This card can be 'programmed' by carefully ripping or cutting a notch between a punched hole and the nearest edge.

The card goes into a card database stack which can then be 'queried' using punch/notch data.

E.g. if you have entered a github profile address here

We can query the database by inserting a needle into that hole of the whole card stack and filtering (lifting) out all cards that do not have github profiles.

email:

.....

github profile:

.....

Notch this hole if you are currently seeking exciting engineering opportunities

CRAPTCHA® Notch this hole to prove you are a human

ANIMOTO

8 7 6 5 4 3 2 1

Punchcard Coding Challenge

When you return this card to us, it will be randomly placed into a stack of other cards that have already been notched to represent unique, sequential id numbers between 0 and 100 so that they can be mechanically sorted.

To complete this challenge, code THIS card by appropriately notching holes 1-8 above to represent this id number:

n =

.....

The card stack will be mechanically sorted using this obscure pseudocode:

```
for ( i = 1; i <= 8; i++) {
  insertNeedleIntoHole(i);
  unnotchedCards = liftOutUnnotchedCards();
  placeAtFrontOfStack(unnotchedCards);
}
```

After running the sorting procedure, the cards will all be in ascending id order and your card should appear in the stack ordered in position n+1

This card can be 'programmed' by carefully ripping or cutting a notch between a punched hole and the nearest edge.

The card goes into a card database stack which can then be 'queried' using punch/notch data.

E.g. if you have entered a github profile address here

We can query the database by inserting a needle into that hole of the whole card stack and filtering (lifting) out all cards that do not have github profiles.

email:

.....

github profile:

.....

Notch this hole if you are currently seeking exciting engineering opportunities

CRAPTCHA® Notch this hole to prove you are a human

ANIMOTO

8 7 6 5 4 3 2 1

Punchcard Coding Challenge

When you return this card to us, it will be randomly placed into a stack of other cards that have already been notched to represent unique, sequential id numbers between 0 and 100 so that they can be mechanically sorted.

To complete this challenge, code THIS card by appropriately notching holes 1-8 above to represent this id number:

n =

.....

The card stack will be mechanically sorted using this obscure pseudocode:

```
for ( i = 1; i <= 8; i++) {
  insertNeedleIntoHole(i);
  unnotchedCards = liftOutUnnotchedCards();
  placeAtFrontOfStack(unnotchedCards);
}
```

After running the sorting procedure, the cards will all be in ascending id order and your card should appear in the stack ordered in position n+1

This card can be 'programmed' by carefully ripping or cutting a notch between a punched hole and the nearest edge.

The card goes into a card database stack which can then be 'queried' using punch/notch data.

E.g. if you have entered a github profile address here

We can query the database by inserting a needle into that hole of the whole card stack and filtering (lifting) out all cards that do not have github profiles.

email:

.....

github profile:

.....

Notch this hole if you are currently seeking exciting engineering opportunities

CRAPTCHA® Notch this hole to prove you are a human

ANIMOTO

8 7 6 5 4 3 2 1

Punchcard Coding Challenge

When you return this card to us, it will be randomly placed into a stack of other cards that have already been notched to represent unique, sequential id numbers between 0 and 100 so that they can be mechanically sorted.

To complete this challenge, code THIS card by appropriately notching holes 1-8 above to represent this id number:

n =

.....

The card stack will be mechanically sorted using this obscure pseudocode:

```
for ( i = 1; i <= 8; i++) {
  insertNeedleIntoHole(i);
  unnotchedCards = liftOutUnnotchedCards();
  placeAtFrontOfStack(unnotchedCards);
}
```

After running the sorting procedure, the cards will all be in ascending id order and your card should appear in the stack ordered in position n+1

This card can be 'programmed' by carefully ripping or cutting a notch between a punched hole and the nearest edge.

The card goes into a card database stack which can then be 'queried' using punch/notch data.

E.g. if you have entered a github profile address here

We can query the database by inserting a needle into that hole of the whole card stack and filtering (lifting) out all cards that do not have github profiles.

email:

.....

github profile:

.....

Notch this hole if you are currently seeking exciting engineering opportunities

CRAPTCHA® Notch this hole to prove you are a human

ANIMOTO

8 7 6 5 4 3 2 1

Punchcard Coding Challenge

When you return this card to us, it will be randomly placed into a stack of other cards that have already been notched to represent unique, sequential id numbers between 0 and 100 so that they can be mechanically sorted.

To complete this challenge, code THIS card by appropriately notching holes 1-8 above to represent this id number:

n =

.....

The card stack will be mechanically sorted using this obscure pseudocode:

```
for ( i = 1; i <= 8; i++) {
  insertNeedleIntoHole(i);
  unnotchedCards = liftOutUnnotchedCards();
  placeAtFrontOfStack(unnotchedCards);
}
```

After running the sorting procedure, the cards will all be in ascending id order and your card should appear in the stack ordered in position n+1