Exam document

Blanks

Initial

```
(assert (and



))
(check-sat)
(get-value (









))
```

FORALL (steps)

```
(forall ((steps Int))
    (=>
        (<= 1 steps EndStep)



    )
)
```

EXISTS (steps)

```
(exists ((step Int))
    (and
        (<= 1 step EndStep)


    )
)
```

DISTINCT ROWS

```
(declare-fun M (Int Int) Int)
(forall ((r Int))
    (=>
        (<= 1 r 5)
        (distinct (M r 1) (M r 2) (M r 3) (M r 4) (M r 5))
    )
)
```

DISTINCT COLS

```
(declare-fun M (Int Int) Int)
(forall ((c Int))
    (=>
        (<= 1 c 5)
        (distinct (M 1 c) (M 2 c) (M 3 c) (M 4 c) (M 5 c))
    )
```

ALL VALUES ARE FROM x TO y

```
(declare-fun M (Int Int) Int)

(define-fun Values ((v Int)) Bool
    (and
        (<= 1 (M 1 v) 10)
        (<= 1 (M 2 v) 10)
        (<= 1 (M 3 v) 10)
        (<= 1 (M 4 v) 10)
        (<= 1 (M 5 v) 10)
        (<= 1 (M 6 v) 10)
        (<= 1 (M 7 v) 10)
        (<= 1 (M 8 v) 10)
    )
)

(Values 1) (Values 2) (Values 3) (Values 4) (Values 5)
(Values 6) (Values 7) (Values 8)
```

COUNT ONE PER ROW

```
(define-fun Count ((v Int)) Bool
    (+
        (M 1 v)
        (M 2 v)
        (M 3 v)
        (M 4 v)
        (M 5 v)
    )
)

(= (Count 1) 1) (= (Count 2) 1) (= (Count 3) 1) (= (Count 4) 1) (= (Count 5)
1)
```

COUNT ONE PER COL

```
(declare-fun M (Int Int) Int)

(define-fun Count ((v Int)) Bool
    (+
        (M v 1)
        (M v 2)
        (M v 3)
        (M v 4)
        (M v 5)
    )
)

(= (Count 1) 1) (= (Count 2) 1) (= (Count 3) 1) (= (Count 4) 1) (= (Count 5) 1)
```