

Sailunsi Chen  
Lab 4 Submission

```
===== Data Wrangler Script: CMSC
from wrangler import dw
import sys

if(len(sys.argv) < 3):
    sys.exit('Error: Please include an input and output file. Example python
script.py input.csv output.csv')

w = dw.DataWrangler()

# Split data repeatedly on newline into rows
w.add(dw.Split(column=["data"],
               table=0,
               status="active",
               drop=True,
               result="row",
               update=False,
               insert_position="right",
               row=None,
               on="\n",
               before=None,
               after=None,
               ignore_between=None,
               which=1,
               max=0,
               positions=None,
               quote_character=None))

# Wrap rows where data starts with 'CMSC'
w.add(dw.Wrap(column=[],
              table=0,
              status="active",
              drop=False,
              row=dw.Row(column=[]),
              table=0,
              status="active",
              drop=False,
              conditions=[dw.StartsWith(column=[],
                                       table=0,
                                       status="active",
                                       drop=False,
                                       lcol="data",
                                       value="CMSC",
                                       op_str="starts with")]))))

w.apply_to_file(sys.argv[1]).print_csv(sys.argv[2])
```

	wrap	wrap1	wrap2	wrap3	wrap4
1	CMSC100	0101	Charles Kassir	Seats (Total: 45, Open: 4, Waitlist: 0)	M 4:00pm - 4:50pm
2	CMSC106	0101	Jianwu Wang	Seats (Total: 45, Open: 0, Waitlist: 5)	TuTh 9:30am - 10:45am
3	CMSC131	0101	Evan Golub	Seats (Total: 31, Open: 0, Waitlist: 0)	MWF 2:00pm - 2:50pm
4	CMSC132	0101	Laurence Herman	Seats (Total: 34, Open: 0, Waitlist: 2)	MWF 10:00am - 10:50am
5	CMSC216	0101	Nelson Padua-Perez	Seats (Total: 28, Open: 2, Waitlist: 0)	TuTh 9:30am - 10:45am
6	CMSC250	0101	Clyde Kruskal	Seats (Total: 29, Open: 0, Waitlist: 4)	TuTh 2:00pm - 3:15pm
7	CMSC250	0101	Thomas Goldstein	Seats (Total: 25, Open: 0, Waitlist: 0)	TuTh 12:30pm - 1:45pm
8	CMSC289	0101	James Reggia	Seats (Total: 60, Open: 0, Waitlist: 36)	TuTh 11:00am - 12:15pm
9	CMSC330	0101	Chau-Wen Tseng	Seats (Total: 27, Open: 0, Waitlist: 0)	TuTh 3:30pm - 4:45pm
10	CMSC351	0101	Hamid Mahini	Seats (Total: 88, Open: 0, Waitlist: 2)	MWF 10:00am - 10:50am
11	CMSC396	0101	Atif Memon , Neil Spring	Seats (Total: 25, Open: 0, Waitlist: 0)	W 1:00pm - 1:50pm
12	CMSC411	0101	Michelle Hugue	Seats (Total: 45, Open: 0, Waitlist: 3)	TuTh 3:30pm - 4:45pm
13	CMSC412	0101	Neil Spring	Seats (Total: 25, Open: 0, Waitlist: 4)	TuTh 11:00am - 12:15pm
14	CMSC414	0101	A.U. Shankar	Seats (Total: 40, Open: 0, Waitlist: 6)	TuTh 12:30pm - 1:45pm

===== Data Wrangler Script: World Cup 1

```
from wrangler import dw
import sys
```

```
if(len(sys.argv) < 3):
    sys.exit('Error: Please include an input and output file. Example python
script.py input.csv output.csv')
```

```
w = dw.DataWrangler()
```

```
# Split data repeatedly on newline into rows
w.add(dw.Split(column=["data"],
    table=0,
    status="active",
    drop=True,
    result="row",
    update=False,
    insert_position="right",
    row=None,
    on="\n",
    before=None,
    after=None,
    ignore_between=None,
    which=1,
    max=0,
    positions=None,
    quote_character=None))
```

```
# Wrap rows where data contains '|-'
w.add(dw.Wrap(column=[],
    table=0,
    status="active",
```

```

        drop=False,
        row=dw.Row(column=[]),
        table=0,
        status="active",
        drop=False,
        conditions=[dw.Contains(column=[],
            table=0,
            status="active",
            drop=False,
            lcol="data",
            value="|-",
            op_str="contains"))]))

```

```

# Drop wrap
w.add(dw.Drop(column=["wrap"],
    table=0,
    status="active",
    drop=True))

```

```

# Delete row 1
w.add(dw.Filter(column=[],
    table=0,
    status="active",
    drop=False,
    row=dw.Row(column=[]),
    table=0,
    status="active",
    drop=False,
    conditions=[dw.RowIndex(column=[],
        table=0,
        status="active",
        drop=False,
        indices=[0])]))))

```

```

# Extract from wrap1 between 'fb|' and '}'
w.add(dw.Extract(column=["wrap1"],
    table=0,
    status="active",
    drop=False,
    result="column",
    update=False,
    insert_position="right",
    row=None,
    on=".*",
    before="}",
    after="fb\\|",
    ignore_between=None,
    which=1,
    max=1,
    positions=None))

```

```

# Drop wrap1
w.add(dw.Drop(column=["wrap1"],
    table=0,
    status="active",
    drop=True))

```

```

# Split wrap2 repeatedly on ',' into rows
w.add(dw.Split(column=["wrap2"],
    table=0,
    status="active",
    drop=True,
    result="row",
    update=False,
    insert_position="right",
    row=None,
    on=",",
    before=None,
    after=None,
    ignore_between=None,
    which=1,
    max="0",
    positions=None,
    quote_character=None))

# Extract from wrap2 between '[' and ' FIFA'
w.add(dw.Extract(column=["wrap2"],
    table=0,
    status="active",
    drop=False,
    result="column",
    update=False,
    insert_position="right",
    row=None,
    on=".*",
    before=" FIFA",
    after="\\[\\[",
    ignore_between=None,
    which=1,
    max=1,
    positions=None))

# Drop wrap2
w.add(dw.Drop(column=["wrap2"],
    table=0,
    status="active",
    drop=True))

# Edit extract1 row 1 to ' 1958, 1 '
w.add(dw.Edit(column=["extract1"],
    table=0,
    status="active",
    drop=False,
    result="column",
    update=True,
    insert_position="right",
    row=dw.Row(column=[]),
    table=0,
    status="active",
    drop=False,
    conditions=[dw.RowIndex(column=[]),
        table=0,
        status="active",
        drop=False,

```

```

        indices=[0]))),
on=None,
before=None,
after=None,
ignore_between=None,
which=1,
max=1,
positions=None,
to="1958, 1",
update_method=None))

# Edit extract1 row 2 to ' 1962, 1 '
w.add(dw.Edit(column=["extract1"],
    table=0,
    status="active",
    drop=False,
    result="column",
    update=True,
    insert_position="right",
    row=dw.Row(column=[]),
    table=0,
    status="active",
    drop=False,
    conditions=[dw.RowIndex(column=[],
        table=0,
        status="active",
        drop=False,
        indices=[1]))),
on=None,
before=None,
after=None,
ignore_between=None,
which=1,
max=1,
positions=None,
to="1962, 1",
update_method=None))

# Edit extract1 row 3 to ' 1970, 1 '
w.add(dw.Edit(column=["extract1"],
    table=0,
    status="active",
    drop=False,
    result="column",
    update=True,
    insert_position="right",
    row=dw.Row(column=[]),
    table=0,
    status="active",
    drop=False,
    conditions=[dw.RowIndex(column=[],
        table=0,
        status="active",
        drop=False,
        indices=[2]))),
on=None,
before=None,

```

```

        after=None,
        ignore_between=None,
        which=1,
        max=1,
        positions=None,
        to="1970, 1",
        update_method=None))

# Edit extract1 row 4 to ' 1994, 1 '
w.add(dw.Edit(column=["extract1"],
    table=0,
    status="active",
    drop=False,
    result="column",
    update=True,
    insert_position="right",
    row=dw.Row(column=[]),
    table=0,
    status="active",
    drop=False,
    conditions=[dw.RowIndex(column=[],
        table=0,
        status="active",
        drop=False,
        indices=[3])]),
    on=None,
    before=None,
    after=None,
    ignore_between=None,
    which=1,
    max=1,
    positions=None,
    to="1994, 1",
    update_method=None))

# Edit extract1 row 5 to ' 2002, 1 '
w.add(dw.Edit(column=["extract1"],
    table=0,
    status="active",
    drop=False,
    result="column",
    update=True,
    insert_position="right",
    row=dw.Row(column=[]),
    table=0,
    status="active",
    drop=False,
    conditions=[dw.RowIndex(column=[],
        table=0,
        status="active",
        drop=False,
        indices=[4])]),
    on=None,
    before=None,
    after=None,
    ignore_between=None,
    which=1,

```

```

max=1,
positions=None,
to="2002, 1",
update_method=None))

```

```

# Edit extract1 row 6 to ' 1954, 1 '
w.add(dw.Edit(column=["extract1"],
               table=0,
               status="active",
               drop=False,
               result="column",
               update=True,
               insert_position="right",
               row=dw.Row(column=[]),
               table=0,
               status="active",
               drop=False,
               conditions=[dw.RowIndex(column=[],
                                       table=0,
                                       status="active",
                                       drop=False,
                                       indices=[5])]),
               on=None,
               before=None,
               after=None,
               ignore_between=None,
               which=1,
               max=1,
               positions=None,
               to="1954, 1",
               update_method=None))

```

```

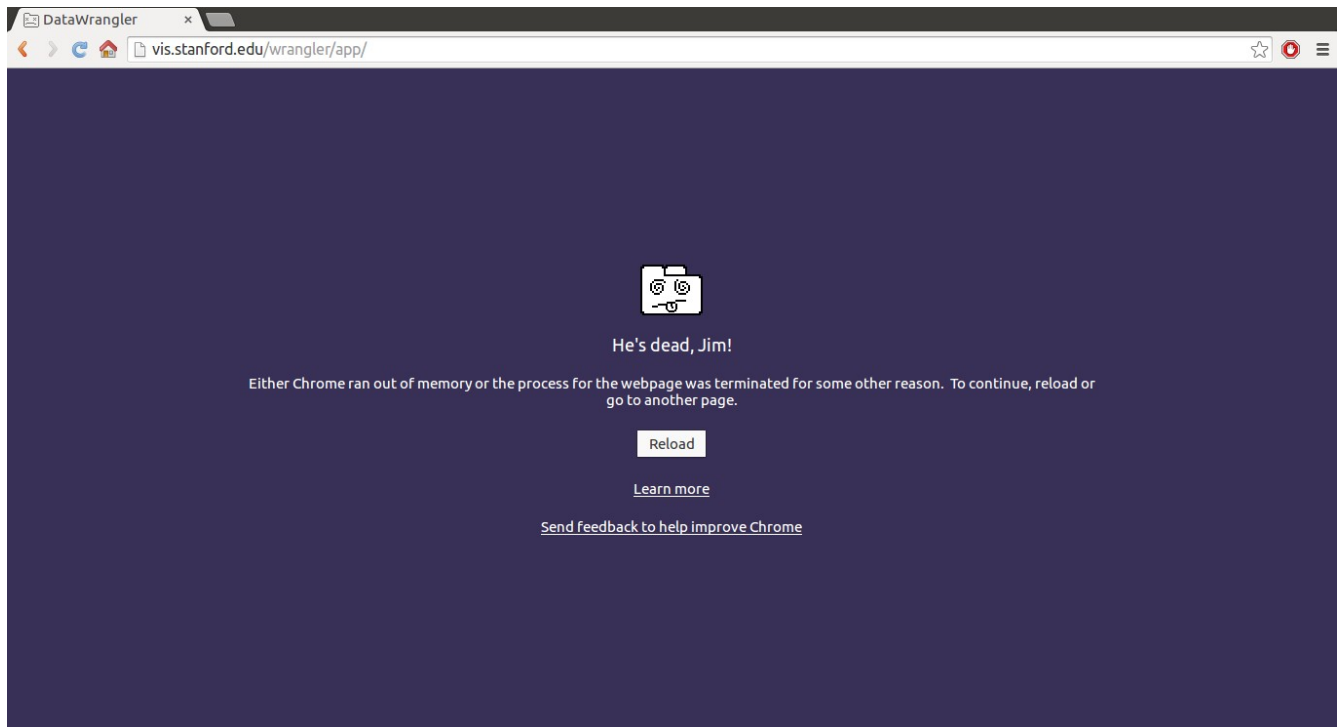
w.apply_to_file(sys.argv[1]).print_csv(sys.argv[2])

```

The screenshot shows the DataWrangler web interface. The top navigation bar includes buttons for Split, Cut, Extract, Edit, Fill, Translate, Drop, Merge, Wrap, Delete, Promote, Fold, Unfold, and Transpose. The main table displays 36 rows of data. The columns are labeled: extract, extract1, wrap3, wrap4, and wrap5. The data includes country codes (BRA, GER), years (1958, 1962, 1970, 1994, 2002, 1954), and detailed tournament information (e.g., '1950 FIFA World Cup', '1938 FIFA World Cup'). A 'Script' panel on the left shows a sequence of operations: 'Split wrap2 repeatedly on \',\' into rows', 'Extract from wrap2 between '[\' and \' FIFA\'', 'Drop wrap2', and 'Edit extract1 row 1 to \' 1958, 1 \''. The table has a 'rows: 36' indicator and 'prev next' navigation links.

	extract	extract1	wrap3	wrap4	wrap5
1	BRA	1958, 1	12 ([[1950 FIFA World Cup 1950]] [#1 *]], [[1998 FIFA World Cup 1998]])	12 ([[1938 FIFA World Cup 1938]], [[1978 FIFA World Cup 1978]])	12 ([[1974 FIFA World Cup 1974]], [[2014 FIFA World Cup 2014]])
2	BRA	1962, 1	12 ([[1950 FIFA World Cup 1950]] [#1 *]], [[1998 FIFA World Cup 1998]])	12 ([[1938 FIFA World Cup 1938]], [[1978 FIFA World Cup 1978]])	12 ([[1974 FIFA World Cup 1974]], [[2014 FIFA World Cup 2014]])
3	BRA	1970, 1	12 ([[1950 FIFA World Cup 1950]] [#1 *]], [[1998 FIFA World Cup 1998]])	12 ([[1938 FIFA World Cup 1938]], [[1978 FIFA World Cup 1978]])	12 ([[1974 FIFA World Cup 1974]], [[2014 FIFA World Cup 2014]])
4	BRA	1994, 1	12 ([[1950 FIFA World Cup 1950]] [#1 *]], [[1998 FIFA World Cup 1998]])	12 ([[1938 FIFA World Cup 1938]], [[1978 FIFA World Cup 1978]])	12 ([[1974 FIFA World Cup 1974]], [[2014 FIFA World Cup 2014]])
5	BRA	2002, 1	12 ([[1950 FIFA World Cup 1950]] [#1 *]], [[1998 FIFA World Cup 1998]])	12 ([[1938 FIFA World Cup 1938]], [[1978 FIFA World Cup 1978]])	12 ([[1974 FIFA World Cup 1974]], [[2014 FIFA World Cup 2014]])
6	GER	1954, 1	14 ([[1966 FIFA World Cup 1966]], [[1982 FIFA World Cup 1982]], [[1986 FIFA World Cup 1986]], [[2002 FIFA World Cup 2002]])	14 ([[1934 FIFA World Cup 1934]], [[1970 FIFA World Cup 1970]], [[2006 FIFA World Cup 2006]] [#1 *]], [[2010 FIFA World Cup 2010]])	11 ([[1958 FIFA World Cup 1958]])
			14 ([[1966 FIFA World Cup 1966]], [[1982 FIFA World Cup 1982]], [[1986 FIFA World Cup 1986]], [[2002 FIFA World Cup 2002]])	14 ([[1934 FIFA World Cup 1934]], [[1970 FIFA World Cup 1970]], [[2006 FIFA World Cup 2006]] [#1 *]], [[2010 FIFA World Cup 2010]])	

===== Data Wrangler Script: World Cup 2  
Data wrangler keeps crashing before I can finish World Cup 1.  
I have given up on moving forward with it. This is not even  
remotely close to a competent tool for cleaning up data.





===== UNIX Tools Command: CMSC

```
cat cmsc.txt | sed '/^$/d; s/[()]/g; s/[[:space:]]*$/ ' | awk '/^([A-Z]{3,3}|([M|W|F|TuTh]+ ))/ {print $1", "$2, $3, $4} !/^([A-Z]{3,3}|([M|W|F|TuTh]+ ))/ {print $0}' | awk '/^CMSC/ {class = $1} /^[0-9]+$/ {print class, $0} !/^CMSC|^([0-9]+|^Seats)/ {print $0} /^Seats/ {print $3, $5, $7}' | awk -F', ' 'BEGIN {printf "Course No., Section No., Instructor, Seats, Open, Waitlist, Days, Time, Bldg., Room No."} /^CMSC/ {print concat; concat = $0} !/^CMSC/ {concat = concat", "$0;} END {print concat}'
```

```
[serrintine@aperturelabs:Tab4]$ cat cmsc.txt | sed '/^$/d; s/[()]/g; s/[[:space:]]*$/ ' | awk '/^([A-Z]{3,3}|([M|W|F|TuTh]+ ))/ {print $1", "$2, $3, $4} !/^([A-Z]{3,3}|([M|W|F|TuTh]+ ))/ {print $0}' | awk '/^CMSC/ {class = $1} /^[0-9]+$/ {print class, $0} !/^CMSC|^([0-9]+|^Seats)/ {print $0} /^Seats/ {print $3, $5, $7}' | awk -F', ' 'BEGIN {printf "Course No., Section No., Instructor, Seats, Open, Waitlist, Days, Time, Bldg., Room No."} /^CMSC/ {print concat; concat = $0} !/^CMSC/ {concat = concat", "$0;} END {print concat}'
Course No., Section No., Instructor, Seats, Open, Waitlist, Days, Time, Bldg., Room No.
CMSC100, 0101, Charles Kassir, 45, 4, 0, M, 4:00pm - 4:50pm, CSI, 2117
CMSC106, 0101, Jianwu Wang, 45, 0, 5, TuTh, 9:30am - 10:45am, CSI, 2117
CMSC131, 0101, Evan Golub, 31, 0, 0, MWF, 2:00pm - 2:50pm, CSI, 1115
CMSC131, 0102, Evan Golub, 31, 0, 0, MWF, 2:00pm - 2:50pm, CSI, 1115
CMSC131, 0103, Evan Golub, 31, 1, 0, MWF, 2:00pm - 2:50pm, CSI, 1115
CMSC131, 0104, Evan Golub, 31, 1, 0, MWF, 2:00pm - 2:50pm, CSI, 1115
CMSC131, 0201, Evan Golub, 31, 1, 0, MWF, 3:00pm - 3:50pm, CSI, 1115
CMSC131, 0202, Evan Golub, 31, 1, 0, MWF, 3:00pm - 3:50pm, CSI, 1115
CMSC131, 0203, Evan Golub, 31, 0, 0, MWF, 3:00pm - 3:50pm, CSI, 1115
CMSC131, 0204, Evan Golub, 31, 0, 1, MWF, 3:00pm - 3:50pm, CSI, 1115
CMSC131, 0301, Thomas Reinhardt, 31, 0, 0, MWF, 11:00am - 11:50am, CSI, 1115
CMSC131, 0302, Thomas Reinhardt, 31, 0, 0, MWF, 11:00am - 11:50am, CSI, 1115
CMSC131, 0303, Thomas Reinhardt, 31, 0, 0, MWF, 11:00am - 11:50am, CSI, 1115
CMSC131, 0304, Thomas Reinhardt, 31, 0, 0, MWF, 11:00am - 11:50am, CSI, 1115
CMSC131, 0401, Thomas Reinhardt, 31, 0, 0, MWF, 12:00pm - 12:50pm, CSI, 1115
CMSC131, 0402, Thomas Reinhardt, 31, 7, 0, MWF, 12:00pm - 12:50pm, CSI, 1115
CMSC131, 0403, Thomas Reinhardt, 31, 0, 0, MWF, 12:00pm - 12:50pm, CSI, 1115
CMSC131, 0404, Thomas Reinhardt, 31, 7, 0, MWF, 12:00pm - 12:50pm, CSI, 1115
CMSC132, 0101, Laurence Herman, 34, 0, 2, MWF, 10:00am - 10:50am, CSI, 1115
CMSC132, 0102, Laurence Herman, 34, 0, 0, MWF, 10:00am - 10:50am, CSI, 1115
CMSC132, 0103, Laurence Herman, 34, 0, 0, MWF, 10:00am - 10:50am, CSI, 1115
CMSC132, 0104, Laurence Herman, 34, 0, 2, MWF, 10:00am - 10:50am, CSI, 1115
CMSC132, 0201, Laurence Herman, 34, 6, 0, MWF, 1:00pm - 1:50pm, CSI, 1115
CMSC132, 0202, Laurence Herman, 34, 1, 0, MWF, 1:00pm - 1:50pm, CSI, 1115
CMSC132, 0203, Laurence Herman, 34, 0, 0, MWF, 1:00pm - 1:50pm, CSI, 1115
CMSC132, 0204, Laurence Herman, 34, 0, 0, MWF, 1:00pm - 1:50pm, CSI, 1115
CMSC132, 0301, Laurence Herman, 30, 3, 0, MWF, 2:00pm - 2:50pm, CSI, 2117
CMSC132, 0302, Laurence Herman, 29, 0, 1, MWF, 2:00pm - 2:50pm, CSI, 2117
CMSC132, 0303, Laurence Herman, 29, 0, 0, MWF, 2:00pm - 2:50pm, CSI, 2117
CMSC216, 0101, Nelson Padua-Perez, 28, 2, 0, TuTh, 9:30am - 10:45am, CSI, 1115
CMSC216, 0102, Nelson Padua-Perez, 28, 0, 0, TuTh, 9:30am - 10:45am, CSI, 1115
CMSC216, 0103, Nelson Padua-Perez, 28, 0, 1, TuTh, 9:30am - 10:45am, CSI, 1115
CMSC216, 0104, Nelson Padua-Perez, 28, 0, 0, TuTh, 9:30am - 10:45am, CSI, 1115
CMSC216, 0201, Nelson Padua-Perez, 28, 0, 0, TuTh, 11:00am - 12:15pm, CSI, 1115
CMSC216, 0202, Nelson Padua-Perez, 28, 0, 1, TuTh, 11:00am - 12:15pm, CSI, 1115
CMSC216, 0203, Nelson Padua-Perez, 28, 0, 1, TuTh, 11:00am - 12:15pm, CSI, 1115
CMSC216, 0204, Nelson Padua-Perez, 28, 0, 0, TuTh, 11:00am - 12:15pm, CSI, 1115
```

===== UNIX Tools Command: World Cup 1

```
cat worldcup.txt | tail -n +3 | tr -d "<>{}[]-|=:#*/\\"" | sed 's/ FIFA W C[0-9]*//g; s/[0-9]\\{1,3\\} //g; s/^ [0-9]\\{1,5\\} //g; s/[0-9]$//g' | tr -d "()-" | awk '/^[A-Z]/ {country = $1} !/^[A-Z]/ {print country; split($0, arr, ","); for (i = 1; i <= 5; i++) printf("%d, %d\\n", arr[i], (NR-1)%7)}' | awk '/^[A-Z]/ {country = $1} /^[1-9]/ {print country, "$0}"
```

```
[serrintine@aperturelabs:lab4]$ cat worldcup.txt | tail -n +3 | tr -d "<>{}[]-|=:#*/\\"" | sed 's/ FIFA W C[0-9]*//g; s/[0-9]\\{1,3\\} //g; s/^ [0-9]\\{1,5\\} //g; s/[0-9]$//g' | tr -d "()-" | awk '/^[A-Z]/ {country = $1} !/^[A-Z]/ {print country; split($0, arr, ","); for (i = 1; i <= 5; i++) printf("%d, %d\\n", arr[i], (NR-1)%7)}' | awk '/^[A-Z]/ {country = $1} /^[1-9]/ {print country, "$0}"
BRA, 1958, 1
BRA, 1962, 1
BRA, 1970, 1
BRA, 1994, 1
BRA, 2002, 1
BRA, 1950, 2
BRA, 1998, 2
BRA, 1938, 3
BRA, 1978, 3
BRA, 1974, 4
BRA, 2014, 4
GER, 1954, 1
GER, 1974, 1
GER, 1990, 1
GER, 2014, 1
GER, 1966, 2
GER, 1982, 2
GER, 1986, 2
GER, 2002, 2
GER, 1934, 3
GER, 1970, 3
GER, 2006, 3
GER, 2010, 3
GER, 1958, 4
ITA, 1934, 1
ITA, 1938, 1
ITA, 1982, 1
ITA, 2006, 1
ITA, 1970, 2
ITA, 1994, 2
ITA, 1990, 3
ITA, 1978, 4
ARG, 1978, 1
ARG, 1986, 1
ARG, 1930, 2
ARG, 1990, 2
```

```

===== Python Script: CMSC
f = open("cmsc.txt","r")
print("Course No., Section No., Instructor, Seats, Open, Waitlist, Days, Time,
Bldg., Room No.")
line = f.readline().strip()
while line:
    course = line
    line = f.readline().strip()
    while line:
        out = []
        out.append(course)
        out.append(line)
        # out.append('' + f.readline().strip() + '')
        out.append(f.readline().strip())
        line = f.readline().strip().split(": ")
        out.append(line[1].split(",")[0])
        out.append(line[2].split(",")[0])
        out.append(line[3].split(" ")[0])
        line = f.readline().strip().split()
        out.append(line[0])
        out.append(' '.join(line[1:]))
        line = f.readline().strip().split()
        out.append(line[0])
        out.append(line[1])
        print(', '.join(map(str,out)))
        line = f.readline().strip()
    line = f.readline().strip()
f.close()

```

```

[serrintine@ApertureLabs:Lab4]$ python cmcsc.py
Course No., Section No., Instructor, Seats, Open, Waitlist, Days, Time, Bldg., Room No.
CMSC100, 0101, Charles Kassir, 45, 4, 0, M, 4:00pm - 4:50pm, CSI, 2117
CMSC106, 0101, Jianwu Wang, 45, 0, 5, TuTh, 9:30am - 10:45am, CSI, 2117
CMSC131, 0101, Evan Golub, 31, 0, 0, MWF, 2:00pm - 2:50pm, CSI, 1115
CMSC131, 0102, Evan Golub, 31, 0, 0, MWF, 2:00pm - 2:50pm, CSI, 1115
CMSC131, 0103, Evan Golub, 31, 1, 0, MWF, 2:00pm - 2:50pm, CSI, 1115
CMSC131, 0104, Evan Golub, 31, 1, 0, MWF, 2:00pm - 2:50pm, CSI, 1115
CMSC131, 0201, Evan Golub, 31, 1, 0, MWF, 3:00pm - 3:50pm, CSI, 1115
CMSC131, 0202, Evan Golub, 31, 1, 0, MWF, 3:00pm - 3:50pm, CSI, 1115
CMSC131, 0203, Evan Golub, 31, 0, 0, MWF, 3:00pm - 3:50pm, CSI, 1115
CMSC131, 0204, Evan Golub, 31, 0, 1, MWF, 3:00pm - 3:50pm, CSI, 1115
CMSC131, 0301, Thomas Reinhardt, 31, 0, 0, MWF, 11:00am - 11:50am, CSI, 1115
CMSC131, 0302, Thomas Reinhardt, 31, 0, 0, MWF, 11:00am - 11:50am, CSI, 1115
CMSC131, 0303, Thomas Reinhardt, 31, 0, 0, MWF, 11:00am - 11:50am, CSI, 1115
CMSC131, 0304, Thomas Reinhardt, 31, 0, 0, MWF, 11:00am - 11:50am, CSI, 1115
CMSC131, 0401, Thomas Reinhardt, 31, 0, 0, MWF, 12:00pm - 12:50pm, CSI, 1115
CMSC131, 0402, Thomas Reinhardt, 31, 7, 0, MWF, 12:00pm - 12:50pm, CSI, 1115
CMSC131, 0403, Thomas Reinhardt, 31, 0, 0, MWF, 12:00pm - 12:50pm, CSI, 1115
CMSC131, 0404, Thomas Reinhardt, 31, 7, 0, MWF, 12:00pm - 12:50pm, CSI, 1115
CMSC132, 0101, Laurence Herman, 34, 0, 2, MWF, 10:00am - 10:50am, CSI, 1115
CMSC132, 0102, Laurence Herman, 34, 0, 0, MWF, 10:00am - 10:50am, CSI, 1115
CMSC132, 0103, Laurence Herman, 34, 0, 0, MWF, 10:00am - 10:50am, CSI, 1115
CMSC132, 0104, Laurence Herman, 34, 0, 2, MWF, 10:00am - 10:50am, CSI, 1115
CMSC132, 0201, Laurence Herman, 34, 6, 0, MWF, 1:00pm - 1:50pm, CSI, 1115
CMSC132, 0202, Laurence Herman, 34, 1, 0, MWF, 1:00pm - 1:50pm, CSI, 1115
CMSC132, 0203, Laurence Herman, 34, 0, 0, MWF, 1:00pm - 1:50pm, CSI, 1115
CMSC132, 0204, Laurence Herman, 34, 0, 0, MWF, 1:00pm - 1:50pm, CSI, 1115
CMSC132, 0301, Laurence Herman, 30, 3, 0, MWF, 2:00pm - 2:50pm, CSI, 2117
CMSC132, 0302, Laurence Herman, 29, 0, 1, MWF, 2:00pm - 2:50pm, CSI, 2117
CMSC132, 0303, Laurence Herman, 29, 8, 0, MWF, 2:00pm - 2:50pm, CSI, 2117
CMSC216, 0101, Nelson Padua-Perez, 28, 2, 0, TuTh, 9:30am - 10:45am, CSI, 1115
CMSC216, 0102, Nelson Padua-Perez, 28, 0, 0, TuTh, 9:30am - 10:45am, CSI, 1115
CMSC216, 0103, Nelson Padua-Perez, 28, 0, 1, TuTh, 9:30am - 10:45am, CSI, 1115
CMSC216, 0104, Nelson Padua-Perez, 28, 0, 0, TuTh, 9:30am - 10:45am, CSI, 1115
CMSC216, 0201, Nelson Padua-Perez, 28, 0, 0, TuTh, 11:00am - 12:15pm, CSI, 1115
CMSC216, 0202, Nelson Padua-Perez, 28, 0, 1, TuTh, 11:00am - 12:15pm, CSI, 1115
CMSC216, 0203, Nelson Padua-Perez, 28, 0, 1, TuTh, 11:00am - 12:15pm, CSI, 1115
CMSC216, 0204, Nelson Padua-Perez, 28, 0, 0, TuTh, 11:00am - 12:15pm, CSI, 1115
CMSC216, 0301, Nelson Padua-Perez, 28, 0, 0, TuTh, 2:00pm - 3:15pm, CSI, 1115
CMSC216, 0302, Nelson Padua-Perez, 28, 2, 0, TuTh, 2:00pm - 3:15pm, CSI, 1115
CMSC216, 0303, Nelson Padua-Perez, 28, 0, 0, TuTh, 2:00pm - 3:15pm, CSI, 1115
CMSC216, 0304, Nelson Padua-Perez, 28, 0, 0, TuTh, 2:00pm - 3:15pm, CSI, 1115
CMSC250, 0101, Clyde Kruskal, 29, 0, 4, TuTh, 2:00pm - 3:15pm, CSI, 2117

```

===== Python Script: World Cup 1

```
import re
```

```
f = open("worldcup.txt","r")
print("Country, Year, Title")
f.readline()
f.readline()
line = f.readline().strip()
while line:
    for i in range(4):
        position = re.findall("\\d{4}]", f.readline().strip())
        for pos in position:
            out = []
            out.append(re.search(r"[A-Z]{3}", line).group(0))
            out.append(pos[1:-2])
            out.append(i+1)
            print(' ', '.join(map(str,out)))
        f.readline()
        f.readline()
        line = f.readline().strip()
f.close()
```

```
[serrintine@aperturelabs:Lab4]$ python worldcup1.py
```

```
Country, Year, Title
```

```
BRA, 1958, 1
BRA, 1962, 1
BRA, 1970, 1
BRA, 1994, 1
BRA, 2002, 1
BRA, 1950, 2
BRA, 1998, 2
BRA, 1938, 3
BRA, 1978, 3
BRA, 1974, 4
BRA, 2014, 4
GER, 1954, 1
GER, 1974, 1
GER, 1990, 1
GER, 2014, 1
GER, 1966, 2
GER, 1982, 2
GER, 1986, 2
GER, 2002, 2
GER, 1934, 3
GER, 1970, 3
GER, 2006, 3
GER, 2010, 3
GER, 1958, 4
ITA, 1934, 1
ITA, 1938, 1
ITA, 1982, 1
ITA, 2006, 1
ITA, 1970, 2
ITA, 1994, 2
ITA, 1990, 3
ITA, 1978, 4
ARG, 1978, 1
ARG, 1986, 1
ARG, 1930, 2
ARG, 1990, 2
ARG, 2014, 2
URU, 1930, 1
URU, 1950, 1
URU, 1954, 4
URU, 1970, 4
```

===== Python Script: World Cup 2

```
import sys
```

```
import pandas as pd
```

```
worldcup = pd.read_csv(sys.stdin, header=0, names=['Country', 'Year', 'Title'])
```

```
pivoted = worldcup.pivot(index='Country', columns='Year', values='Title')
```

```
print pivoted.to_string(na_rep='-', index_names=False)
```

```
[serrintine@aperturelabs:lab4]$ python worldcup1.py > worldcup1.csv
[serrintine@aperturelabs:lab4]$ python worldcup2.py < worldcup1.csv
```

	1930	1934	1938	1950	1954	1958	1962	1966	1970	1974	1978	1982	1986	1990	1994	1998	2002	2006	2010	2014
ARG	2	-	-	-	-	-	-	-	-	-	1	-	1	2	-	-	-	-	-	2
AUT	-	4	-	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BEL	-	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-	-	-	-	-
BRA	-	-	3	2	-	1	1	-	1	4	3	-	-	-	1	2	1	-	-	4
BUL	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-	-	-
CHI	-	-	-	-	-	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-
CRO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	-	-	-	-
ENG	-	-	-	-	-	-	-	1	-	-	-	-	-	4	-	-	-	-	-	-
ESP	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-
FRA	-	-	-	-	-	3	-	-	-	-	-	4	3	-	-	1	-	2	-	-
GER	-	3	-	-	1	4	-	2	3	1	-	2	2	1	-	-	2	3	3	1
HUN	-	-	2	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ITA	-	1	1	-	-	-	-	-	2	-	4	1	-	3	2	-	-	1	-	-
KOR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-
NED	-	-	-	-	-	-	-	-	-	2	2	-	-	-	-	4	-	-	2	3
POL	-	-	-	-	-	-	-	-	-	3	-	3	-	-	-	-	-	-	-	-
POR	-	-	-	-	-	-	-	3	-	-	-	-	-	-	-	-	-	4	-	-
SWE	-	-	4	3	-	2	-	-	-	-	-	-	-	3	-	-	-	-	-	-
TCH	-	2	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-
TUR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	-	-	-
URS	-	-	-	-	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-
URU	1	-	-	1	4	-	-	-	4	-	-	-	-	-	-	-	-	-	4	-
USA	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
YUG	4	-	-	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-

```
[serrintine@aperturelabs:lab4]$
```