```
11/30/2023 14:33:30
 ***  phi = 0  ***
Original points
V1  858 341  nextY 9
V2  627 427  nextY 8
V3  612 104  nextY 4
V4  466 177  nextY 6
V5  368 98  nextY 3
V6  203 204  nextY 7
V7  309 334  nextY 1
V8  272 483  nextY 14
V9  152 384  nextY 2
V10  108 663  nextY 15
V11  245 841  nextY 13
V12  389 752  nextY 11
V13  614 1011  nextY 0
V14  519 599  nextY 10
V15  738 667  nextY 12

Sorted points
Y1  368 98  V5
Y2  612 104  V3
Y3  466 177  V4
Y4  203 204  V6
Y5  309 334  V7
Y6  858 341  V1
Y7  152 384  V9
Y8  627 427  V2
Y9  272 483  V8
Y10  519 599  V14
Y11  108 663  V10
Y12  738 667  V15
Y13  389 752  V12
Y14  245 841  V11
Y15  614 1011  V13

k=1  V5  x368  y98
EdgeTree is empty
   Start
Insert Edge: #1  from V5 to V6
For Edge #1  from V5: set helper to V5

k=2  V3  x612  y104
R----H1 #1  From V5  prev: #--  next: #--
   Start
Insert Edge: #2  from V3 to V4
For Edge #2  from V3: set helper to V3

k=3  V4  x466  y177
R----H2 #1  From V5  prev: #--  next: #2
     R----H1 #2  From V3  prev: #1  next: #--
   Merge
Delete Edge: #2  from V3
Found EdgeJ #1  from V5 has helper V5
EdgeJ #1  from V5: set helper to V4

k=4  V6  x203  y204
R----H1 #1  From V5  prev: #--  next: #--
   Regular (down)
V(i-1) V5  has helper V4(m)
Parting line: V6  to helper V4
Delete Edge: #1  from V5
Insert Edge: #3  from V6 to V7
For Edge #3  from V6: set helper to V6

k=5  V7  x309  y334
R----H1 #3  From V6  prev: #--  next: #--
   Regular (down)
V(i-1) V6  has helper V6(r)
Delete Edge: #3  from V6
Insert Edge: #4  from V7 to V8
For Edge #4  from V7: set helper to V7

k=6  V1  x858  y341
R----H1 #4  From V7  prev: #--  next: #--
   Start
Insert Edge: #5  from V1 to V2
For Edge #5  from V1: set helper to V1
```

```
k=7  V9  x152  y384
R----H2 #4  From V7  prev: #--  next: #5
     R----H1 #5  From V1  prev: #4  next: #--
   Start
Insert Edge: #6  from V9 to V10
For Edge #6  from V9: set helper to V9

k=8  V2  x627  y427
R----H2 #4  From V7  prev: #6  next: #5
     L----H1 #6  From V9  prev: #--  next: #4
     R----H1 #5  From V1  prev: #4  next: #--
   Merge
Delete Edge: #5  from V1
Found EdgeJ #4  from V7 has helper V7
EdgeJ #4  from V7: set helper to V2

k=9  V8  x272  y483
R----H2 #4  From V7  prev: #6  next: #--
     L----H1 #6  From V9  prev: #--  next: #4
   Merge
Parting line: V8 to helper V2
Delete Edge: #4  from V7
Found EdgeJ #6  from V9 has helper V9
EdgeJ #6  from V9: set helper to V8

k=10  V14  x519  y599
R----H1 #6  From V9  prev: #--  next: #--
   Split
Found EdgeJ #6  from V9 with helper V8
Parting line: V14 to helper V8
EdgeJ #6  from V9: set helper to V14
Insert Edge: #7  from V14 to V15

k=11  V10  x108  y663
R----H2 #6  From V9  prev: #--  next: #7
     R----H1 #7  From V14  prev: #6  next: #--
   Regular (down)
V(i-1) V9  has helper V14(x)
Delete Edge: #6  from V9
Insert Edge: #8  from V10 to V11
For Edge #8  from V10: set helper to V10

k=12  V15  x738  y667
R----H2 #7  From V14  prev: #8  next: #--
     L----H1 #8  From V10  prev: #--  next: #7
   End
Delete Edge: #7  from V14

k=13  V12  x389  y752
R----H1 #8  From V10  prev: #--  next: #--
   Split
Found EdgeJ #8  from V10 with helper V10
Parting line: V12 to helper V10
EdgeJ #8  from V10: set helper to V12
Insert Edge: #9  from V12 to V13

k=14  V11  x245  y841
R----H2 #8  From V10  prev: #--  next: #9
     R----H1 #9  From V12  prev: #8  next: #--
   End
Delete Edge: #8  from V10

k=15  V13  x614  y1011
R----H1 #9  From V12  prev: #--  next: #--
   End
Delete Edge: #9  from V12

Parting lines
1  V6  V4
2  V8  V2
3  V8  V14
4  V10  V12

   Starting points for polygons
  V1  V3  V5  V9  V10
  partD/flags data
1 s  V0  --
2 m  V8  -f
3 s  V0  --
4 m  V6  -f
5 s  V0  --
6 r  V0  s-
7 r  V0  --
8 m  V14  ss
9 s  V0  --
10 r  V12  -s
11 e  V0  --
12 x  V0  f-
13 e  V0  --
14 x  V0  f-
15 e  V0  --

Processing monotone polygons

Processing polygon #1 with top vertex: V1
#1 j, jprev, jLeft, jRight: V1  V0  V2  V15
#2 j, jprev, jLeft, jRight: V2  V0  V8  V15

#3 j, jprev, jLeft, jRight: V8  V2  V14  V15
V(j) and V(S(nS)) V8  V2
V(j) and V(S(nS)) on same chain
S: 1  2
Checking intervening point 2
V(k) invalid: V8 to V1

#3 j, jprev, jLeft, jRight: V14  V8  V15  V15
V(j) and V(S(nS)) V14  V8
V(j) and V(S(nS)) on same chain
S: 1  2  8
Checking intervening point 8
Add V(k): V14 to V2 to V8
Checking intervening point 2
Add V(k): V14 to V1 to V2

Post processing
Add V(k): V14 to V1 to V15
======================================
Processing polygon #2 with top vertex: V3
#1 j, jprev, jLeft, jRight: V3  V0  V4  V2
#2 j, jprev, jLeft, jRight: V4  V0  V6  V2

#3 j, jprev, jLeft, jRight: V6  V4  V7  V2
V(j) and V(S(nS)) V6  V4
V(j) and V(S(nS)) on same chain
S: 3  4
Checking intervening point 4
V(k) invalid: V6 to V3

#3 j, jprev, jLeft, jRight: V7  V6  V8  V2
V(j) and V(S(nS)) V7  V6
V(j) and V(S(nS)) on same chain
S: 3  4  6
Checking intervening point 6
Add V(k): V7 to V4 to V6
Checking intervening point 4
Add V(k): V7 to V3 to V4

#3 j, jprev, jLeft, jRight: V2  V7  V8  V8
V(j) and V(S(nS)) V2  V7
V(j) and V(S(nS)) on different chains
S: 3  7
Add V(k): V2 to V7 to V3

Post processing
Add V(k): V2 to V7 to V8
======================================
Processing polygon #3 with top vertex: V5
#1 j, jprev, jLeft, jRight: V5  V0  V6  V4
#2 j, jprev, jLeft, jRight: V4  V0  V6  V6
#3 j, jprev, jLeft, jRight: V4  V4  V6  V6
V(j) and V(S(nS)) V4  V4
V(j) and V(S(nS)) on same chain
S: 5  4
Checking intervening point 4
V(k) invalid: V4 to V5

Post processing
Add V(k): V4 to V5 to V6
======================================
Processing polygon #4 with top vertex: V9
#1 j, jprev, jLeft, jRight: V9  V0  V10  V8
#2 j, jprev, jLeft, jRight: V8  V0  V10  V14

#3 j, jprev, jLeft, jRight: V14  V8  V10  V13
V(j) and V(S(nS)) V14  V8
V(j) and V(S(nS)) on same chain
S: 9  8
Checking intervening point 8
V(k) invalid: V14 to V9

#3 j, jprev, jLeft, jRight: V10  V14  V12  V13
V(j) and V(S(nS)) V10  V14
V(j) and V(S(nS)) on different chains
S: 9  8  14
Add V(k): V10 to V14 to V8
Add V(k): V10 to V8 to V9

#3 j, jprev, jLeft, jRight: V12  V10  V13  V13
V(j) and V(S(nS)) V12  V10
V(j) and V(S(nS)) on same chain
S: 14  10
Checking intervening point 10
Add V(k): V12 to V14 to V10

Post processing
Add V(k): V12 to V14 to V13
======================================
Processing polygon #5 with top vertex: V10
#1 j, jprev, jLeft, jRight: V10  V0  V11  V12
#2 j, jprev, jLeft, jRight: V12  V0  V11  V11

#3 j, jprev, jLeft, jRight: V12  V12  V11  V11
V(j) and V(S(nS)) V12  V12
V(j) and V(S(nS)) on same chain
S: 10  12
Checking intervening point 12
V(k) invalid: V12 to V10

Post processing
Add V(k): V12 to V10 to V11
End of monotone processing
13 triangles.

Triangle indices and areas
1:   14 2 8    55012  8%  0.43
2:   14 1 2    30444  5%  0.17
   ***Eliminating sliver***
3:   14 1 15    79554  12%  0.44
2:   2 15 14    45012  7%  0.25  revised
3:   2 15 1    64986  10%  0.36  revised
4:   7 4 6    37052  6%  0.53
5:   7 3 4    11461  2%  0.08
   ***Eliminating sliver***
6:   2 7 3    101319  16%  0.7
5:   4 2 7    64527  10%  0.45  revised
6:   4 2 3    48253  7%  0.33  revised
7:   2 7 8    50823  8%  0.39
8:   4 5 6    23423  4%  0.34
9:   10 14 8    63484  10%  0.37
10:   10 8 9    37836  6%  0.47
11:   12 14 10    54563  8%  0.32
12:   12 14 13    68095  10%  0.38
13:   12 10 11    37825  6%  0.44
Polygon area: 650891
```