814      818      821      826      832
      836      842      859      865      867      876      890      907      913      913      936      964
723      762      784      793      805      809      833      837      841      851      853      869      870
      871      880      893      896      902      915      917      918      923      934      955      971
799      822      838      847      852      858      858      858      860      884      892      910      911
      917      920      921      923      927      928      929      930      947      976      978      983
858      872      875      887      898      917      928      938      938      942      944      947
      948      949      952      953      958      964      966      967      986      988      989      997
864      913      928      938      942      948      951      955      957      964      965      972      973
      974      976      979      991      991      996      997      998      998      998      999      999
+++++++

```

Problem #12
60 Points

12. Willie

Program Name: Willie.java

Input File: willie.dat

Test Input File:

```
12
1
-----
| Q |
-----
*****
2
-----
|   |   |
-----
|   | Q |
-----
*****
2
-----
| Q |   |
-----
|   | Q |
-----
*****
3
-----
| Q |   |   |
-----
|   | Q |   |
-----
|   |   | Q |
-----
*****
4
-----
|   |   | Q |   |
-----
| Q |   |   |   |
-----
|   |   |   | Q |
-----
|   | Q |   |   |
-----
*****
4
-----
|   |   | Q |   |
-----
|   | Q |   |   |
-----
|   |   |   | Q |
-----
| Q |   |   |   |
-----
*****
```

(input continued next page)

(Willie – input file continued)

```
8
-----
|   |   |   | Q |   |   |   |   |
-----
|   | Q |   |   |   |   |   |   |
-----
|   |   |   |   |   |   |   | Q |
-----
|   |   | Q |   |   |   |   |   |
-----
|   |   |   |   |   |   | Q |   |
-----
|   |   |   |   |   |   |   | Q |
-----
|   |   |   |   | Q |   |   |   |
-----
| Q |   |   |   |   |   |   |   |
-----
```

8

```
-----
| Q |   |   |   |   |   |   |   |
-----
|   |   |   |   | Q |   |   |   |
-----
|   |   |   |   |   |   |   | Q |
-----
|   |   |   |   |   | Q |   |   |
-----
|   |   | Q |   |   |   |   |   |
-----
|   |   |   |   |   |   |   | Q |
-----
|   | Q |   |   |   |   |   |   |
-----
|   |   |   | Q |   |   |   |   |
-----
```

8

```
-----
|   |   |   | Q |   |   |   |   |
-----
| Q |   |   |   |   |   |   |   |
-----
|   |   |   |   |   |   |   | Q |
-----
|   | Q |   |   |   |   |   |   |
-----
|   |   |   |   |   | Q |   |   |
-----
|   |   |   |   |   |   |   | Q |
-----
|   |   |   |   | Q |   |   |   |
-----
|   |   |   |   |   |   |   |   |
-----
```

(input continued next page)

(Willie – input file continued)

8

			Q						
							Q		
									Q
							Q		
			Q						
									Q
			Q						
					Q				

8

							Q		
	Q								Q
							Q		
			Q						
									Q
			Q						
					Q				

10

									Q	
							Q			
			Q							
									Q	
							Q			
			Q							

(end of input file)

Willie - Test Output To Screen:

```
valid solution
incorrect attempt
incorrect attempt
incorrect attempt
valid solution
incorrect attempt
valid solution
valid solution
incorrect attempt
incorrect attempt
incorrect attempt
valid solution
```