Project 1

Amanda Fox

2024-02-21

Intro For this project, we were asked to take a highly formatted report of chess tournament scores and create a flat .csv file with selected stats by player for use in a database.

The report was not suitable in its initial format and extensive preparation was needed: each player's data was wrapped onto two rows with dashed lines separating them, along with inconsistent separators between fields, missing values, and other challenges. Additionally, one of the metrics requested was the average pre-chess rating of each player's opponents, which required joins and aggregation.

Importing the Data I began by reading in the text file as a vector to unwrap it using the scan function, specifying a pipe delimiter, removing extra white space, and skipping the first four rows (which followed a different pattern from the rest of the data).

Then I used rbind and split to force it into 23 columns per row to make my first dataframe of unwrapped, mostly-parsed data:

```
#--------READ IN FILE, UNWRAP ROWS, SPLIT INTO 23 FIELDS AND MAKE A DF

d<-"c:/users/amand/git_projects/DATA607/Project_1/data.txt"
v <- scan(file=d, sep="|",what='',skip=4,strip.white = TRUE)
unwrapped<-do.call(rbind, split(v, rep(1:(length(v) %/% 23), each=23)))
unwrapped<-as.data.frame(unwrapped)

head(v,40)
```

```
##  [1] "1"
##  [2] "GARY HUA"
##  [3] "6.0"
##  [4] "W  39"
##  [5] "W  21"
##  [6] "W  18"
##  [7] "W  14"
##  [8] "W   7"
##  [9] "D  12"
## [10] "D   4"
## [11] ""
## [12] "ON"
## [13] "15445895 / R: 1794   ->1817"
## [14] "N:2"
## [15] "W"
## [16] "B"
## [17] "W"
## [18] "B"
```

```
## [19] "W"
## [20] "B"
## [21] "W"
## [22] ""
## [23] "-----------------------------------------------------------------------------------------"
## [24] "2"
## [25] "DAKSHESH DARURI"
## [26] "6.0"
## [27] "W  63"
## [28] "W  58"
## [29] "L   4"
## [30] "W  17"
## [31] "W  16"
## [32] "W  20"
## [33] "W   7"
## [34] ""
## [35] "MI"
## [36] "14598900 / R: 1553   ->1663"
## [37] "N:2"
## [38] "B"
## [39] "W"
## [40] "B"
```

```
head(unwrapped)
```

```
##   V1                  V2  V3    V4    V5    V6    V7    V8    V9   V10 V11 V12
## 1  1           GARY HUA 6.0 W  39 W  21 W  18 W  14 W   7 D  12 D   4       ON
## 2  2     DAKSHESH DARURI 6.0 W  63 W  58 L   4 W  17 W  16 W  20 W   7       MI
## 3  3        ADITYA BAJAJ 6.0 L   8 W  61 W  25 W  21 W  11 W  13 W  12       MI
## 4  4 PATRICK H SCHILLING 5.5 W  23 D  28 W   2 W  26 D   5 W  19 D   1       MI
## 5  5          HANSHI ZUO 5.5 W  45 W  37 D  12 D  13 D   4 W  14 W  17       MI
## 6  6         HANSEN SONG 5.0 W  34 D  29 L  11 W  35 D  10 W  27 W  21       OH
##                          V13 V14 V15 V16 V17 V18 V19 V20 V21 V22
## 1 15445895 / R: 1794   ->1817 N:2   W   B   W   B   W   B   W
## 2 14598900 / R: 1553   ->1663 N:2   B   W   B   W   B   W   B
## 3 14959604 / R: 1384   ->1640 N:2   W   B   W   B   W   B   W
## 4 12616049 / R: 1716   ->1744 N:2   W   B   W   B   W   B   B
## 5 14601533 / R: 1655   ->1690 N:2   B   W   B   W   B   W   B
## 6 15055204 / R: 1686   ->1687 N:3   W   B   W   B   B   W   B
##                                                                                          V23
## 1 -----------------------------------------------------------------------------------------
## 2 -----------------------------------------------------------------------------------------
## 3 -----------------------------------------------------------------------------------------
## 4 -----------------------------------------------------------------------------------------
## 5 -----------------------------------------------------------------------------------------
## 6 -----------------------------------------------------------------------------------------
```

Cleanup I did some basic cleanup on this dataframe, removing blank columns and adding column names:

```
#--------DROP COLUMNS 11 & 22 (BLANK) AND 23 (DASHES), ADD COLUMN NAMES
unwrapped<-unwrapped[,1:21]
unwrapped<-unwrapped[,-11]
```

```
colnames(unwrapped)<-c("Pair","Player_Name","Points","Round_1","Round_2","Round_3","Round_4",
                       "Round_5","Round_6","Round_7","State","Player_ID_Rating","Not_Used1","Not_Used2"
                       "Not_Used3","Not_Used4","Not_Used5","Not_Used6","Not_Used7","Not_Used8")

head(unwrapped)
```

```
##   Pair         Player_Name Points Round_1 Round_2 Round_3 Round_4 Round_5
## 1    1            GARY HUA    6.0   W  39   W  21   W  18   W  14   W   7
## 2    2     DAKSHESH DARURI    6.0   W  63   W  58   L   4   W  17   W  16
## 3    3        ADITYA BAJAJ    6.0   L   8   W  61   W  25   W  21   W  11
## 4    4 PATRICK H SCHILLING    5.5   W  23   D  28   W   2   W  26   D   5
## 5    5          HANSHI ZUO    5.5   W  45   W  37   D  12   D  13   D   4
## 6    6         HANSEN SONG    5.0   W  34   D  29   L  11   W  35   D  10
##   Round_6 Round_7 State            Player_ID_Rating Not_Used1 Not_Used2
## 1   D  12   D   4    ON 15445895 / R: 1794   ->1817       N:2         W
## 2   W  20   W   7    MI 14598900 / R: 1553   ->1663       N:2         B
## 3   W  13   W  12    MI 14959604 / R: 1384   ->1640       N:2         W
## 4   W  19   D   1    MI 12616049 / R: 1716   ->1744       N:2         W
## 5   W  14   W  17    MI 14601533 / R: 1655   ->1690       N:2         B
## 6   W  27   W  21    OH 15055204 / R: 1686   ->1687       N:3         W
##   Not_Used3 Not_Used4 Not_Used5 Not_Used6 Not_Used7 Not_Used8
## 1         B         W         B         W         B         W
## 2         W         B         W         B         W         B
## 3         B         W         B         W         B         W
## 4         B         W         B         W         B         B
## 5         W         B         W         B         W         B
## 6         B         W         B         B         W         B
```

Separating Fields Many columns created by the pipe-delimited methodology above needed to be separated further. I used three different versions of the separate function to do so:

1. Rounds: Used "separate" to break into "WLD" and "Opponent". R was able to separate these pretty well, but I needed to handle two cases: I used "extra" to force R to not break double-digit player IDs into two fields, and "fill" to handle cases where there was no opponent.
2. Player ID and Rating Pre/Post: I used regex to separate these, which took more time than "separate" above but allowed me to be very specific.
3. Provisional Rating: Finally, I used delimiter = "P" to easily separate the provisional designations from ratings where they were present. This was necessary in order to convert ratings to numeric for aggregation later.

```
#--------SEPARATE ROUNDS DATA

unwrapped_sep<-
  unwrapped %>% separate(Round_1, c("Round_1_WLD","Round_1_Opponent"),extra="merge",fill="right")
unwrapped_sep<-
  unwrapped_sep %>% separate(Round_2, c("Round_2_WLD","Round_2_Opponent"),extra="merge",fill="right")
unwrapped_sep<-
  unwrapped_sep %>% separate(Round_3, c("Round_3_WLD","Round_3_Opponent"),extra="merge",fill="right")
unwrapped_sep<-
  unwrapped_sep %>% separate(Round_4, c("Round_4_WLD","Round_4_Opponent"),extra="merge",fill="right")
```

```
unwrapped_sep<-
  unwrapped_sep %>% separate(Round_5, c("Round_5_WLD","Round_5_Opponent"),extra="merge",fill="right")
unwrapped_sep<-
  unwrapped_sep %>% separate(Round_6, c("Round_6_WLD","Round_6_Opponent"),extra="merge",fill="right")
unwrapped_sep<-
  unwrapped_sep %>% separate(Round_7, c("Round_7_WLD","Round_7_Opponent"),extra="merge",fill="right")

#--------SEPARATE PLAYER ID AND RATING PRE- POST-

unwrapped_sep<- unwrapped_sep %>% separate_wider_regex(Player_ID_Rating, c(Player_ID = "^\\d.*","\\/\\s
unwrapped_sep<- unwrapped_sep %>% separate_wider_regex(misc, c(Rating_Pre = "^\\s*\\d*\\d*\\d*\\d*.*.*.

#--------SEPARATE PROVISIONAL DESIGNATIONS

unwrapped_sep<- unwrapped_sep %>% separate_wider_delim(Rating_Pre, delim="P",names=c("Rating_Pre","P_Pr
unwrapped_sep<- unwrapped_sep %>% separate_wider_delim(Rating_Post, delim="P",names=c("Rating_Post","P_

head(unwrapped_sep)
```

```
## # A tibble: 6 x 31
##   Pair  Player_Name         Points Round_1_WLD Round_1_Opponent Round_2_WLD
##   <chr> <chr>               <chr>  <chr>       <chr>            <chr>
## 1 1     GARY HUA            6.0    W           39               W
## 2 2     DAKSHESH DARURI     6.0    W           63               W
## 3 3     ADITYA BAJAJ        6.0    L           8                W
## 4 4     PATRICK H SCHILLING 5.5    W           23               D
## 5 5     HANSHI ZUO          5.5    W           45               W
## 6 6     HANSEN SONG         5.0    W           34               D
## # i 25 more variables: Round_2_Opponent <chr>, Round_3_WLD <chr>,
## #   Round_3_Opponent <chr>, Round_4_WLD <chr>, Round_4_Opponent <chr>,
## #   Round_5_WLD <chr>, Round_5_Opponent <chr>, Round_6_WLD <chr>,
## #   Round_6_Opponent <chr>, Round_7_WLD <chr>, Round_7_Opponent <chr>,
## #   State <chr>, Player_ID <chr>, Rating_Pre <chr>, P_Pre <chr>,
## #   Rating_Post <chr>, P_Post <chr>, Not_Used1 <chr>, Not_Used2 <chr>,
## #   Not_Used3 <chr>, Not_Used4 <chr>, Not_Used5 <chr>, Not_Used6 <chr>, ...
```

Numeric Fields After separating fields, I used transform to change five fields to data type numeric. This also got rid of any leading/trailing spaces left in the ratings columns by separating them above.

```
#--------MAKE POINTS AND RATINGS NUMERIC

unwrapped_sep_fin <- transform(unwrapped_sep,Points = as.numeric(Points))
unwrapped_sep_fin <- transform(unwrapped_sep_fin,Rating_Pre = as.numeric(Rating_Pre))
unwrapped_sep_fin <- transform(unwrapped_sep_fin,Rating_Post = as.numeric(Rating_Post))
unwrapped_sep_fin <- transform(unwrapped_sep_fin,P_Pre = as.numeric(P_Pre))
unwrapped_sep_fin <- transform(unwrapped_sep_fin,P_Post = as.numeric(P_Post))
```

Simplified Dataframes for Final Steps With all cleanup complete, I created a simplified dataframe with only columns required for the analysis. I also created a small dataframe of ratings by player to use as a lookup for the final step of adding opponents' average ratings.

After creating these final dataframes, I used str() to verify columns and data types before joining them:

```
#--------CREATE SIMPLIFIED DF

df <- unwrapped_sep_fin[,c("Pair","Player_ID","Player_Name","State","Points",
                           "Rating_Pre","Round_1_Opponent","Round_2_Opponent","Round_3_Opponent",
                           "Round_4_Opponent","Round_5_Opponent","Round_6_Opponent","Round_7_Opponent")]

# ---------MAKE RATINGS TABLE

df_ratings <- df[,c("Pair","Rating_Pre")]


str(df)
```

```
## 'data.frame':    64 obs. of  13 variables:
##  $ Pair            : chr  "1" "2" "3" "4" ...
##  $ Player_ID       : chr  "15445895 " "14598900 " "14959604 " "12616049 " ...
##  $ Player_Name     : chr  "GARY HUA" "DAKSHESH DARURI" "ADITYA BAJAJ" "PATRICK H SCHILLING" ...
##  $ State           : chr  "ON" "MI" "MI" "MI" ...
##  $ Points          : num  6 6 6 5.5 5.5 5 5 5 5 5 ...
##  $ Rating_Pre      : num  1794 1553 1384 1716 1655 ...
##  $ Round_1_Opponent: chr  "39" "63" "8" "23" ...
##  $ Round_2_Opponent: chr  "21" "58" "61" "28" ...
##  $ Round_3_Opponent: chr  "18" "4" "25" "2" ...
##  $ Round_4_Opponent: chr  "14" "17" "21" "26" ...
##  $ Round_5_Opponent: chr  "7" "16" "11" "5" ...
##  $ Round_6_Opponent: chr  "12" "20" "13" "19" ...
##  $ Round_7_Opponent: chr  "4" "7" "12" "1" ...
```

```
str(df_ratings)
```

```
## 'data.frame':    64 obs. of  2 variables:
##  $ Pair      : chr  "1" "2" "3" "4" ...
##  $ Rating_Pre: num  1794 1553 1384 1716 1655 ...
```

```
head(df)
```

```
##   Pair Player_ID         Player_Name State Points Rating_Pre Round_1_Opponent
## 1    1 15445895            GARY HUA    ON    6.0       1794               39
## 2    2 14598900      DAKSHESH DARURI    MI    6.0       1553               63
## 3    3 14959604         ADITYA BAJAJ    MI    6.0       1384                8
## 4    4 12616049  PATRICK H SCHILLING    MI    5.5       1716               23
## 5    5 14601533           HANSHI ZUO    MI    5.5       1655               45
## 6    6 15055204          HANSEN SONG    OH    5.0       1686               34
##   Round_2_Opponent Round_3_Opponent Round_4_Opponent Round_5_Opponent
## 1               21               18               14                7
## 2               58                4               17               16
## 3               61               25               21               11
## 4               28                2               26                5
## 5               37               12               13                4
## 6               29               11               35               10
##   Round_6_Opponent Round_7_Opponent
## 1               12                4
```

```
## 2               20                7
## 3               13               12
## 4               19                1
## 5               14               17
## 6               27               21
```

```
head(df_ratings)
```

```
##   Pair Rating_Pre
## 1    1       1794
## 2    2       1553
## 3    3       1384
## 4    4       1716
## 5    5       1655
## 6    6       1686
```

Add Opponents' Ratings For Each Player I used a left join between my two new simplified dataframes to add seven new columns: each player's opponents' ratings by Round. I then renamed the new columns and validated with str().

```
# ---------POPULATE OPP RATINGS IN DF

df <- df %>% left_join(df_ratings, join_by(x$Round_1_Opponent == y$Pair))
df <- df %>% left_join(df_ratings, join_by(x$Round_2_Opponent == y$Pair))
df <- df %>% left_join(df_ratings, join_by(x$Round_3_Opponent == y$Pair))
df <- df %>% left_join(df_ratings, join_by(x$Round_4_Opponent == y$Pair))
df <- df %>% left_join(df_ratings, join_by(x$Round_5_Opponent == y$Pair))
df <- df %>% left_join(df_ratings, join_by(x$Round_6_Opponent == y$Pair))
df <- df %>% left_join(df_ratings, join_by(x$Round_7_Opponent == y$Pair))


#---------RENAME COLUMNS

colnames(df)<-c("Pair","Player_ID","Player_Name","State","Points","Rating_Pre","Round_1_Opponent",
                "Round_2_Opponent","Round_3_Opponent","Round_4_Opponent","Round_5_Opponent","Round_6_Op
                "Round_7_Opponent","Round_1_Opp_Rate","Round_2_Opp_Rate","Round_3_Opp_Rate",
                "Round_4_Opp_Rate","Round_5_Opp_Rate","Round_6_Opp_Rate","Round_7_Opp_Rate")

#str(df)
head(df)
```

```
##   Pair Player_ID         Player_Name State Points Rating_Pre Round_1_Opponent
## 1    1 15445895            GARY HUA    ON    6.0       1794               39
## 2    2 14598900      DAKSHESH DARURI    MI    6.0       1553               63
## 3    3 14959604         ADITYA BAJAJ    MI    6.0       1384                8
## 4    4 12616049  PATRICK H SCHILLING    MI    5.5       1716               23
## 5    5 14601533           HANSHI ZUO    MI    5.5       1655               45
## 6    6 15055204          HANSEN SONG    OH    5.0       1686               34
##   Round_2_Opponent Round_3_Opponent Round_4_Opponent Round_5_Opponent
## 1               21               18               14                7
## 2               58                4               17               16
## 3               61               25               21               11
```

```
## 4                28                 2                26                 5
## 5                37                12                13                 4
## 6                29                11                35                10
##   Round_6_Opponent Round_7_Opponent Round_1_Opp_Rate Round_2_Opp_Rate
## 1               12                4             1436             1563
## 2               20                7             1175              917
## 3               13               12             1641              955
## 4               19                1             1363             1507
## 5               14               17             1242              980
## 6               27               21             1399             1602
##   Round_3_Opp_Rate Round_4_Opp_Rate Round_5_Opp_Rate Round_6_Opp_Rate
## 1             1600             1610             1649             1663
## 2             1716             1629             1604             1595
## 3             1745             1563             1712             1666
## 4             1553             1579             1655             1564
## 5             1663             1666             1716             1610
## 6             1712             1438             1365             1552
##   Round_7_Opp_Rate
## 1             1716
## 2             1649
## 3             1663
## 4             1794
## 5             1629
## 6             1563
```

Add Mean Opponents' Rating by Player Finally, the transform function was used to add the final column: the mean of each player's opponents' pre-chess ratings. This was a little challenging due to missing values: the na.rm argument was required to ignore (remove) the NA values for accurate means.

```
#---------ADD COL WITH AVERAGE OPP SCORES

df <- transform(df, Opp_Avg_Rate = round(rowMeans(df[,14:20], na.rm = TRUE)))

head(df)
```

```
##   Pair Player_ID         Player_Name State Points Rating_Pre Round_1_Opponent
## 1    1 15445895            GARY HUA    ON    6.0       1794               39
## 2    2 14598900      DAKSHESH DARURI    MI    6.0       1553               63
## 3    3 14959604         ADITYA BAJAJ    MI    6.0       1384                8
## 4    4 12616049  PATRICK H SCHILLING    MI    5.5       1716               23
## 5    5 14601533          HANSHI ZUO    MI    5.5       1655               45
## 6    6 15055204         HANSEN SONG    OH    5.0       1686               34
##   Round_2_Opponent Round_3_Opponent Round_4_Opponent Round_5_Opponent
## 1               21               18               14                7
## 2               58                4               17               16
## 3               61               25               21               11
## 4               28                2               26                5
## 5               37               12               13                4
## 6               29               11               35               10
##   Round_6_Opponent Round_7_Opponent Round_1_Opp_Rate Round_2_Opp_Rate
## 1               12                4             1436             1563
## 2               20                7             1175              917
## 3               13               12             1641              955
```

```
## 4               19                1             1363             1507
## 5               14               17             1242              980
## 6               27               21             1399             1602
##   Round_3_Opp_Rate Round_4_Opp_Rate Round_5_Opp_Rate Round_6_Opp_Rate
## 1             1600             1610             1649             1663
## 2             1716             1629             1604             1595
## 3             1745             1563             1712             1666
## 4             1553             1579             1655             1564
## 5             1663             1666             1716             1610
## 6             1712             1438             1365             1552
##   Round_7_Opp_Rate Opp_Avg_Rate
## 1             1716         1605
## 2             1649         1469
## 3             1663         1564
## 4             1794         1574
## 5             1629         1501
## 6             1563         1519
```

Final Output With all required columns now in the dataframe "df," I created the final output dataframe "chess_data" with only the columns required for this exercise and used write.csv to export it.

```
#---------MAKE FINAL SUMMARY TABLE AND GENERATE .CSV TO WORKING DIRECTORY

chess_data <- df [,c("Player_Name","State","Points","Rating_Pre","Opp_Avg_Rate")]
write.csv(chess_data,file='chess_data_output.csv', row.names=FALSE)

chess_data
```

```
##                   Player_Name State Points Rating_Pre Opp_Avg_Rate
## 1                    GARY HUA    ON    6.0       1794         1605
## 2              DAKSHESH DARURI    MI    6.0       1553         1469
## 3                 ADITYA BAJAJ    MI    6.0       1384         1564
## 4          PATRICK H SCHILLING    MI    5.5       1716         1574
## 5                   HANSHI ZUO    MI    5.5       1655         1501
## 6                  HANSEN SONG    OH    5.0       1686         1519
## 7            GARY DEE SWATHELL    MI    5.0       1649         1372
## 8             EZEKIEL HOUGHTON    MI    5.0       1641         1468
## 9                  STEFANO LEE    ON    5.0       1411         1523
## 10                   ANVIT RAO    MI    5.0       1365         1554
## 11   CAMERON WILLIAM MC LEMAN    MI    4.5       1712         1468
## 12              KENNETH J TACK    MI    4.5       1663         1506
## 13          TORRANCE HENRY JR    MI    4.5       1666         1498
## 14                BRADLEY SHAW    MI    4.5       1610         1515
## 15     ZACHARY JAMES HOUGHTON    MI    4.5       1220         1484
## 16                MIKE NIKITIN    MI    4.0       1604         1386
## 17         RONALD GRZEGORCZYK    MI    4.0       1629         1499
## 18               DAVID SUNDEEN    MI    4.0       1600         1480
## 19                DIPANKAR ROY    MI    4.0       1564         1426
## 20                JASON ZHENG    MI    4.0       1595         1411
## 21              DINH DANG BUI    ON    4.0       1563         1470
## 22            EUGENE L MCCLURE    MI    4.0       1555         1300
## 23                     ALAN BUI    ON    4.0       1363         1214
## 24          MICHAEL R ALDRICH    MI    4.0       1229         1357
```

```
## 25           LOREN SCHWIEBERT    MI    3.5       1745         1363
## 26                  MAX ZHU      ON    3.5       1579         1507
## 27             GAURAV GIDWANI    MI    3.5       1552         1222
## 28 SOFIA ADINA STANESCU-BELLU    MI    3.5       1507         1522
## 29           CHIEDOZIE OKORIE    MI    3.5       1602         1314
## 30         GEORGE AVERY JONES    ON    3.5       1522         1144
## 31               RISHI SHETTY    MI    3.5       1494         1260
## 32      JOSHUA PHILIP MATHEWS    ON    3.5       1441         1379
## 33                  JADE GE      MI    3.5       1449         1277
## 34     MICHAEL JEFFERY THOMAS    MI    3.5       1399         1375
## 35           JOSHUA DAVID LEE    MI    3.5       1438         1150
## 36             SIDDHARTH JHA     MI    3.5       1355         1388
## 37       AMIYATOSH PWNANANDAM    MI    3.5        980         1385
## 38                 BRIAN LIU     MI    3.0       1423         1539
## 39              JOEL R HENDON    MI    3.0       1436         1430
## 40               FOREST ZHANG    MI    3.0       1348         1391
## 41        KYLE WILLIAM MURPHY    MI    3.0       1403         1248
## 42                  JARED GE     MI    3.0       1332         1150
## 43          ROBERT GLEN VASEY    MI    3.0       1283         1107
## 44         JUSTIN D SCHILLING    MI    3.0       1199         1327
## 45                 DEREK YAN     MI    3.0       1242         1152
## 46   JACOB ALEXANDER LAVALLEY    MI    3.0        377         1358
## 47               ERIC WRIGHT     MI    2.5       1362         1392
## 48              DANIEL KHAIN     MI    2.5       1382         1356
## 49           MICHAEL J MARTIN    MI    2.5       1291         1286
## 50                 SHIVAM JHA    MI    2.5       1056         1296
## 51            TEJAS AYYAGARI     MI    2.5       1011         1356
## 52                 ETHAN GUO     MI    2.5        935         1495
## 53              JOSE C YBARRA    MI    2.0       1393         1345
## 54               LARRY HODGE     MI    2.0       1270         1206
## 55                 ALEX KONG     MI    2.0       1186         1406
## 56              MARISA RICCI     MI    2.0       1153         1414
## 57                MICHAEL LU     MI    2.0       1092         1363
## 58              VIRAJ MOHILE     MI    2.0        917         1391
## 59         SEAN M MC CORMICK     MI    2.0        853         1319
## 60                JULIA SHEN     MI    1.5        967         1330
## 61             JEZZEL FARKAS     ON    1.5        955         1327
## 62             ASHWIN BALAJI     MI    1.0       1530         1186
## 63      THOMAS JOSEPH HOSMER     MI    1.0       1175         1350
## 64                  BEN LI       MI    1.0       1163         1263
```