



# UIL Computer Science Competition

## 2017 Invitational A

### Programming Judges Packet

#### Table of Contents

Number	Name
Problem 1	Abdul
Problem 2	Daniel
Problem 3	Geming
Problem 4	Htoo
Problem 5	Isabella
Problem 6	Jean
Problem 7	Kimberley
Problem 8	Luana
Problem 9	Maria
Problem 10	Patricia
Problem 11	Rohit
Problem 12	Tisha

**Problem #1**  
**60 Points**

**1. Abdul**

**Program Name: Abdul.java**

**Input File: abdul.dat**

**Test Input File: abdul.dat**

5  
11  
16  
3  
10  
99

**Test Output To Screen**

25  
121  
256  
9  
100  
9801

**Problem #2**  
**60 Points**

## 2. Daniel

**Program Name: Daniel.java**

**Input File: daniel.dat**

**Test Input File: daniel.dat**

Be so good they can't ignore you. - Steve Martin

**Ever tried? Ever failed? No matter. Try again. Fail again. Fail better. - Samuel Beckett**

Impossible is just a big word thrown around by small men who find it easier to live in the world they've been given than to explore the power they have to change it. Impossible is not a fact. It's an opinion. Impossible is not a declaration. It's a dare. Impossible is potential. Impossible is temporary. Impossible is nothing. - Muhammed Ali

**At the end of the day, let there be no excuses, no explanations, no regrets. - Steve Maraboli**

Every life form seems to strive to its maximum except human beings. How tall will a tree grow? As tall as it possibly can. Human beings, on the other hand, have been given the dignity of choice. You can choose to be all or you can choose to be less. Why not stretch up to the full measure of the challenge and see what all you can do? - Jim Rohn

**I AM THAT I AM - GOD**

**Test Output To Screen**

```
Be
so good
they can't ignore
you. - Steve Martin
*****
Ever
tried? Ever
failed? No matter.
Try again. Fail again.
Fail better. - Samuel Beckett
*****
Impossible
is just
a big word
thrown around by small
men who find it easier
to live in the world they've
been given than to explore the
power
they have
to change it.
Impossible is not a
fact. It's an opinion.
Impossible
is not
a declaration. It's
a dare. Impossible is
potential. Impossible is
temporary.
Impossible is
nothing. - Muhammed
Ali
*****
```

```
At
the end
of the day,
let there be no
excuses, no explanations, no
regrets.
- Steve
Maraboli
*****
Every
life form
seems to strive
to its maximum except
human beings. How tall will
a tree grow? As tall as
it possibly can. Human beings,
on
the other
hand, have been
given the dignity of
choice. You can choose to
be all or you can choose
to be less. Why not stretch up
to the full measure of the
challenge
and see
what all you
can do? - Jim
Rohn
*****
I
AM THAT
I AM -
GOD
*****
```

**Problem #3**  
**60 Points**

### 3. Geming

**Program Name:** Geming.java

**Input File:** geming.dat

**Test Input File:**

```
0
1
-11
25
-456
1000
-9999
10000
```

**Test Output To Screen**

```
0 -1
1 -2
-11 10
25 -26
-456 455
1000 -1001
-9999 9998
10000 -10001
```

**Problem #4**  
**60 Points**

## 4. Htoo

**Program Name: Htoo.java**

**Input File: htoo.dat**

**Test Input File: htoo.dat**

```
11
(0, 0) > (2, 2)
(2, 0) > (0, 2)
(0, 0) > (2, 2)
(0, 1) > (2, 3)
(0, 0) > (2, 2)
(4, 4) > (2, 2)
(0, 0) > (2, 0)
(1, 0) > (1, 1)
(2, 1) > (0, 1)
(0, 2) > (0, 0)
(-2, -1) > (-1, -1)
(0, 2) > (0, 0)
(0, 0) > (2, 2)
(2, 3) > (4, 2)
(0, 0) > (0, 100)
(4, 99) > (4, 112)
(0, 0) > (0, 100)
(0, 101) > (0, 112)
(0, 0) > (0, 100)
(0, 5) > (0, 12)
(0, 0) > (2, 2)
(4, 4) > (3, 3)
```

**Test Output To Screen**

```
INTERSECT
NO INTERSECTION
INTERSECT
INTERSECT
INTERSECT
NO INTERSECTION
NO INTERSECTION
NO INTERSECTION
NO INTERSECTION
INTERSECT
NO INTERSECTION
```

**Problem #5**  
**60 Points**

## 5. Isabella

**Program Name: Isabella.java**

**Input File: isabella.dat**

**Test Input File: isabella.dat**

```

12
ABC DEF 10
ABC GHI 0.000000000000005
ABC JKL 400000000
DEF ABC 0.1
DEF GHI 0.000000000000005
DEF JKL 40000000
GHI ABC 20000000000000
GHI DEF 20000000000000
GHI JKL 80000000000000000000
JKL ABC 0.0000000025
JKL DEF 0.0000000025
JKL GHI 0.0000000000000000000125
10
3
1000 ABC
6890 DEF
43.3 GHI
866000000016890.0 DEF
3
1000 ABC
6890 DEF
43.3 GHI
43.30000000084450 GHI
10
1000 ABC
1000 DEF
1000 JKL
1000 JKL
1000 JKL
1000 DEF
1000 ABC
1000 ABC
1000 ABC
10000.000 ABC
4
1000 ABC
1000 ABC
1000 ABC
1000 ABC
4000 ABC
8

```

```

1 ABC
1 DEF
1 GHI
1 JKL
200000.948 ABC
4720.4294 DEF
324.2 JKL
0.000003 GHI
32432 JKL
1
1.489 ABC
1.489 ABC
2
1.0 ABC
0.0 GHI
10.0 DEF
1
0.0 GHI
0.0 DEF
5
1234567890987654.976689 GHI
4567890876543456.374839 JKL
3456789098765.4758 JKL
2345678.738 ABC
4567890987654.3783 ABC
1234567890987657.26064138085114020277
731329875 GHI
3
1.000001 ABC
10.000002 DEF
5.73782 GHI
5799.0330159786 JKL
Test Output To Screen
GOOD LEDGER
GOOD LEDGER
BAD LEDGER
GOOD LEDGER
BAD LEDGER
GOOD LEDGER
GOOD LEDGER
GOOD LEDGER
GOOD LEDGER
GOOD LEDGER
BAD LEDGER

```

**Problem #6**  
**60 Points**

**6. Jean**

**Program Name: Jean.java**

**Input File: jean.dat**

**Test Input File: jean.dat**

```
10
3
A -> A B C
B -> B
C -> NONE
4
E -> E
F -> F
G -> G
D -> NONE
5
Q -> B C V
B -> B C V
C -> B C
D -> B C V
V -> NONE
1
A -> NONE
6
A -> A C E F
B -> A D E
C -> A B
D -> NONE
E -> A F
F -> NONE
1
A -> A
2
A -> NONE
B -> NONE
5
A -> A B C D
B -> NONE
```

```
C -> NONE
D -> NONE
E -> NONE
6
A -> B
B -> NONE
C -> B
D -> B
E -> B
F -> B
0
```

**Test Output To Screen**

```
Sources: NONE
Sinks: C
Sources: NONE
Sinks: NONE
Sources: D Q
Sinks: V
Sources: NONE
Sinks: NONE
Sources: NONE
Sinks: D F
Sources: NONE
Sinks: NONE
Sources: NONE
Sinks: NONE
Sources: NONE
Sinks: B C D
Sources: A C D E F
Sinks: B
Sources: NONE
Sinks: NONE
```

**Problem #7**  
**60 Points**

## 7. Kimberley

**Program Name: Kimberley.java**

**Input File: kimberley.dat**

**Test Input File: kimberley.dat**

```
9 3 1 3 5 2
12 6 1 3 5 7 10 11 4
10 7 1 3 4 6 8 9 10 3
8 4 4 5 6 7 4
10 4 1 4 8 10 5
20 9 2 5 8 9 10 11 14 17 20 5
10 6 3 4 6 7 9 10 5
10 4 7 8 9 10 7
```

**Test Output To Screen**

```
2 4 6
2 4 6 8 10 12
1 2 4 6 7 8 10
2 4 6 8
1 3 5 8
1 4 6 7 9 11 13 15 18
2 3 5 7 9 10
2 4 6 10
```



**Problem #8**  
**60 Points**

**8. Luana**

**Program Name: Luana.java**

**Input File: luana.dat**

**Test Input File: luana.dat**

```
*-----  
*-----  
--*-----  
----*-----  
--*---*---*  
-*---*---*  
--*-----*  
-*-----*  
*-----*  
-----*  
7  
2 2 4 2 3 4 4 5 2  
3 1 2 3 3 1 2 3 2 5 10  
2 1 2 2 1 2 3 1 1  
2 1 3 2 1 3 1 2 1  
2 1 3 2 1 3 3 5 5  
2 3 4 2 2 3 2 5 9  
2 2 3 2 1 2 5 5 9
```

**Test Output To Screen**

```
DADAA  
ADD  
AADD  
AA  
DDDA  
DDD  
DAADDA
```

**Problem #9**  
**60 Points**

**9. Maria**

**Program Name: Maria.java**

**Input File: maria.dat**

**Test Input File: maria.dat**

5 1 4 2 6 3

**Test Output To Screen**

\* \_ \*

\_ \* \_

\* \_ \*

\_ \_ \_

\_ \* \_

\_ \_ \_

\* \_ \*

\_ \_ \_

\* \_ \*

\* \_ \_

\_ \_ \_

\_ \_ \*

\* \_ \*

\* \_ \*

\* \_ \*

\* \_ \_

\_ \* \_

\_ \_ \*

**Problem #10**  
**60 Points**

**9. Patricia**

**Program Name: Patricia.java**

**Input File: patricia.dat**

**Test Input File: patricia.dat**

```
11
1 8 2 R
2 8 2 N R G
2 8 2 Y R G
1 7 7 G
2 4 8 Y G R
2 5 11 Y R R
1 8 2 G
2 5 11 Y R G
2 6 8 N G R
2 10 8 N G G
2 10 8 N R R
```

**Test Output To Screen**

```
8/10
16/90
16/100
7/14
32/144
25/256
2/10
55/256
48/182
56/306
90/306
```

**Problem #11**  
**60 Points**

**11. Rohit**

**Program Name: Rohit.java**

**Input File: rohit.dat**

**Test Input File: rohit.dat**

```
10
2 3 4
4
5 2 2
23
2 2 2 2 2 2 2
119
1 1 1 1 1
12
10 20 30 40 50
1000
1
0
0
8
100000
99999
100000
100000
100000
100001
```

**Test Output To Screen**

```
13 14 15 16
NO CHILDREN
239 240
NOT IN TREE
37771 37772 37773 37774 37775 37776 37777 37778 37779 37780 37781 37782 37783
37784 37785 37786 37787 37788 37789 37790 37791 37792 37793 37794 37795 37796
37797 37798 37799 37800 37801 37802 37803 37804 37805 37806 37807 37808 37809
37810
1
NOT IN TREE
NO CHILDREN
NO CHILDREN
NOT IN TREE
```

**Problem #12**  
**60 Points**

## 12. Tisha

**Program Name:** Tisha.java

**Input File:** None

**Test Input File:**None

**Sample Output To Screen**

```
  " ^ ^ ^ "
 /         \
@   o   o   @
 |     v   |
 \     -   /
  |       |
  |       |
```

**Judging criteria:**

Answers will vary.

The entire pattern:

- must span exactly eight rows and eight columns,
- with at least one character on each row,
- and at least one character in each column.
- There must be at least one \ (backslash) character and
- one " (double quote) character in the pattern.