

Stepwise Refinement

Problem 1

- Write a program that outputs the following lines:

0 0 0

0 0 0

0 0 0

0 0 0

0 0 0

Stepwise Refinement

- Step 1:

do 5 times, $i = 1, \dots, 5$

print "0 0 0 \n"

- Step 2:

do 5 times, $i = 1, \dots, 5$

do 3 times, $j = 1, 2, 3$

print "0 "

print "\n"

Stepwise Refinement

- Step 3:

```
// do 5 times, i = 1, ..., 5
for (i=1; i<=5; i++) {
    // do 3 times, j = 1, 2, 3
    for (j=1; j<=3; j++)
        // print "0  "
        printf("0  ");
    // print "\n"
    printf("\n");
}
```

Problem 2

- Write a program that outputs the following lines:

```
1 2 3 4 5 6 7 8 9
2 3 4 5 6 7 8 9
3 4 5 6 7 8 9
4 5 6 7 8 9
5 6 7 8 9
6 7 8 9
7 8 9
8 9
9
```

Stepwise Refinement

- Step 1:

do 9 times, $i = 1, \dots, 9$

print a line "? ... 9 \n"

- Step 2:

do 9 times, $i = 1, \dots, 9$

print i ... 9 "\n"

Stepwise Refinement

- Step 3:

do 9 times, $i = 1, \dots, 9$

for $j = i, \dots, 9$

print j

print "\n"

Stepwise Refinement

- Step 4:

```
// do 9 times, i = 1, ..., 9
for (i=1; i<=9; i++) {
    // for j = i, ..., 9
    for (j=i; j<=9; j++)
        // print j
        printf("%d  ", j);
    // print "\n"
    printf("\n");
}
```


Problem 3

- Write a program that outputs the following lines:

```
9
8 9
7 8 9
6 7 8 9
5 6 7 8 9
4 5 6 7 8 9
3 4 5 6 7 8 9
2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
```

Stepwise Refinement

- Step 1:

do 9 times, $i = 1, \dots, 9$

print a line "? ... 9 \n"

- Step 2:

do 9 times, $i = 1, \dots, 9$

print i numbers, $10-i \dots 9 \text{ "\n"}$

Stepwise Refinement

- Step 3:

do 9 times, $i = 1, \dots, 9$

for $j = 10-i, \dots, 9$

print j

print "\n"

Stepwise Refinement

- Step 4:

```
// do 9 times, i = 1, ..., 9
for (i=1; i<=9; i++) {
    // for j = 10-i, ..., 9
    for (j=10-i; j<=9; j++)
        // print j
        printf("%d  ", j);
    // print "\n"
    printf("\n");
}
```

Further Extensions

- Try other possible extensions, such as
- Other orderings, e.g.,
9, 8, ..., 1
2, 5, 8, 11, ...
- Other shapes, e.g.,

xxxxxx	or	xxxxxx	or	x
xxxxx		xxxxxx		xxx
xxx		xxxxxx		xxxxxx
xx		xxxxxx		xxx
x		xxxxxx		x

Loop invariant

- Regularities that one can exploit to write a loop successfully