```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from xlogit import MixedLogit

In [3]: df1 = pd.read_csv(r"C:\Users\THW_9\Desktop\deskyop\Housing Research LKO\Qualtrics\Housing Policy Data\P.Housing_Merged_2(viii_ii)

df2 = pd.read_csv(r"C:\Users\THW_9\Desktop\deskyop\Housing Research LKO\Qualtrics\Housing Policy Data\HousingOptionsSummary.csv")
```

## Making the Dataset

## Running for post-policy

```
In [ ]: #This is the variable of interest.
         y_column = '4ChoiceRANK'
         #this is for post-policy changes
         post = 1
         #for option choices
         abcd = ['A','B','C','D']
abc = ['A','','B','C']
         rankt = [1,2,3,4]
         df1_to_run = df1
         df_new = pd.DataFrame()
         x = 0
         for i in range(0,len(df1_to_run)):
             print(i)
             for rank in rankt:
                 j = i + x #to index the dataframe
                 x += 1
                 df_new.at[j,'SN'] = df1_to_run.at[i,'SN']
                 y_column_choice = y_column + str(rank)
                 print(i, j, y_column_choice)
                 doub = df1_to_run.at[i, 'DualIncome'] > 0
                 sing = df1_to_run.at[i,'SingleIncome'] > 0
                 ineli = df1_to_run.at[i,'SingleIncome'] == 3
                 df_new.at[j,'Rank'] = rank
                 #Processing for doubles
                 if (doub and not sing):
                     income = df1_to_run.at[i,'DualIncome']
                     choice = abcd[
                         int(df1_to_run.at[i,y_column_choice]) - 1 #To identify chocies by alphabets
                     print(income, choice, post, ineli)
                     #to get attributes from df2
                     tempdf2 = df2.loc[
                         df2['DualIncome'] == income
                         ].loc[
                             df2['Option'] == choice
                             ].loc[
                                  df2['Post'] == post
                             ].reset_index()
                     for cols in list(df2.columns)[:8]:
                         df_new.at[j,cols] = tempdf2.at[0,cols] #copies and pastes the housing attribute into dataset
                     df_new.at[j,y_column_choice] = 1
                 #Repeated process for singles
                 elif sing and not doub:
                     income = df1_to_run.at[i,'SingleIncome']
```

```
if not (df1_to_run.at[i,y_column_choice] > 0):
               print('error')
               continue
            if not ineli:
               choice = abc[
                   int(df1_to_run.at[i,y_column_choice]) - 1
            else:
                choice = abcd[
                   int(df1_to_run.at[i,y_column_choice]) - 1
            print(income,choice,post)
            tempdf2 = df2.loc[
               df2['SingleIncome'] == income
                   df2['Option'] == choice
                   ].loc[
                        df2['Post'] == post
                   ].reset_index()
            for cols in list(df2.columns)[:8]:
                df_new.at[j,cols] = tempdf2.at[0,cols]
            df_new.at[j,y_column_choice] = 1
        else:
            print('error')
df_new = df_new.reset_index()
for i in range(0,len(df_new)):
   print(i, 'out of', len(df_new))
   df1_to_copy = df1_to_run.loc[df1_to_run['SN'] == df_new.at[i,'SN']].reset_index()
   for cols in df1_to_copy.columns[22:]:
        df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
# df_new.to_csv('P.Housing_Merged_2(ix)(forMixedLogitModel)(post-policy).csv')
```

```
0
0 0 4ChoiceRANK1
2.0 B 1 False
0 1 4ChoiceRANK2
2.0 D 1 False
0 2 4ChoiceRANK3
2.0 A 1 False
0 3 4ChoiceRANK4
2.0 C 1 False
1 5 4ChoiceRANK1
3.0 D 1
1 6 4ChoiceRANK2
3.0 B 1
1 7 4ChoiceRANK3
3.0 C 1
1 8 4ChoiceRANK4
3.0 A 1
2
2 10 4ChoiceRANK1
4.0 C 1 False
2 11 4ChoiceRANK2
4.0 A 1 False
2 12 4ChoiceRANK3
4.0 D 1 False
2 13 4ChoiceRANK4
4.0 B 1 False
3 15 4ChoiceRANK1
4.0 D 1 False
3 16 4ChoiceRANK2
4.0 C 1 False
3 17 4ChoiceRANK3
4.0 B 1 False
3 18 4ChoiceRANK4
4.0 A 1 False
4 20 4ChoiceRANK1
1.0 C 1 False
4 21 4ChoiceRANK2
1.0 A 1 False
4 22 4ChoiceRANK3
1.0 B 1 False
4 23 4ChoiceRANK4
1.0 D 1 False
5
5 25 4ChoiceRANK1
1.0 B 1 False
5 26 4ChoiceRANK2
1.0 D 1 False
5 27 4ChoiceRANK3
1.0 A 1 False
5 28 4ChoiceRANK4
1.0 C 1 False
6
6 30 4ChoiceRANK1
1.0 A 1 False
6 31 4ChoiceRANK2
1.0 C 1 False
6 32 4ChoiceRANK3
1.0 B 1 False
6 33 4ChoiceRANK4
1.0 D 1 False
7 35 4ChoiceRANK1
2.0 B 1
7 36 4ChoiceRANK2
2.0 C 1
7 37 4ChoiceRANK3
2.0 A 1
7 38 4ChoiceRANK4
error
8
8 40 4ChoiceRANK1
4.0 D 1 False
8 41 4ChoiceRANK2
4.0 C 1 False
8 42 4ChoiceRANK3
4.0 B 1 False
8 43 4ChoiceRANK4
4.0 A 1 False
9 45 4ChoiceRANK1
2.0 D 1 False
9 46 4ChoiceRANK2
2.0 C 1 False
9 47 4ChoiceRANK3
2.0 B 1 False
9 48 4ChoiceRANK4
2.0 A 1 False
10
10 50 4ChoiceRANK1
2.0 A 1 False
```

10 51 4ChoiceRANK2 2.0 B 1 False 10 52 4ChoiceRANK3 2.0 D 1 False 10 53 4ChoiceRANK4 2.0 C 1 False 11 11 55 4ChoiceRANK1 2.0 B 1 11 56 4ChoiceRANK2 2.0 A 1 11 57 4ChoiceRANK3 2.0 C 1 11 58 4ChoiceRANK4 error 12 12 60 4ChoiceRANK1 2.0 A 1 False 12 61 4ChoiceRANK2 2.0 C 1 False 12 62 4ChoiceRANK3 2.0 B 1 False 12 63 4ChoiceRANK4 2.0 D 1 False 13 65 4ChoiceRANK1 2.0 A 1 False 13 66 4ChoiceRANK2 2.0 C 1 False 13 67 4ChoiceRANK3 2.0 D 1 False 13 68 4ChoiceRANK4 2.0 B 1 False 14 14 70 4ChoiceRANK1 2.0 A 1 14 71 4ChoiceRANK2 2.0 C 1 14 72 4ChoiceRANK3 2.0 B 1 14 73 4ChoiceRANK4 error 15 15 75 4ChoiceRANK1 4.0 C 1 False 15 76 4ChoiceRANK2 4.0 D 1 False 15 77 4ChoiceRANK3 4.0 A 1 False 15 78 4ChoiceRANK4 4.0 B 1 False 16 16 80 4ChoiceRANK1 2.0 B 1 False 16 81 4ChoiceRANK2 2.0 D 1 False 16 82 4ChoiceRANK3 2.0 C 1 False 16 83 4ChoiceRANK4 2.0 A 1 False 17 17 85 4ChoiceRANK1 2.0 A 1 False 17 86 4ChoiceRANK2 2.0 B 1 False 17 87 4ChoiceRANK3 2.0 D 1 False 17 88 4ChoiceRANK4 2.0 C 1 False 18 90 4ChoiceRANK1 2.0 C 1 False 18 91 4ChoiceRANK2 2.0 D 1 False 18 92 4ChoiceRANK3 2.0 A 1 False 18 93 4ChoiceRANK4 2.0 B 1 False 19 19 95 4ChoiceRANK1 2.0 B 1 False 19 96 4ChoiceRANK2 2.0 A 1 False 19 97 4ChoiceRANK3 2.0 D 1 False 19 98 4ChoiceRANK4 2.0 C 1 False 20 20 100 4ChoiceRANK1 2.0 B 1 False 20 101 4ChoiceRANK2 2.0 D 1 False

20 102 4ChoiceRANK3

```
2.0 A 1 False
20 103 4ChoiceRANK4
2.0 C 1 False
21 105 4ChoiceRANK1
3.0 C 1
21 106 4ChoiceRANK2
3.0 B 1
21 107 4ChoiceRANK3
3.0 D 1
21 108 4ChoiceRANK4
3.0 A 1
22
22 110 4ChoiceRANK1
2.0 D 1 False
22 111 4ChoiceRANK2
2.0 B 1 False
22 112 4ChoiceRANK3
2.0 A 1 False
22 113 4ChoiceRANK4
2.0 C 1 False
23
23 115 4ChoiceRANK1
2.0 D 1 False
23 116 4ChoiceRANK2
2.0 C 1 False
23 117 4ChoiceRANK3
2.0 A 1 False
23 118 4ChoiceRANK4
2.0 B 1 False
24 120 4ChoiceRANK1
2.0 D 1 False
24 121 4ChoiceRANK2
2.0 B 1 False
24 122 4ChoiceRANK3
2.0 A 1 False
24 123 4ChoiceRANK4
2.0 C 1 False
25
25 125 4ChoiceRANK1
3.0 B 1
25 126 4ChoiceRANK2
3.0 C 1
25 127 4ChoiceRANK3
3.0 D 1
25 128 4ChoiceRANK4
3.0 A 1
26
26 130 4ChoiceRANK1
4.0 D 1 False
26 131 4ChoiceRANK2
4.0 B 1 False
26 132 4ChoiceRANK3
4.0 C 1 False
26 133 4ChoiceRANK4
4.0 A 1 False
27
27 135 4ChoiceRANK1
2.0 A 1 False
27 136 4ChoiceRANK2
2.0 C 1 False
27 137 4ChoiceRANK3
2.0 D 1 False
27 138 4ChoiceRANK4
2.0 B 1 False
28 140 4ChoiceRANK1
2.0 B 1
28 141 4ChoiceRANK2
2.0 A 1
28 142 4ChoiceRANK3
2.0 C 1
28 143 4ChoiceRANK4
error
29
29 145 4ChoiceRANK1
2.0 C 1
29 146 4ChoiceRANK2
2.0 B 1
29 147 4ChoiceRANK3
2.0 A 1
29 148 4ChoiceRANK4
error
30 150 4ChoiceRANK1
2.0 B 1
30 151 4ChoiceRANK2
2.0 C 1
30 152 4ChoiceRANK3
2.0 A 1
30 153 4ChoiceRANK4
```

error

```
31
31 155 4ChoiceRANK1
4.0 B 1 False
31 156 4ChoiceRANK2
4.0 A 1 False
31 157 4ChoiceRANK3
4.0 C 1 False
31 158 4ChoiceRANK4
4.0 D 1 False
32 160 4ChoiceRANK1
2.0 A 1 False
32 161 4ChoiceRANK2
2.0 B 1 False
32 162 4ChoiceRANK3
2.0 C 1 False
32 163 4ChoiceRANK4
2.0 D 1 False
33
33 165 4ChoiceRANK1
2.0 D 1 False
33 166 4ChoiceRANK2
2.0 C 1 False
33 167 4ChoiceRANK3
2.0 B 1 False
33 168 4ChoiceRANK4
2.0 A 1 False
34
34 170 4ChoiceRANK1
2.0 A 1 False
34 171 4ChoiceRANK2
2.0 B 1 False
34 172 4ChoiceRANK3
2.0 D 1 False
34 173 4ChoiceRANK4
2.0 C 1 False
35
35 175 4ChoiceRANK1
2.0 B 1
35 176 4ChoiceRANK2
2.0 A 1
35 177 4ChoiceRANK3
2.0 C 1
35 178 4ChoiceRANK4
error
36
36 180 4ChoiceRANK1
1.0 C 1 False
36 181 4ChoiceRANK2
1.0 D 1 False
36 182 4ChoiceRANK3
1.0 A 1 False
36 183 4ChoiceRANK4
1.0 B 1 False
37
37 185 4ChoiceRANK1
2.0 A 1
37 186 4ChoiceRANK2
2.0 B 1
37 187 4ChoiceRANK3
2.0 C 1
37 188 4ChoiceRANK4
error
38
38 190 4ChoiceRANK1
2.0 D 1 False
38 191 4ChoiceRANK2
2.0 C 1 False
38 192 4ChoiceRANK3
2.0 B 1 False
38 193 4ChoiceRANK4
2.0 A 1 False
39
39 195 4ChoiceRANK1
2.0 B 1 False
39 196 4ChoiceRANK2
2.0 A 1 False
39 197 4ChoiceRANK3
2.0 D 1 False
39 198 4ChoiceRANK4
2.0 C 1 False
40
40 200 4ChoiceRANK1
2.0 A 1 False
40 201 4ChoiceRANK2
2.0 B 1 False
40 202 4ChoiceRANK3
2.0 C 1 False
40 203 4ChoiceRANK4
2.0 D 1 False
41
41 205 4ChoiceRANK1
```

4.0 B 1 False

```
41 206 4ChoiceRANK2
4.0 A 1 False
41 207 4ChoiceRANK3
4.0 C 1 False
41 208 4ChoiceRANK4
4.0 D 1 False
42 210 4ChoiceRANK1
4.0 D 1 False
42 211 4ChoiceRANK2
4.0 B 1 False
42 212 4ChoiceRANK3
4.0 C 1 False
42 213 4ChoiceRANK4
4.0 A 1 False
43
43 215 4ChoiceRANK1
2.0 B 1 False
43 216 4ChoiceRANK2
2.0 A 1 False
43 217 4ChoiceRANK3
2.0 D 1 False
43 218 4ChoiceRANK4
2.0 C 1 False
44 220 4ChoiceRANK1
2.0 C 1 False
44 221 4ChoiceRANK2
2.0 A 1 False
44 222 4ChoiceRANK3
2.0 B 1 False
44 223 4ChoiceRANK4
2.0 D 1 False
45
45 225 4ChoiceRANK1
3.0 C 1
45 226 4ChoiceRANK2
3.0 D 1
45 227 4ChoiceRANK3
3.0 A 1
45 228 4ChoiceRANK4
3.0 B 1
46
46 230 4ChoiceRANK1
2.0 A 1 False
46 231 4ChoiceRANK2
2.0 D 1 False
46 232 4ChoiceRANK3
2.0 B 1 False
46 233 4ChoiceRANK4
2.0 C 1 False
47
47 235 4ChoiceRANK1
4.0 B 1 False
47 236 4ChoiceRANK2
4.0 D 1 False
47 237 4ChoiceRANK3
4.0 C 1 False
47 238 4ChoiceRANK4
4.0 A 1 False
48
48 240 4ChoiceRANK1
2.0 B 1 False
48 241 4ChoiceRANK2
2.0 A 1 False
48 242 4ChoiceRANK3
2.0 D 1 False
48 243 4ChoiceRANK4
2.0 C 1 False
49 245 4ChoiceRANK1
2.0 A 1 False
49 246 4ChoiceRANK2
2.0 C 1 False
49 247 4ChoiceRANK3
2.0 D 1 False
49 248 4ChoiceRANK4
2.0 B 1 False
50
50 250 4ChoiceRANK1
2.0 B 1 False
50 251 4ChoiceRANK2
2.0 A 1 False
50 252 4ChoiceRANK3
2.0 D 1 False
50 253 4ChoiceRANK4
2.0 C 1 False
51
51 255 4ChoiceRANK1
2.0 A 1 False
51 256 4ChoiceRANK2
2.0 C 1 False
```

51 257 4ChoiceRANK3

```
2.0 B 1 False
51 258 4ChoiceRANK4
2.0 D 1 False
52 260 4ChoiceRANK1
2.0 C 1
52 261 4ChoiceRANK2
2.0 B 1
52 262 4ChoiceRANK3
2.0 A 1
52 263 4ChoiceRANK4
error
53
53 265 4ChoiceRANK1
2.0 D 1 False
53 266 4ChoiceRANK2
2.0 B 1 False
53 267 4ChoiceRANK3
2.0 A 1 False
53 268 4ChoiceRANK4
2.0 C 1 False
54
54 270 4ChoiceRANK1
2.0 C 1 False
54 271 4ChoiceRANK2
2.0 A 1 False
54 272 4ChoiceRANK3
2.0 D 1 False
54 273 4ChoiceRANK4
2.0 B 1 False
55 275 4ChoiceRANK1
3.0 D 1
55 276 4ChoiceRANK2
3.0 B 1
55 277 4ChoiceRANK3
3.0 C 1
55 278 4ChoiceRANK4
3.0 A 1
56
56 280 4ChoiceRANK1
4.0 D 1 False
56 281 4ChoiceRANK2
4.0 B 1 False
56 282 4ChoiceRANK3
4.0 C 1 False
56 283 4ChoiceRANK4
4.0 A 1 False
57
57 285 4ChoiceRANK1
2.0 B 1 False
57 286 4ChoiceRANK2
2.0 A 1 False
57 287 4ChoiceRANK3
2.0 D 1 False
57 288 4ChoiceRANK4
2.0 C 1 False
58 290 4ChoiceRANK1
1.0 B 1 False
58 291 4ChoiceRANK2
1.0 D 1 False
58 292 4ChoiceRANK3
1.0 A 1 False
58 293 4ChoiceRANK4
1.0 C 1 False
59
59 295 4ChoiceRANK1
2.0 B 1 False
59 296 4ChoiceRANK2
2.0 D 1 False
59 297 4ChoiceRANK3
2.0 A 1 False
59 298 4ChoiceRANK4
2.0 C 1 False
60
60 300 4ChoiceRANK1
2.0 C 1 False
60 301 4ChoiceRANK2
2.0 B 1 False
60 302 4ChoiceRANK3
2.0 A 1 False
60 303 4ChoiceRANK4
2.0 D 1 False
61 305 4ChoiceRANK1
2.0 D 1 False
61 306 4ChoiceRANK2
2.0 B 1 False
61 307 4ChoiceRANK3
2.0 C 1 False
61 308 4ChoiceRANK4
2.0 A 1 False
```

```
62
62 310 4ChoiceRANK1
2.0 C 1
62 311 4ChoiceRANK2
2.0 B 1
62 312 4ChoiceRANK3
2.0 A 1
62 313 4ChoiceRANK4
63 315 4ChoiceRANK1
2.0 D 1 False
63 316 4ChoiceRANK2
2.0 C 1 False
63 317 4ChoiceRANK3
2.0 B 1 False
63 318 4ChoiceRANK4
2.0 A 1 False
64
64 320 4ChoiceRANK1
3.0 D 1
64 321 4ChoiceRANK2
3.0 B 1
64 322 4ChoiceRANK3
3.0 A 1
64 323 4ChoiceRANK4
3.0 C 1
65
65 325 4ChoiceRANK1
1.0 D 1 False
65 326 4ChoiceRANK2
1.0 A 1 False
65 327 4ChoiceRANK3
1.0 B 1 False
65 328 4ChoiceRANK4
1.0 C 1 False
66 330 4ChoiceRANK1
4.0 D 1 False
66 331 4ChoiceRANK2
4.0 C 1 False
66 332 4ChoiceRANK3
4.0 B 1 False
66 333 4ChoiceRANK4
4.0 A 1 False
67
67 335 4ChoiceRANK1
2.0 A 1 False
67 336 4ChoiceRANK2
2.0 C 1 False
67 337 4ChoiceRANK3
2.0 D 1 False
67 338 4ChoiceRANK4
2.0 B 1 False
68
68 340 4ChoiceRANK1
2.0 A 1
68 341 4ChoiceRANK2
2.0 B 1
68 342 4ChoiceRANK3
2.0 C 1
68 343 4ChoiceRANK4
error
69
69 345 4ChoiceRANK1
2.0 C 1
69 346 4ChoiceRANK2
2.0 B 1
69 347 4ChoiceRANK3
2.0 A 1
69 348 4ChoiceRANK4
error
70
70 350 4ChoiceRANK1
1.0 A 1 False
70 351 4ChoiceRANK2
1.0 C 1 False
70 352 4ChoiceRANK3
1.0 B 1 False
70 353 4ChoiceRANK4
1.0 D 1 False
71
71 355 4ChoiceRANK1
2.0 B 1 False
71 356 4ChoiceRANK2
2.0 D 1 False
71 357 4ChoiceRANK3
2.0 A 1 False
71 358 4ChoiceRANK4
2.0 C 1 False
72
72 360 4ChoiceRANK1
2.0 C 1 False
```

```
72 361 4ChoiceRANK2
2.0 A 1 False
72 362 4ChoiceRANK3
2.0 B 1 False
72 363 4ChoiceRANK4
2.0 D 1 False
73 365 4ChoiceRANK1
2.0 D 1 False
73 366 4ChoiceRANK2
2.0 C 1 False
73 367 4ChoiceRANK3
2.0 B 1 False
73 368 4ChoiceRANK4
2.0 A 1 False
74
74 370 4ChoiceRANK1
2.0 B 1 False
74 371 4ChoiceRANK2
2.0 D 1 False
74 372 4ChoiceRANK3
2.0 A 1 False
74 373 4ChoiceRANK4
2.0 C 1 False
75 375 4ChoiceRANK1
3.0 B 1
75 376 4ChoiceRANK2
3.0 D 1
75 377 4ChoiceRANK3
3.0 C 1
75 378 4ChoiceRANK4
3.0 A 1
76
76 380 4ChoiceRANK1
1.0 B 1
76 381 4ChoiceRANK2
1.0 A 1
76 382 4ChoiceRANK3
1.0 C 1
76 383 4ChoiceRANK4
error
77
77 385 4ChoiceRANK1
2.0 B 1 False
77 386 4ChoiceRANK2
2.0 A 1 False
77 387 4ChoiceRANK3
2.0 C 1 False
77 388 4ChoiceRANK4
2.0 D 1 False
78
78 390 4ChoiceRANK1
3.0 C 1
78 391 4ChoiceRANK2
3.0 B 1
78 392 4ChoiceRANK3
3.0 A 1
78 393 4ChoiceRANK4
3.0 D 1
79
79 395 4ChoiceRANK1
2.0 A 1
79 396 4ChoiceRANK2
2.0 B 1
79 397 4ChoiceRANK3
2.0 C 1
79 398 4ChoiceRANK4
error
80 400 4ChoiceRANK1
2.0 A 1 False
80 401 4ChoiceRANK2
2.0 B 1 False
80 402 4ChoiceRANK3
2.0 C 1 False
80 403 4ChoiceRANK4
2.0 D 1 False
81
81 405 4ChoiceRANK1
2.0 B 1 False
81 406 4ChoiceRANK2
2.0 A 1 False
81 407 4ChoiceRANK3
2.0 D 1 False
81 408 4ChoiceRANK4
2.0 C 1 False
82
82 410 4ChoiceRANK1
3.0 C 1
82 411 4ChoiceRANK2
3.0 A 1
```

82 412 4ChoiceRANK3

```
3.0 B 1
82 413 4ChoiceRANK4
3.0 D 1
83 415 4ChoiceRANK1
2.0 A 1 False
83 416 4ChoiceRANK2
2.0 B 1 False
83 417 4ChoiceRANK3
2.0 D 1 False
83 418 4ChoiceRANK4
2.0 C 1 False
84
84 420 4ChoiceRANK1
4.0 C 1 False
84 421 4ChoiceRANK2
4.0 A 1 False
84 422 4ChoiceRANK3
4.0 B 1 False
84 423 4ChoiceRANK4
4.0 D 1 False
85
85 425 4ChoiceRANK1
1.0 C 1 False
85 426 4ChoiceRANK2
1.0 A 1 False
85 427 4ChoiceRANK3
1.0 D 1 False
85 428 4ChoiceRANK4
1.0 B 1 False
86 430 4ChoiceRANK1
2.0 C 1
86 431 4ChoiceRANK2
2.0 A 1
86 432 4ChoiceRANK3
2.0 B 1
86 433 4ChoiceRANK4
error
87
87 435 4ChoiceRANK1
2.0 A 1
87 436 4ChoiceRANK2
2.0 C 1
87 437 4ChoiceRANK3
2.0 B 1
87 438 4ChoiceRANK4
error
88
88 440 4ChoiceRANK1
2.0 B 1 False
88 441 4ChoiceRANK2
2.0 A 1 False
88 442 4ChoiceRANK3
2.0 D 1 False
88 443 4ChoiceRANK4
2.0 C 1 False
89 445 4ChoiceRANK1
2.0 D 1 False
89 446 4ChoiceRANK2
2.0 C 1 False
89 447 4ChoiceRANK3
2.0 A 1 False
89 448 4ChoiceRANK4
2.0 B 1 False
90
90 450 4ChoiceRANK1
2.0 B 1 False
90 451 4ChoiceRANK2
2.0 D 1 False
90 452 4ChoiceRANK3
2.0 A 1 False
90 453 4ChoiceRANK4
2.0 C 1 False
91
91 455 4ChoiceRANK1
2.0 A 1 False
91 456 4ChoiceRANK2
2.0 B 1 False
91 457 4ChoiceRANK3
2.0 C 1 False
91 458 4ChoiceRANK4
2.0 D 1 False
92 460 4ChoiceRANK1
2.0 A 1
92 461 4ChoiceRANK2
2.0 B 1
92 462 4ChoiceRANK3
2.0 C 1
92 463 4ChoiceRANK4
```

error

```
93
93 465 4ChoiceRANK1
2.0 B 1 False
93 466 4ChoiceRANK2
2.0 A 1 False
93 467 4ChoiceRANK3
2.0 D 1 False
93 468 4ChoiceRANK4
2.0 C 1 False
94 470 4ChoiceRANK1
3.0 B 1
94 471 4ChoiceRANK2
3.0 A 1
94 472 4ChoiceRANK3
3.0 C 1
94 473 4ChoiceRANK4
3.0 D 1
95
95 475 4ChoiceRANK1
2.0 C 1 False
95 476 4ChoiceRANK2
2.0 A 1 False
95 477 4ChoiceRANK3
2.0 B 1 False
95 478 4ChoiceRANK4
2.0 D 1 False
96
96 480 4ChoiceRANK1
2.0 B 1 False
96 481 4ChoiceRANK2
2.0 D 1 False
96 482 4ChoiceRANK3
2.0 A 1 False
96 483 4ChoiceRANK4
2.0 C 1 False
97
97 485 4ChoiceRANK1
1.0 A 1
97 486 4ChoiceRANK2
1.0 B 1
97 487 4ChoiceRANK3
1.0 C 1
97 488 4ChoiceRANK4
error
98
98 490 4ChoiceRANK1
2.0 A 1 False
98 491 4ChoiceRANK2
2.0 B 1 False
98 492 4ChoiceRANK3
2.0 C 1 False
98 493 4ChoiceRANK4
2.0 D 1 False
99
99 495 4ChoiceRANK1
1.0 B 1 False
99 496 4ChoiceRANK2
1.0 A 1 False
99 497 4ChoiceRANK3
1.0 D 1 False
99 498 4ChoiceRANK4
1.0 C 1 False
100
100 500 4ChoiceRANK1
2.0 D 1 False
100 501 4ChoiceRANK2
2.0 B 1 False
100 502 4ChoiceRANK3
2.0 C 1 False
100 503 4ChoiceRANK4
2.0 A 1 False
101
101 505 4ChoiceRANK1
4.0 D 1 False
101 506 4ChoiceRANK2
4.0 B 1 False
101 507 4ChoiceRANK3
4.0 C 1 False
101 508 4ChoiceRANK4
4.0 A 1 False
102
102 510 4ChoiceRANK1
4.0 D 1 False
102 511 4ChoiceRANK2
4.0 C 1 False
102 512 4ChoiceRANK3
4.0 B 1 False
102 513 4ChoiceRANK4
4.0 A 1 False
103
103 515 4ChoiceRANK1
1.0 A 1 False
```

```
103 516 4ChoiceRANK2
1.0 B 1 False
103 517 4ChoiceRANK3
1.0 C 1 False
103 518 4ChoiceRANK4
1.0 D 1 False
104 520 4ChoiceRANK1
2.0 C 1 False
104 521 4ChoiceRANK2
2.0 A 1 False
104 522 4ChoiceRANK3
2.0 D 1 False
104 523 4ChoiceRANK4
2.0 B 1 False
105
105 525 4ChoiceRANK1
2.0 B 1 False
105 526 4ChoiceRANK2
2.0 D 1 False
105 527 4ChoiceRANK3
2.0 C 1 False
105 528 4ChoiceRANK4
2.0 A 1 False
106
106 530 4ChoiceRANK1
2.0 B 1
106 531 4ChoiceRANK2
2.0 A 1
106 532 4ChoiceRANK3
2.0 C 1
106 533 4ChoiceRANK4
error
107
107 535 4ChoiceRANK1
2.0 B 1 False
107 536 4ChoiceRANK2
2.0 A 1 False
107 537 4ChoiceRANK3
2.0 D 1 False
107 538 4ChoiceRANK4
2.0 C 1 False
108
108 540 4ChoiceRANK1
2.0 A 1
108 541 4ChoiceRANK2
2.0 B 1
108 542 4ChoiceRANK3
2.0 C 1
108 543 4ChoiceRANK4
error
109
109 545 4ChoiceRANK1
2.0 C 1 False
109 546 4ChoiceRANK2
2.0 D 1 False
109 547 4ChoiceRANK3
2.0 A 1 False
109 548 4ChoiceRANK4
2.0 B 1 False
110
110 550 4ChoiceRANK1
2.0 C 1
110 551 4ChoiceRANK2
2.0 B 1
110 552 4ChoiceRANK3
2.0 A 1
110 553 4ChoiceRANK4
error
111 555 4ChoiceRANK1
2.0 A 1 False
111 556 4ChoiceRANK2
2.0 C 1 False
111 557 4ChoiceRANK3
2.0 B 1 False
111 558 4ChoiceRANK4
2.0 D 1 False
112
112 560 4ChoiceRANK1
3.0 A 1
112 561 4ChoiceRANK2
3.0 B 1
112 562 4ChoiceRANK3
3.0 C 1
112 563 4ChoiceRANK4
3.0 D 1
113
113 565 4ChoiceRANK1
2.0 A 1 False
113 566 4ChoiceRANK2
2.0 B 1 False
```

113 567 4ChoiceRANK3

```
2.0 D 1 False
113 568 4ChoiceRANK4
2.0 C 1 False
114
114 570 4ChoiceRANK1
2.0 A 1 False
114 571 4ChoiceRANK2
2.0 B 1 False
114 572 4ChoiceRANK3
2.0 C 1 False
114 573 4ChoiceRANK4
2.0 D 1 False
115
115 575 4ChoiceRANK1
2.0 C 1
115 576 4ChoiceRANK2
2.0 B 1
115 577 4ChoiceRANK3
2.0 A 1
115 578 4ChoiceRANK4
error
116
116 580 4ChoiceRANK1
2.0 B 1 False
116 581 4ChoiceRANK2
2.0 C 1 False
116 582 4ChoiceRANK3
2.0 A 1 False
116 583 4ChoiceRANK4
2.0 D 1 False
117
117 585 4ChoiceRANK1
3.0 B 1 False
117 586 4ChoiceRANK2
3.0 D 1 False
117 587 4ChoiceRANK3
3.0 A 1 False
117 588 4ChoiceRANK4
3.0 C 1 False
118
118 590 4ChoiceRANK1
3.0 D 1 False
118 591 4ChoiceRANK2
3.0 B 1 False
118 592 4ChoiceRANK3
3.0 C 1 False
118 593 4ChoiceRANK4
3.0 A 1 False
119
119 595 4ChoiceRANK1
2.0 A 1 False
119 596 4ChoiceRANK2
2.0 C 1 False
119 597 4ChoiceRANK3
2.0 B 1 False
119 598 4ChoiceRANK4
2.0 D 1 False
120
120 600 4ChoiceRANK1
3.0 B 1 False
120 601 4ChoiceRANK2
3.0 A 1 False
120 602 4ChoiceRANK3
3.0 D 1 False
120 603 4ChoiceRANK4
3.0 C 1 False
121
121 605 4ChoiceRANK1
2.0 C 1
121 606 4ChoiceRANK2
2.0 B 1
121 607 4ChoiceRANK3
2.0 A 1
121 608 4ChoiceRANK4
error
122
122 610 4ChoiceRANK1
3.0 C 1
122 611 4ChoiceRANK2
3.0 D 1
122 612 4ChoiceRANK3
3.0 A 1
122 613 4ChoiceRANK4
3.0 B 1
123
123 615 4ChoiceRANK1
2.0 B 1 False
123 616 4ChoiceRANK2
2.0 A 1 False
123 617 4ChoiceRANK3
2.0 D 1 False
```

123 618 4ChoiceRANK4

2.0 C 1 False

```
124
124 620 4ChoiceRANK1
2.0 B 1
124 621 4ChoiceRANK2
2.0 C 1
124 622 4ChoiceRANK3
2.0 A 1
124 623 4ChoiceRANK4
error
125
125 625 4ChoiceRANK1
3.0 A 1 False
125 626 4ChoiceRANK2
3.0 B 1 False
125 627 4ChoiceRANK3
3.0 D 1 False
125 628 4ChoiceRANK4
3.0 C 1 False
126
126 630 4ChoiceRANK1
3.0 A 1 False
126 631 4ChoiceRANK2
3.0 D 1 False
126 632 4ChoiceRANK3
3.0 C 1 False
126 633 4ChoiceRANK4
3.0 B 1 False
127
127 635 4ChoiceRANK1
3.0 B 1 False
127 636 4ChoiceRANK2
3.0 A 1 False
127 637 4ChoiceRANK3
3.0 D 1 False
127 638 4ChoiceRANK4
3.0 C 1 False
128
128 640 4ChoiceRANK1
2.0 A 1 False
128 641 4ChoiceRANK2
2.0 D 1 False
128 642 4ChoiceRANK3
2.0 C 1 False
128 643 4ChoiceRANK4
2.0 B 1 False
129
129 645 4ChoiceRANK1
2.0 C 1
129 646 4ChoiceRANK2
2.0 B 1
129 647 4ChoiceRANK3
2.0 A 1
129 648 4ChoiceRANK4
error
130
130 650 4ChoiceRANK1
2.0 B 1 False
130 651 4ChoiceRANK2
2.0 D 1 False
130 652 4ChoiceRANK3
2.0 A 1 False
130 653 4ChoiceRANK4
2.0 C 1 False
131
131 655 4ChoiceRANK1
3.0 C 1 False
131 656 4ChoiceRANK2
3.0 D 1 False
131 657 4ChoiceRANK3
3.0 B 1 False
131 658 4ChoiceRANK4
3.0 A 1 False
132
132 660 4ChoiceRANK1
3.0 A 1 False
132 661 4ChoiceRANK2
3.0 B 1 False
132 662 4ChoiceRANK3
3.0 C 1 False
132 663 4ChoiceRANK4
3.0 D 1 False
133
133 665 4ChoiceRANK1
2.0 D 1 False
133 666 4ChoiceRANK2
2.0 A 1 False
133 667 4ChoiceRANK3
2.0 C 1 False
133 668 4ChoiceRANK4
2.0 B 1 False
134
```

134 670 4ChoiceRANK1

1.0 A 1

```
134 671 4ChoiceRANK2
1.0 C 1
134 672 4ChoiceRANK3
1.0 B 1
134 673 4ChoiceRANK4
error
135
135 675 4ChoiceRANK1
2.0 D 1 False
135 676 4ChoiceRANK2
2.0 A 1 False
135 677 4ChoiceRANK3
2.0 B 1 False
135 678 4ChoiceRANK4
2.0 C 1 False
136
136 680 4ChoiceRANK1
4.0 A 1 False
136 681 4ChoiceRANK2
4.0 B 1 False
136 682 4ChoiceRANK3
4.0 C 1 False
136 683 4ChoiceRANK4
4.0 D 1 False
137
137 685 4ChoiceRANK1
3.0 A 1 False
137 686 4ChoiceRANK2
3.0 C 1 False
137 687 4ChoiceRANK3
3.0 D 1 False
137 688 4ChoiceRANK4
3.0 B 1 False
138
138 690 4ChoiceRANK1
2.0 B 1 False
138 691 4ChoiceRANK2
2.0 D 1 False
138 692 4ChoiceRANK3
2.0 A 1 False
138 693 4ChoiceRANK4
2.0 C 1 False
139
139 695 4ChoiceRANK1
3.0 B 1 False
139 696 4ChoiceRANK2
3.0 D 1 False
139 697 4ChoiceRANK3
3.0 A 1 False
139 698 4ChoiceRANK4
3.0 C 1 False
140
140 700 4ChoiceRANK1
3.0 D 1
140 701 4ChoiceRANK2
3.0 C 1
140 702 4ChoiceRANK3
3.0 B 1
140 703 4ChoiceRANK4
3.0 A 1
141
141 705 4ChoiceRANK1
2.0 A 1 False
141 706 4ChoiceRANK2
2.0 B 1 False
141 707 4ChoiceRANK3
2.0 C 1 False
141 708 4ChoiceRANK4
2.0 D 1 False
142 710 4ChoiceRANK1
2.0 D 1 False
142 711 4ChoiceRANK2
2.0 B 1 False
142 712 4ChoiceRANK3
2.0 A 1 False
142 713 4ChoiceRANK4
2.0 C 1 False
143
143 715 4ChoiceRANK1
3.0 A 1 False
143 716 4ChoiceRANK2
3.0 B 1 False
143 717 4ChoiceRANK3
3.0 C 1 False
143 718 4ChoiceRANK4
3.0 D 1 False
144
144 720 4ChoiceRANK1
3.0 D 1 False
144 721 4ChoiceRANK2
3.0 B 1 False
144 722 4ChoiceRANK3
```

```
3.0 A 1 False
144 723 4ChoiceRANK4
3.0 C 1 False
145 725 4ChoiceRANK1
3.0 A 1 False
145 726 4ChoiceRANK2
3.0 B 1 False
145 727 4ChoiceRANK3
3.0 D 1 False
145 728 4ChoiceRANK4
3.0 C 1 False
146
146 730 4ChoiceRANK1
3.0 D 1
146 731 4ChoiceRANK2
3.0 B 1
146 732 4ChoiceRANK3
3.0 C 1
146 733 4ChoiceRANK4
3.0 A 1
147
147 735 4ChoiceRANK1
3.0 B 1 False
147 736 4ChoiceRANK2
3.0 D 1 False
147 737 4ChoiceRANK3
3.0 A 1 False
147 738 4ChoiceRANK4
3.0 C 1 False
148
148 740 4ChoiceRANK1
2.0 B 1 False
148 741 4ChoiceRANK2
2.0 A 1 False
148 742 4ChoiceRANK3
2.0 D 1 False
148 743 4ChoiceRANK4
2.0 C 1 False
149
149 745 4ChoiceRANK1
4.0 D 1 False
149 746 4ChoiceRANK2
4.0 C 1 False
149 747 4ChoiceRANK3
4.0 B 1 False
149 748 4ChoiceRANK4
4.0 A 1 False
150
150 750 4ChoiceRANK1
4.0 D 1 False
150 751 4ChoiceRANK2
4.0 C 1 False
150 752 4ChoiceRANK3
4.0 B 1 False
150 753 4ChoiceRANK4
4.0 A 1 False
151
151 755 4ChoiceRANK1
2.0 C 1
151 756 4ChoiceRANK2
2.0 B 1
151 757 4ChoiceRANK3
2.0 A 1
151 758 4ChoiceRANK4
error
152
152 760 4ChoiceRANK1
2.0 B 1 False
152 761 4ChoiceRANK2
2.0 D 1 False
152 762 4ChoiceRANK3
2.0 A 1 False
152 763 4ChoiceRANK4
2.0 C 1 False
153
153 765 4ChoiceRANK1
1.0 A 1 False
153 766 4ChoiceRANK2
1.0 D 1 False
153 767 4ChoiceRANK3
1.0 B 1 False
153 768 4ChoiceRANK4
1.0 C 1 False
154
154 770 4ChoiceRANK1
2.0 B 1 False
154 771 4ChoiceRANK2
2.0 A 1 False
154 772 4ChoiceRANK3
2.0 D 1 False
```

154 773 4ChoiceRANK4 2.0 C 1 False

```
155
155 775 4ChoiceRANK1
2.0 B 1 False
155 776 4ChoiceRANK2
2.0 A 1 False
155 777 4ChoiceRANK3
2.0 D 1 False
155 778 4ChoiceRANK4
2.0 C 1 False
156
156 780 4ChoiceRANK1
3.0 C 1
156 781 4ChoiceRANK2
3.0 D 1
156 782 4ChoiceRANK3
3.0 A 1
156 783 4ChoiceRANK4
3.0 B 1
157
157 785 4ChoiceRANK1
3.0 B 1 False
157 786 4ChoiceRANK2
3.0 D 1 False
157 787 4ChoiceRANK3
3.0 A 1 False
157 788 4ChoiceRANK4
3.0 C 1 False
158
158 790 4ChoiceRANK1
3.0 A 1 False
158 791 4ChoiceRANK2
3.0 B 1 False
158 792 4ChoiceRANK3
3.0 C 1 False
158 793 4ChoiceRANK4
3.0 D 1 False
159
159 795 4ChoiceRANK1
3.0 A 1 False
159 796 4ChoiceRANK2
3.0 B 1 False
159 797 4ChoiceRANK3
3.0 D 1 False
159 798 4ChoiceRANK4
3.0 C 1 False
160
160 800 4ChoiceRANK1
2.0 A 1 False
160 801 4ChoiceRANK2
2.0 D 1 False
160 802 4ChoiceRANK3
2.0 B 1 False
160 803 4ChoiceRANK4
2.0 C 1 False
161
161 805 4ChoiceRANK1
2.0 B 1 False
161 806 4ChoiceRANK2
2.0 A 1 False
161 807 4ChoiceRANK3
2.0 C 1 False
161 808 4ChoiceRANK4
2.0 D 1 False
162
162 810 4ChoiceRANK1
1.0 B 1
162 811 4ChoiceRANK2
1.0 A 1
162 812 4ChoiceRANK3
1.0 C 1
162 813 4ChoiceRANK4
error
163
163 815 4ChoiceRANK1
3.0 A 1 False
163 816 4ChoiceRANK2
3.0 B 1 False
163 817 4ChoiceRANK3
3.0 D 1 False
163 818 4ChoiceRANK4
3.0 C 1 False
164
164 820 4ChoiceRANK1
2.0 A 1 False
164 821 4ChoiceRANK2
2.0 C 1 False
164 822 4ChoiceRANK3
2.0 B 1 False
164 823 4ChoiceRANK4
2.0 D 1 False
165
```

165 825 4ChoiceRANK1

3.0 A 1

```
165 826 4ChoiceRANK2
3.0 D 1
165 827 4ChoiceRANK3
3.0 B 1
165 828 4ChoiceRANK4
3.0 C 1
166
166 830 4ChoiceRANK1
2.0 B 1 False
166 831 4ChoiceRANK2
2.0 A 1 False
166 832 4ChoiceRANK3
2.0 D 1 False
166 833 4ChoiceRANK4
2.0 C 1 False
167
167 835 4ChoiceRANK1
2.0 A 1 False
167 836 4ChoiceRANK2
2.0 B 1 False
167 837 4ChoiceRANK3
2.0 C 1 False
167 838 4ChoiceRANK4
2.0 D 1 False
168
168 840 4ChoiceRANK1
1.0 B 1 False
168 841 4ChoiceRANK2
1.0 A 1 False
168 842 4ChoiceRANK3
1.0 C 1 False
168 843 4ChoiceRANK4
1.0 D 1 False
169
169 845 4ChoiceRANK1
3.0 A 1 False
169 846 4ChoiceRANK2
3.0 B 1 False
169 847 4ChoiceRANK3
3.0 D 1 False
169 848 4ChoiceRANK4
3.0 C 1 False
170
170 850 4ChoiceRANK1
2.0 B 1 False
170 851 4ChoiceRANK2
2.0 A 1 False
170 852 4ChoiceRANK3
2.0 C 1 False
170 853 4ChoiceRANK4
2.0 D 1 False
171
171 855 4ChoiceRANK1
3.0 A 1 False
171 856 4ChoiceRANK2
3.0 B 1 False
171 857 4ChoiceRANK3
3.0 C 1 False
171 858 4ChoiceRANK4
3.0 D 1 False
172
172 860 4ChoiceRANK1
2.0 C 1
172 861 4ChoiceRANK2
2.0 A 1
172 862 4ChoiceRANK3
2.0 B 1
172 863 4ChoiceRANK4
error
173 865 4ChoiceRANK1
4.0 A 1 False
173 866 4ChoiceRANK2
4.0 B 1 False
173 867 4ChoiceRANK3
4.0 C 1 False
173 868 4ChoiceRANK4
4.0 D 1 False
174
174 870 4ChoiceRANK1
2.0 A 1 False
174 871 4ChoiceRANK2
2.0 B 1 False
174 872 4ChoiceRANK3
2.0 D 1 False
174 873 4ChoiceRANK4
2.0 C 1 False
175
175 875 4ChoiceRANK1
3.0 A 1 False
175 876 4ChoiceRANK2
```

3.0 B 1 False

175 877 4ChoiceRANK3

```
3.0 D 1 False
175 878 4ChoiceRANK4
3.0 C 1 False
176
176 880 4ChoiceRANK1
3.0 A 1 False
176 881 4ChoiceRANK2
3.0 C 1 False
176 882 4ChoiceRANK3
3.0 D 1 False
176 883 4ChoiceRANK4
3.0 B 1 False
177
177 885 4ChoiceRANK1
2.0 D 1 False
177 886 4ChoiceRANK2
2.0 C 1 False
177 887 4ChoiceRANK3
2.0 A 1 False
177 888 4ChoiceRANK4
2.0 B 1 False
178
178 890 4ChoiceRANK1
2.0 C 1
178 891 4ChoiceRANK2
2.0 B 1
178 892 4ChoiceRANK3
2.0 A 1
178 893 4ChoiceRANK4
error
179
179 895 4ChoiceRANK1
1.0 B 1
179 896 4ChoiceRANK2
1.0 C 1
179 897 4ChoiceRANK3
1.0 A 1
179 898 4ChoiceRANK4
error
180
180 900 4ChoiceRANK1
2.0 A 1
180 901 4ChoiceRANK2
2.0 B 1
180 902 4ChoiceRANK3
2.0 C 1
180 903 4ChoiceRANK4
error
181
181 905 4ChoiceRANK1
2.0 A 1 False
181 906 4ChoiceRANK2
2.0 B 1 False
181 907 4ChoiceRANK3
2.0 C 1 False
181 908 4ChoiceRANK4
2.0 D 1 False
182
182 910 4ChoiceRANK1
2.0 A 1 False
182 911 4ChoiceRANK2
2.0 B 1 False
182 912 4ChoiceRANK3
2.0 D 1 False
182 913 4ChoiceRANK4
2.0 C 1 False
183
183 915 4ChoiceRANK1
2.0 A 1 False
183 916 4ChoiceRANK2
2.0 D 1 False
183 917 4ChoiceRANK3
2.0 C 1 False
183 918 4ChoiceRANK4
2.0 B 1 False
184
184 920 4ChoiceRANK1
3.0 B 1
184 921 4ChoiceRANK2
3.0 A 1
184 922 4ChoiceRANK3
3.0 D 1
184 923 4ChoiceRANK4
3.0 C 1
185
185 925 4ChoiceRANK1
2.0 B 1 False
185 926 4ChoiceRANK2
2.0 A 1 False
185 927 4ChoiceRANK3
2.0 C 1 False
```

185 927 4ChoiceRANK3
2.0 C 1 False
185 928 4ChoiceRANK4
2.0 D 1 False

```
186
186 930 4ChoiceRANK1
2.0 B 1 False
186 931 4ChoiceRANK2
2.0 A 1 False
186 932 4ChoiceRANK3
2.0 D 1 False
186 933 4ChoiceRANK4
2.0 C 1 False
187 935 4ChoiceRANK1
2.0 C 1
187 936 4ChoiceRANK2
2.0 B 1
187 937 4ChoiceRANK3
2.0 A 1
187 938 4ChoiceRANK4
error
188
188 940 4ChoiceRANK1
3.0 B 1 False
188 941 4ChoiceRANK2
3.0 A 1 False
188 942 4ChoiceRANK3
3.0 D 1 False
188 943 4ChoiceRANK4
3.0 C 1 False
189
189 945 4ChoiceRANK1
2.0 D 1 False
189 946 4ChoiceRANK2
2.0 B 1 False
189 947 4ChoiceRANK3
2.0 A 1 False
189 948 4ChoiceRANK4
2.0 C 1 False
190
190 950 4ChoiceRANK1
2.0 A 1 False
190 951 4ChoiceRANK2
2.0 B 1 False
190 952 4ChoiceRANK3
2.0 C 1 False
190 953 4ChoiceRANK4
2.0 D 1 False
191
191 955 4ChoiceRANK1
3.0 C 1 False
191 956 4ChoiceRANK2
3.0 A 1 False
191 957 4ChoiceRANK3
3.0 B 1 False
191 958 4ChoiceRANK4
3.0 D 1 False
192
192 960 4ChoiceRANK1
2.0 A 1 False
192 961 4ChoiceRANK2
2.0 B 1 False
192 962 4ChoiceRANK3
2.0 C 1 False
192 963 4ChoiceRANK4
2.0 D 1 False
193
193 965 4ChoiceRANK1
2.0 C 1
193 966 4ChoiceRANK2
2.0 B 1
193 967 4ChoiceRANK3
2.0 A 1
193 968 4ChoiceRANK4
error
194
194 970 4ChoiceRANK1
3.0 A 1 False
194 971 4ChoiceRANK2
3.0 C 1 False
194 972 4ChoiceRANK3
3.0 B 1 False
194 973 4ChoiceRANK4
3.0 D 1 False
195
195 975 4ChoiceRANK1
3.0 B 1 False
195 976 4ChoiceRANK2
3.0 D 1 False
195 977 4ChoiceRANK3
3.0 A 1 False
195 978 4ChoiceRANK4
3.0 C 1 False
196
```

196 980 4ChoiceRANK1

1.0 B 1 False

```
196 981 4ChoiceRANK2
1.0 A 1 False
196 982 4ChoiceRANK3
1.0 D 1 False
196 983 4ChoiceRANK4
1.0 C 1 False
197
197 985 4ChoiceRANK1
2.0 D 1 False
197 986 4ChoiceRANK2
2.0 B 1 False
197 987 4ChoiceRANK3
2.0 A 1 False
197 988 4ChoiceRANK4
2.0 C 1 False
198
198 990 4ChoiceRANK1
4.0 D 1 False
198 991 4ChoiceRANK2
4.0 C 1 False
198 992 4ChoiceRANK3
4.0 B 1 False
198 993 4ChoiceRANK4
4.0 A 1 False
199 995 4ChoiceRANK1
3.0 A 1 False
199 996 4ChoiceRANK2
3.0 B 1 False
199 997 4ChoiceRANK3
3.0 C 1 False
199 998 4ChoiceRANK4
3.0 D 1 False
200
200 1000 4ChoiceRANK1
3.0 B 1 False
200 1001 4ChoiceRANK2
3.0 A 1 False
200 1002 4ChoiceRANK3
3.0 D 1 False
200 1003 4ChoiceRANK4
3.0 C 1 False
201
201 1005 4ChoiceRANK1
2.0 C 1 False
201 1006 4ChoiceRANK2
2.0 D 1 False
201 1007 4ChoiceRANK3
2.0 B 1 False
201 1008 4ChoiceRANK4
2.0 A 1 False
202
202 1010 4ChoiceRANK1
3.0 A 1 False
202 1011 4ChoiceRANK2
3.0 B 1 False
202 1012 4ChoiceRANK3
3.0 C 1 False
202 1013 4ChoiceRANK4
3.0 D 1 False
203
203 1015 4ChoiceRANK1
2.0 A 1
203 1016 4ChoiceRANK2
2.0 B 1
203 1017 4ChoiceRANK3
2.0 C 1
203 1018 4ChoiceRANK4
204 1020 4ChoiceRANK1
1.0 D 1 False
204 1021 4ChoiceRANK2
1.0 B 1 False
204 1022 4ChoiceRANK3
1.0 C 1 False
204 1023 4ChoiceRANK4
1.0 A 1 False
205
205 1025 4ChoiceRANK1
2.0 B 1 False
205 1026 4ChoiceRANK2
2.0 D 1 False
205 1027 4ChoiceRANK3
2.0 C 1 False
205 1028 4ChoiceRANK4
2.0 A 1 False
206
206 1030 4ChoiceRANK1
2.0 A 1 False
206 1031 4ChoiceRANK2
2.0 C 1 False
```

206 1032 4ChoiceRANK3

```
2.0 B 1 False
206 1033 4ChoiceRANK4
2.0 D 1 False
207 1035 4ChoiceRANK1
3.0 C 1 False
207 1036 4ChoiceRANK2
3.0 D 1 False
207 1037 4ChoiceRANK3
3.0 B 1 False
207 1038 4ChoiceRANK4
3.0 A 1 False
208
208 1040 4ChoiceRANK1
4.0 A 1 False
208 1041 4ChoiceRANK2
4.0 B 1 False
208 1042 4ChoiceRANK3
4.0 C 1 False
208 1043 4ChoiceRANK4
4.0 D 1 False
209
209 1045 4ChoiceRANK1
3.0 B 1 False
209 1046 4ChoiceRANK2
3.0 D 1 False
209 1047 4ChoiceRANK3
3.0 C 1 False
209 1048 4ChoiceRANK4
3.0 A 1 False
210
210 1050 4ChoiceRANK1
2.0 A 1 False
210 1051 4ChoiceRANK2
2.0 C 1 False
210 1052 4ChoiceRANK3
2.0 B 1 False
210 1053 4ChoiceRANK4
2.0 D 1 False
211
211 1055 4ChoiceRANK1
3.0 B 1 False
211 1056 4ChoiceRANK2
3.0 C 1 False
211 1057 4ChoiceRANK3
3.0 D 1 False
211 1058 4ChoiceRANK4
3.0 A 1 False
212
212 1060 4ChoiceRANK1
2.0 A 1
212 1061 4ChoiceRANK2
2.0 B 1
212 1062 4ChoiceRANK3
2.0 C 1
212 1063 4ChoiceRANK4
error
213
213 1065 4ChoiceRANK1
1.0 B 1 False
213 1066 4ChoiceRANK2
1.0 D 1 False
213 1067 4ChoiceRANK3
1.0 C 1 False
213 1068 4ChoiceRANK4
1.0 A 1 False
214
214 1070 4ChoiceRANK1
3.0 D 1
214 1071 4ChoiceRANK2
3.0 B 1
214 1072 4ChoiceRANK3
3.0 C 1
214 1073 4ChoiceRANK4
3.0 A 1
215
215 1075 4ChoiceRANK1
3.0 A 1 False
215 1076 4ChoiceRANK2
3.0 B 1 False
215 1077 4ChoiceRANK3
3.0 D 1 False
215 1078 4ChoiceRANK4
3.0 C 1 False
216
216 1080 4ChoiceRANK1
3.0 D 1 False
216 1081 4ChoiceRANK2
3.0 B 1 False
216 1082 4ChoiceRANK3
3.0 C 1 False
```

216 1083 4ChoiceRANK4

3.0 A 1 False

217 217 1085 4ChoiceRANK1 3.0 B 1 False 217 1086 4ChoiceRANK2 3.0 C 1 False 217 1087 4ChoiceRANK3 3.0 A 1 False 217 1088 4ChoiceRANK4 3.0 D 1 False 218 218 1090 4ChoiceRANK1 2.0 D 1 False 218 1091 4ChoiceRANK2 2.0 B 1 False 218 1092 4ChoiceRANK3 2.0 A 1 False 218 1093 4ChoiceRANK4 2.0 C 1 False 219 219 1095 4ChoiceRANK1 2.0 B 1 False 219 1096 4ChoiceRANK2 2.0 D 1 False 219 1097 4ChoiceRANK3 2.0 A 1 False 219 1098 4ChoiceRANK4 2.0 C 1 False 220 220 1100 4ChoiceRANK1 1.0 B 1 False 220 1101 4ChoiceRANK2 1.0 D 1 False 220 1102 4ChoiceRANK3 1.0 A 1 False 220 1103 4ChoiceRANK4 1.0 C 1 False 221 221 1105 4ChoiceRANK1 1.0 A 1 False 221 1106 4ChoiceRANK2 1.0 B 1 False 221 1107 4ChoiceRANK3 1.0 D 1 False 221 1108 4ChoiceRANK4 1.0 C 1 False 222 222 1110 4ChoiceRANK1 3.0 B 1 False 222 1111 4ChoiceRANK2 3.0 A 1 False 222 1112 4ChoiceRANK3 3.0 D 1 False 222 1113 4ChoiceRANK4 3.0 C 1 False 223 223 1115 4ChoiceRANK1 2.0 B 1 False 223 1116 4ChoiceRANK2 2.0 D 1 False 223 1117 4ChoiceRANK3 2.0 C 1 False 223 1118 4ChoiceRANK4 2.0 A 1 False 224 224 1120 4ChoiceRANK1 2.0 B 1 False 224 1121 4ChoiceRANK2 2.0 A 1 False 224 1122 4ChoiceRANK3 2.0 D 1 False 224 1123 4ChoiceRANK4 2.0 C 1 False 225 225 1125 4ChoiceRANK1 2.0 A 1 False 225 1126 4ChoiceRANK2 2.0 D 1 False 225 1127 4ChoiceRANK3 2.0 B 1 False 225 1128 4ChoiceRANK4 2.0 C 1 False 226 226 1130 4ChoiceRANK1 3.0 A 1 False 226 1131 4ChoiceRANK2 3.0 B 1 False 226 1132 4ChoiceRANK3 3.0 D 1 False 226 1133 4ChoiceRANK4 3.0 C 1 False 227

227 1135 4ChoiceRANK1 2.0 D 1 False

```
227 1136 4ChoiceRANK2
2.0 C 1 False
227 1137 4ChoiceRANK3
2.0 B 1 False
227 1138 4ChoiceRANK4
2.0 A 1 False
228 1140 4ChoiceRANK1
2.0 D 1 False
228 1141 4ChoiceRANK2
2.0 C 1 False
228 1142 4ChoiceRANK3
2.0 B 1 False
228 1143 4ChoiceRANK4
2.0 A 1 False
229
229 1145 4ChoiceRANK1
2.0 B 1 False
229 1146 4ChoiceRANK2
2.0 A 1 False
229 1147 4ChoiceRANK3
2.0 D 1 False
229 1148 4ChoiceRANK4
2.0 C 1 False
230 1150 4ChoiceRANK1
3.0 A 1
230 1151 4ChoiceRANK2
3.0 B 1
230 1152 4ChoiceRANK3
3.0 D 1
230 1153 4ChoiceRANK4
3.0 C 1
231
231 1155 4ChoiceRANK1
1.0 B 1 False
231 1156 4ChoiceRANK2
1.0 A 1 False
231 1157 4ChoiceRANK3
1.0 D 1 False
231 1158 4ChoiceRANK4
1.0 C 1 False
232
232 1160 4ChoiceRANK1
3.0 B 1 False
232 1161 4ChoiceRANK2
3.0 A 1 False
232 1162 4ChoiceRANK3
3.0 D 1 False
232 1163 4ChoiceRANK4
3.0 C 1 False
233
233 1165 4ChoiceRANK1
3.0 D 1
233 1166 4ChoiceRANK2
3.0 C 1
233 1167 4ChoiceRANK3
3.0 B 1
233 1168 4ChoiceRANK4
3.0 A 1
234
234 1170 4ChoiceRANK1
1.0 A 1 False
234 1171 4ChoiceRANK2
1.0 B 1 False
234 1172 4ChoiceRANK3
1.0 C 1 False
234 1173 4ChoiceRANK4
1.0 D 1 False
235 1175 4ChoiceRANK1
2.0 A 1 False
235 1176 4ChoiceRANK2
2.0 D 1 False
235 1177 4ChoiceRANK3
2.0 B 1 False
235 1178 4ChoiceRANK4
2.0 C 1 False
236
236 1180 4ChoiceRANK1
2.0 C 1
236 1181 4ChoiceRANK2
2.0 B 1
236 1182 4ChoiceRANK3
2.0 A 1
236 1183 4ChoiceRANK4
error
237
237 1185 4ChoiceRANK1
2.0 B 1 False
237 1186 4ChoiceRANK2
2.0 C 1 False
237 1187 4ChoiceRANK3
```

```
2.0 D 1 False
237 1188 4ChoiceRANK4
2.0 A 1 False
238
238 1190 4ChoiceRANK1
1.0 A 1 False
238 1191 4ChoiceRANK2
1.0 B 1 False
238 1192 4ChoiceRANK3
1.0 D 1 False
238 1193 4ChoiceRANK4
1.0 C 1 False
239
239 1195 4ChoiceRANK1
2.0 B 1 False
239 1196 4ChoiceRANK2
2.0 A 1 False
239 1197 4ChoiceRANK3
2.0 C 1 False
239 1198 4ChoiceRANK4
2.0 D 1 False
240
240 1200 4ChoiceRANK1
2.0 D 1 False
240 1201 4ChoiceRANK2
2.0 B 1 False
240 1202 4ChoiceRANK3
2.0 A 1 False
240 1203 4ChoiceRANK4
2.0 C 1 False
241
241 1205 4ChoiceRANK1
2.0 B 1
241 1206 4ChoiceRANK2
2.0 C 1
241 1207 4ChoiceRANK3
2.0 A 1
241 1208 4ChoiceRANK4
error
242
242 1210 4ChoiceRANK1
3.0 D 1 False
242 1211 4ChoiceRANK2
3.0 C 1 False
242 1212 4ChoiceRANK3
3.0 B 1 False
242 1213 4ChoiceRANK4
3.0 A 1 False
243
243 1215 4ChoiceRANK1
3.0 D 1 False
243 1216 4ChoiceRANK2
3.0 B 1 False
243 1217 4ChoiceRANK3
3.0 A 1 False
243 1218 4ChoiceRANK4
3.0 C 1 False
244
244 1220 4ChoiceRANK1
3.0 A 1 False
244 1221 4ChoiceRANK2
3.0 D 1 False
244 1222 4ChoiceRANK3
3.0 C 1 False
244 1223 4ChoiceRANK4
3.0 B 1 False
245
245 1225 4ChoiceRANK1
2.0 D 1 False
245 1226 4ChoiceRANK2
2.0 C 1 False
245 1227 4ChoiceRANK3
2.0 A 1 False
245 1228 4ChoiceRANK4
2.0 B 1 False
246
246 1230 4ChoiceRANK1
2.0 C 1
246 1231 4ChoiceRANK2
2.0 B 1
246 1232 4ChoiceRANK3
2.0 A 1
246 1233 4ChoiceRANK4
error
247
247 1235 4ChoiceRANK1
1.0 A 1 False
247 1236 4ChoiceRANK2
1.0 D 1 False
247 1237 4ChoiceRANK3
```

1.0 C 1 False 247 1238 4ChoiceRANK4 1.0 B 1 False

248 248 1240 4ChoiceRANK1 2.0 A 1 False 248 1241 4ChoiceRANK2 2.0 B 1 False 248 1242 4ChoiceRANK3 2.0 C 1 False 248 1243 4ChoiceRANK4 2.0 D 1 False 249 1245 4ChoiceRANK1 2.0 C 1 False 249 1246 4ChoiceRANK2 2.0 A 1 False 249 1247 4ChoiceRANK3 2.0 D 1 False 249 1248 4ChoiceRANK4 2.0 B 1 False 250 250 1250 4ChoiceRANK1 3.0 D 1 250 1251 4ChoiceRANK2 3.0 B 1 250 1252 4ChoiceRANK3 3.0 C 1 250 1253 4ChoiceRANK4 3.0 A 1 251 251 1255 4ChoiceRANK1 2.0 B 1 False 251 1256 4ChoiceRANK2 2.0 C 1 False 251 1257 4ChoiceRANK3 2.0 A 1 False 251 1258 4ChoiceRANK4 2.0 D 1 False 252 252 1260 4ChoiceRANK1 2.0 B 1 False 252 1261 4ChoiceRANK2 2.0 A 1 False 252 1262 4ChoiceRANK3 2.0 C 1 False 252 1263 4ChoiceRANK4 2.0 D 1 False 253 253 1265 4ChoiceRANK1 2.0 A 1 253 1266 4ChoiceRANK2 2.0 C 1 253 1267 4ChoiceRANK3 2.0 B 1 253 1268 4ChoiceRANK4 254 254 1270 4ChoiceRANK1 2.0 B 1 254 1271 4ChoiceRANK2 2.0 C 1 254 1272 4ChoiceRANK3 2.0 A 1 254 1273 4ChoiceRANK4 error 255 255 1275 4ChoiceRANK1 4.0 A 1 False 255 1276 4ChoiceRANK2 4.0 B 1 False 255 1277 4ChoiceRANK3 4.0 C 1 False 255 1278 4ChoiceRANK4 4.0 D 1 False 256 256 1280 4ChoiceRANK1 2.0 B 1 256 1281 4ChoiceRANK2 2.0 A 1 256 1282 4ChoiceRANK3 2.0 C 1 256 1283 4ChoiceRANK4 error 257 257 1285 4ChoiceRANK1 2.0 C 1 False 257 1286 4ChoiceRANK2 2.0 A 1 False 257 1287 4ChoiceRANK3 2.0 D 1 False 257 1288 4ChoiceRANK4 2.0 B 1 False 258 258 1290 4ChoiceRANK1

3.0 A 1 False

```
258 1291 4ChoiceRANK2
3.0 C 1 False
258 1292 4ChoiceRANK3
3.0 B 1 False
258 1293 4ChoiceRANK4
3.0 D 1 False
259 1295 4ChoiceRANK1
2.0 B 1 False
259 1296 4ChoiceRANK2
2.0 D 1 False
259 1297 4ChoiceRANK3
2.0 A 1 False
259 1298 4ChoiceRANK4
2.0 C 1 False
260
260 1300 4ChoiceRANK1
2.0 D 1 False
260 1301 4ChoiceRANK2
2.0 B 1 False
260 1302 4ChoiceRANK3
2.0 A 1 False
260 1303 4ChoiceRANK4
2.0 C 1 False
261 1305 4ChoiceRANK1
2.0 D 1 False
261 1306 4ChoiceRANK2
2.0 B 1 False
261 1307 4ChoiceRANK3
2.0 A 1 False
261 1308 4ChoiceRANK4
2.0 C 1 False
262
262 1310 4ChoiceRANK1
2.0 A 1
262 1311 4ChoiceRANK2
2.0 C 1
262 1312 4ChoiceRANK3
2.0 B 1
262 1313 4ChoiceRANK4
error
263
263 1315 4ChoiceRANK1
3.0 C 1 False
263 1316 4ChoiceRANK2
3.0 D 1 False
263 1317 4ChoiceRANK3
3.0 A 1 False
263 1318 4ChoiceRANK4
3.0 B 1 False
264
264 1320 4ChoiceRANK1
2.0 D 1 False
264 1321 4ChoiceRANK2
2.0 B 1 False
264 1322 4ChoiceRANK3
2.0 A 1 False
264 1323 4ChoiceRANK4
2.0 C 1 False
265
265 1325 4ChoiceRANK1
2.0 A 1
265 1326 4ChoiceRANK2
2.0 B 1
265 1327 4ChoiceRANK3
2.0 C 1
265 1328 4ChoiceRANK4
266 1330 4ChoiceRANK1
1.0 B 1 False
266 1331 4ChoiceRANK2
1.0 D 1 False
266 1332 4ChoiceRANK3
1.0 A 1 False
266 1333 4ChoiceRANK4
1.0 C 1 False
267
267 1335 4ChoiceRANK1
1.0 A 1 False
267 1336 4ChoiceRANK2
1.0 C 1 False
267 1337 4ChoiceRANK3
1.0 B 1 False
267 1338 4ChoiceRANK4
1.0 D 1 False
268
268 1340 4ChoiceRANK1
2.0 C 1
268 1341 4ChoiceRANK2
2.0 B 1
```

268 1342 4ChoiceRANK3

```
2.0 A 1
268 1343 4ChoiceRANK4
269
269 1345 4ChoiceRANK1
3.0 A 1 False
269 1346 4ChoiceRANK2
3.0 B 1 False
269 1347 4ChoiceRANK3
3.0 D 1 False
269 1348 4ChoiceRANK4
3.0 C 1 False
270
270 1350 4ChoiceRANK1
2.0 D 1 False
270 1351 4ChoiceRANK2
2.0 C 1 False
270 1352 4ChoiceRANK3
2.0 B 1 False
270 1353 4ChoiceRANK4
2.0 A 1 False
271
271 1355 4ChoiceRANK1
3.0 B 1 False
271 1356 4ChoiceRANK2
3.0 D 1 False
271 1357 4ChoiceRANK3
3.0 A 1 False
271 1358 4ChoiceRANK4
3.0 C 1 False
272
272 1360 4ChoiceRANK1
3.0 B 1 False
272 1361 4ChoiceRANK2
3.0 D 1 False
272 1362 4ChoiceRANK3
3.0 C 1 False
272 1363 4ChoiceRANK4
3.0 A 1 False
273
273 1365 4ChoiceRANK1
2.0 A 1
273 1366 4ChoiceRANK2
2.0 B 1
273 1367 4ChoiceRANK3
2.0 C 1
273 1368 4ChoiceRANK4
error
274
274 1370 4ChoiceRANK1
2.0 B 1 False
274 1371 4ChoiceRANK2
2.0 D 1 False
274 1372 4ChoiceRANK3
2.0 C 1 False
274 1373 4ChoiceRANK4
2.0 A 1 False
275
275 1375 4ChoiceRANK1
2.0 D 1 False
275 1376 4ChoiceRANK2
2.0 C 1 False
275 1377 4ChoiceRANK3
2.0 B 1 False
275 1378 4ChoiceRANK4
2.0 A 1 False
276
276 1380 4ChoiceRANK1
2.0 B 1
276 1381 4ChoiceRANK2
2.0 A 1
276 1382 4ChoiceRANK3
2.0 C 1
276 1383 4ChoiceRANK4
error
277
277 1385 4ChoiceRANK1
2.0 C 1
277 1386 4ChoiceRANK2
2.0 B 1
277 1387 4ChoiceRANK3
2.0 A 1
277 1388 4ChoiceRANK4
error
278
278 1390 4ChoiceRANK1
2.0 B 1 False
278 1391 4ChoiceRANK2
2.0 D 1 False
278 1392 4ChoiceRANK3
2.0 A 1 False
278 1393 4ChoiceRANK4
```

2.0 C 1 False

```
279
279 1395 4ChoiceRANK1
4.0 B 1 False
279 1396 4ChoiceRANK2
4.0 A 1 False
279 1397 4ChoiceRANK3
4.0 C 1 False
279 1398 4ChoiceRANK4
4.0 D 1 False
280 1400 4ChoiceRANK1
3.0 D 1
280 1401 4ChoiceRANK2
3.0 B 1
280 1402 4ChoiceRANK3
3.0 C 1
280 1403 4ChoiceRANK4
3.0 A 1
281
281 1405 4ChoiceRANK1
2.0 D 1 False
281 1406 4ChoiceRANK2
2.0 B 1 False
281 1407 4ChoiceRANK3
2.0 A 1 False
281 1408 4ChoiceRANK4
2.0 C 1 False
282
282 1410 4ChoiceRANK1
2.0 A 1 False
282 1411 4ChoiceRANK2
2.0 B 1 False
282 1412 4ChoiceRANK3
2.0 C 1 False
282 1413 4ChoiceRANK4
2.0 D 1 False
283
283 1415 4ChoiceRANK1
1.0 A 1 False
283 1416 4ChoiceRANK2
1.0 B 1 False
283 1417 4ChoiceRANK3
1.0 C 1 False
283 1418 4ChoiceRANK4
1.0 D 1 False
284
284 1420 4ChoiceRANK1
2.0 A 1
284 1421 4ChoiceRANK2
2.0 B 1
284 1422 4ChoiceRANK3
2.0 C 1
284 1423 4ChoiceRANK4
error
285
285 1425 4ChoiceRANK1
3.0 B 1 False
285 1426 4ChoiceRANK2
3.0 D 1 False
285 1427 4ChoiceRANK3
3.0 A 1 False
285 1428 4ChoiceRANK4
3.0 C 1 False
286
286 1430 4ChoiceRANK1
3.0 B 1 False
286 1431 4ChoiceRANK2
3.0 A 1 False
286 1432 4ChoiceRANK3
3.0 D 1 False
286 1433 4ChoiceRANK4
3.0 C 1 False
287
287 1435 4ChoiceRANK1
2.0 B 1 False
287 1436 4ChoiceRANK2
2.0 A 1 False
287 1437 4ChoiceRANK3
2.0 D 1 False
287 1438 4ChoiceRANK4
2.0 C 1 False
288
288 1440 4ChoiceRANK1
2.0 C 1
288 1441 4ChoiceRANK2
2.0 B 1
288 1442 4ChoiceRANK3
2.0 A 1
288 1443 4ChoiceRANK4
error
289
289 1445 4ChoiceRANK1
```

3.0 C 1

```
289 1446 4ChoiceRANK2
3.0 B 1
289 1447 4ChoiceRANK3
3.0 A 1
289 1448 4ChoiceRANK4
3.0 D 1
290
290 1450 4ChoiceRANK1
3.0 A 1 False
290 1451 4ChoiceRANK2
3.0 B 1 False
290 1452 4ChoiceRANK3
3.0 C 1 False
290 1453 4ChoiceRANK4
3.0 D 1 False
291
291 1455 4ChoiceRANK1
2.0 C 1 False
291 1456 4ChoiceRANK2
2.0 B 1 False
291 1457 4ChoiceRANK3
2.0 A 1 False
291 1458 4ChoiceRANK4
2.0 D 1 False
292 1460 4ChoiceRANK1
2.0 C 1
292 1461 4ChoiceRANK2
2.0 B 1
292 1462 4ChoiceRANK3
2.0 A 1
292 1463 4ChoiceRANK4
error
293
293 1465 4ChoiceRANK1
2.0 A 1 False
293 1466 4ChoiceRANK2
2.0 D 1 False
293 1467 4ChoiceRANK3
2.0 C 1 False
293 1468 4ChoiceRANK4
2.0 B 1 False
294
294 1470 4ChoiceRANK1
3.0 A 1 False
294 1471 4ChoiceRANK2
3.0 B 1 False
294 1472 4ChoiceRANK3
3.0 C 1 False
294 1473 4ChoiceRANK4
3.0 D 1 False
295
295 1475 4ChoiceRANK1
2.0 A 1 False
295 1476 4ChoiceRANK2
2.0 B 1 False
295 1477 4ChoiceRANK3
2.0 D 1 False
295 1478 4ChoiceRANK4
2.0 C 1 False
296
296 1480 4ChoiceRANK1
3.0 B 1 False
296 1481 4ChoiceRANK2
3.0 A 1 False
296 1482 4ChoiceRANK3
3.0 D 1 False
296 1483 4ChoiceRANK4
3.0 C 1 False
297
297 1485 4ChoiceRANK1
3.0 A 1 False
297 1486 4ChoiceRANK2
3.0 B 1 False
297 1487 4ChoiceRANK3
3.0 C 1 False
297 1488 4ChoiceRANK4
3.0 D 1 False
298
298 1490 4ChoiceRANK1
3.0 B 1 False
298 1491 4ChoiceRANK2
3.0 A 1 False
298 1492 4ChoiceRANK3
3.0 D 1 False
298 1493 4ChoiceRANK4
3.0 C 1 False
299
299 1495 4ChoiceRANK1
3.0 B 1 False
299 1496 4ChoiceRANK2
3.0 A 1 False
```

299 1497 4ChoiceRANK3

```
3.0 D 1 False
299 1498 4ChoiceRANK4
3.0 C 1 False
300 1500 4ChoiceRANK1
1.0 D 1 False
300 1501 4ChoiceRANK2
1.0 C 1 False
300 1502 4ChoiceRANK3
1.0 B 1 False
300 1503 4ChoiceRANK4
1.0 A 1 False
301
301 1505 4ChoiceRANK1
1.0 B 1 False
301 1506 4ChoiceRANK2
1.0 D 1 False
301 1507 4ChoiceRANK3
1.0 A 1 False
301 1508 4ChoiceRANK4
1.0 C 1 False
302
302 1510 4ChoiceRANK1
3.0 A 1 False
302 1511 4ChoiceRANK2
3.0 B 1 False
302 1512 4ChoiceRANK3
3.0 C 1 False
302 1513 4ChoiceRANK4
3.0 D 1 False
303 1515 4ChoiceRANK1
2.0 B 1 False
303 1516 4ChoiceRANK2
2.0 A 1 False
303 1517 4ChoiceRANK3
2.0 D 1 False
303 1518 4ChoiceRANK4
2.0 C 1 False
304
304 1520 4ChoiceRANK1
2.0 B 1 False
304 1521 4ChoiceRANK2
2.0 A 1 False
304 1522 4ChoiceRANK3
2.0 D 1 False
304 1523 4ChoiceRANK4
2.0 C 1 False
305
305 1525 4ChoiceRANK1
3.0 A 1 False
305 1526 4ChoiceRANK2
3.0 D 1 False
305 1527 4ChoiceRANK3
3.0 B 1 False
305 1528 4ChoiceRANK4
3.0 C 1 False
306 1530 4ChoiceRANK1
3.0 B 1 False
306 1531 4ChoiceRANK2
3.0 D 1 False
306 1532 4ChoiceRANK3
3.0 A 1 False
306 1533 4ChoiceRANK4
3.0 C 1 False
307
307 1535 4ChoiceRANK1
3.0 A 1 False
307 1536 4ChoiceRANK2
3.0 B 1 False
307 1537 4ChoiceRANK3
3.0 D 1 False
307 1538 4ChoiceRANK4
3.0 C 1 False
308
308 1540 4ChoiceRANK1
2.0 B 1
308 1541 4ChoiceRANK2
2.0 A 1
308 1542 4ChoiceRANK3
2.0 C 1
308 1543 4ChoiceRANK4
error
309
309 1545 4ChoiceRANK1
3.0 B 1 False
309 1546 4ChoiceRANK2
3.0 A 1 False
309 1547 4ChoiceRANK3
3.0 D 1 False
```

309 1548 4ChoiceRANK4

3.0 C 1 False

```
310
310 1550 4ChoiceRANK1
2.0 C 1
310 1551 4ChoiceRANK2
2.0 B 1
310 1552 4ChoiceRANK3
2.0 A 1
310 1553 4ChoiceRANK4
error
311
311 1555 4ChoiceRANK1
2.0 B 1
311 1556 4ChoiceRANK2
2.0 C 1
311 1557 4ChoiceRANK3
2.0 A 1
311 1558 4ChoiceRANK4
error
312
312 1560 4ChoiceRANK1
1.0 D 1 False
312 1561 4ChoiceRANK2
1.0 C 1 False
312 1562 4ChoiceRANK3
1.0 B 1 False
312 1563 4ChoiceRANK4
1.0 A 1 False
313
313 1565 4ChoiceRANK1
2.0 B 1 False
313 1566 4ChoiceRANK2
2.0 D 1 False
313 1567 4ChoiceRANK3
2.0 A 1 False
313 1568 4ChoiceRANK4
2.0 C 1 False
314
314 1570 4ChoiceRANK1
2.0 C 1
314 1571 4ChoiceRANK2
2.0 A 1
314 1572 4ChoiceRANK3
2.0 B 1
314 1573 4ChoiceRANK4
error
315
315 1575 4ChoiceRANK1
4.0 A 1 False
315 1576 4ChoiceRANK2
4.0 B 1 False
315 1577 4ChoiceRANK3
4.0 C 1 False
315 1578 4ChoiceRANK4
4.0 D 1 False
316
316 1580 4ChoiceRANK1
2.0 A 1 False
316 1581 4ChoiceRANK2
2.0 D 1 False
316 1582 4ChoiceRANK3
2.0 C 1 False
316 1583 4ChoiceRANK4
2.0 B 1 False
317
317 1585 4ChoiceRANK1
1.0 B 1 False
317 1586 4ChoiceRANK2
1.0 A 1 False
317 1587 4ChoiceRANK3
1.0 D 1 False
317 1588 4ChoiceRANK4
1.0 C 1 False
318
318 1590 4ChoiceRANK1
2.0 B 1 False
318 1591 4ChoiceRANK2
2.0 A 1 False
318 1592 4ChoiceRANK3
2.0 D 1 False
318 1593 4ChoiceRANK4
2.0 C 1 False
319
319 1595 4ChoiceRANK1
2.0 A 1 False
319 1596 4ChoiceRANK2
2.0 C 1 False
319 1597 4ChoiceRANK3
2.0 B 1 False
319 1598 4ChoiceRANK4
2.0 D 1 False
320
```

320 1600 4ChoiceRANK1 2.0 C 1

```
320 1601 4ChoiceRANK2
2.0 B 1
320 1602 4ChoiceRANK3
2.0 A 1
320 1603 4ChoiceRANK4
error
321
321 1605 4ChoiceRANK1
3.0 B 1 False
321 1606 4ChoiceRANK2
3.0 A 1 False
321 1607 4ChoiceRANK3
3.0 D 1 False
321 1608 4ChoiceRANK4
3.0 C 1 False
322
322 1610 4ChoiceRANK1
3.0 B 1 False
322 1611 4ChoiceRANK2
3.0 A 1 False
322 1612 4ChoiceRANK3
3.0 D 1 False
322 1613 4ChoiceRANK4
3.0 C 1 False
323 1615 4ChoiceRANK1
2.0 B 1 False
323 1616 4ChoiceRANK2
2.0 C 1 False
323 1617 4ChoiceRANK3
2.0 A 1 False
323 1618 4ChoiceRANK4
2.0 D 1 False
324
324 1620 4ChoiceRANK1
2.0 C 1
324 1621 4ChoiceRANK2
2.0 B 1
324 1622 4ChoiceRANK3
2.0 A 1
324 1623 4ChoiceRANK4
error
325
325 1625 4ChoiceRANK1
2.0 B 1 False
325 1626 4ChoiceRANK2
2.0 C 1 False
325 1627 4ChoiceRANK3
2.0 A 1 False
325 1628 4ChoiceRANK4
2.0 D 1 False
326
326 1630 4ChoiceRANK1
3.0 B 1 False
326 1631 4ChoiceRANK2
3.0 D 1 False
326 1632 4ChoiceRANK3
3.0 A 1 False
326 1633 4ChoiceRANK4
3.0 C 1 False
327
327 1635 4ChoiceRANK1
3.0 A 1 False
327 1636 4ChoiceRANK2
3.0 B 1 False
327 1637 4ChoiceRANK3
3.0 D 1 False
327 1638 4ChoiceRANK4
3.0 C 1 False
328 1640 4ChoiceRANK1
3.0 D 1 False
328 1641 4ChoiceRANK2
3.0 B 1 False
328 1642 4ChoiceRANK3
3.0 A 1 False
328 1643 4ChoiceRANK4
3.0 C 1 False
329
329 1645 4ChoiceRANK1
2.0 D 1 False
329 1646 4ChoiceRANK2
2.0 A 1 False
329 1647 4ChoiceRANK3
2.0 B 1 False
329 1648 4ChoiceRANK4
2.0 C 1 False
330
330 1650 4ChoiceRANK1
1.0 B 1 False
330 1651 4ChoiceRANK2
1.0 D 1 False
```

330 1652 4ChoiceRANK3

```
1.0 C 1 False
330 1653 4ChoiceRANK4
1.0 A 1 False
331
331 1655 4ChoiceRANK1
3.0 B 1 False
331 1656 4ChoiceRANK2
3.0 D 1 False
331 1657 4ChoiceRANK3
3.0 A 1 False
331 1658 4ChoiceRANK4
3.0 C 1 False
332
332 1660 4ChoiceRANK1
3.0 D 1
332 1661 4ChoiceRANK2
3.0 B 1
332 1662 4ChoiceRANK3
3.0 C 1
332 1663 4ChoiceRANK4
3.0 A 1
333
333 1665 4ChoiceRANK1
2.0 D 1 False
333 1666 4ChoiceRANK2
2.0 B 1 False
333 1667 4ChoiceRANK3
2.0 C 1 False
333 1668 4ChoiceRANK4
2.0 A 1 False
334
334 1670 4ChoiceRANK1
4.0 A 1 False
334 1671 4ChoiceRANK2
4.0 B 1 False
334 1672 4ChoiceRANK3
4.0 C 1 False
334 1673 4ChoiceRANK4
4.0 D 1 False
335
335 1675 4ChoiceRANK1
3.0 D 1
335 1676 4ChoiceRANK2
3.0 B 1
335 1677 4ChoiceRANK3
3.0 C 1
335 1678 4ChoiceRANK4
3.0 A 1
336
336 1680 4ChoiceRANK1
3.0 B 1 False
336 1681 4ChoiceRANK2
3.0 A 1 False
336 1682 4ChoiceRANK3
3.0 C 1 False
336 1683 4ChoiceRANK4
3.0 D 1 False
337
337 1685 4ChoiceRANK1
3.0 B 1 False
337 1686 4ChoiceRANK2
3.0 D 1 False
337 1687 4ChoiceRANK3
3.0 C 1 False
337 1688 4ChoiceRANK4
3.0 A 1 False
338
338 1690 4ChoiceRANK1
2.0 A 1
338 1691 4ChoiceRANK2
2.0 B 1
338 1692 4ChoiceRANK3
2.0 C 1
338 1693 4ChoiceRANK4
error
339
339 1695 4ChoiceRANK1
2.0 B 1 False
339 1696 4ChoiceRANK2
2.0 A 1 False
339 1697 4ChoiceRANK3
2.0 D 1 False
339 1698 4ChoiceRANK4
2.0 C 1 False
340
340 1700 4ChoiceRANK1
1.0 B 1 False
340 1701 4ChoiceRANK2
1.0 D 1 False
340 1702 4ChoiceRANK3
1.0 A 1 False
```

340 1703 4ChoiceRANK4

1.0 C 1 False

341 341 1705 4ChoiceRANK1 2.0 C 1 341 1706 4ChoiceRANK2 2.0 A 1 341 1707 4ChoiceRANK3 2.0 B 1 341 1708 4ChoiceRANK4 error 342 1710 4ChoiceRANK1 3.0 C 1 False 342 1711 4ChoiceRANK2 3.0 D 1 False 342 1712 4ChoiceRANK3 3.0 A 1 False 342 1713 4ChoiceRANK4 3.0 B 1 False 343 343 1715 4ChoiceRANK1 3.0 D 1 False 343 1716 4ChoiceRANK2 3.0 C 1 False 343 1717 4ChoiceRANK3 3.0 B 1 False 343 1718 4ChoiceRANK4 3.0 A 1 False 344 344 1720 4ChoiceRANK1 2.0 B 1 False 344 1721 4ChoiceRANK2 2.0 D 1 False 344 1722 4ChoiceRANK3 2.0 A 1 False 344 1723 4ChoiceRANK4 2.0 C 1 False 345 345 1725 4ChoiceRANK1 1.0 B 1 False 345 1726 4ChoiceRANK2 1.0 D 1 False 345 1727 4ChoiceRANK3 1.0 A 1 False 345 1728 4ChoiceRANK4 1.0 C 1 False 346 346 1730 4ChoiceRANK1 1.0 A 1 False 346 1731 4ChoiceRANK2 1.0 D 1 False 346 1732 4ChoiceRANK3 1.0 C 1 False 346 1733 4ChoiceRANK4 1.0 B 1 False 347 347 1735 4ChoiceRANK1 2.0 A 1 347 1736 4ChoiceRANK2 2.0 C 1 347 1737 4ChoiceRANK3 2.0 B 1 347 1738 4ChoiceRANK4 error 348 348 1740 4ChoiceRANK1 2.0 A 1 False 348 1741 4ChoiceRANK2 2.0 B 1 False 348 1742 4ChoiceRANK3 2.0 C 1 False 348 1743 4ChoiceRANK4 2.0 D 1 False 349 349 1745 4ChoiceRANK1 2.0 A 1 False 349 1746 4ChoiceRANK2 2.0 C 1 False 349 1747 4ChoiceRANK3 2.0 B 1 False 349 1748 4ChoiceRANK4 2.0 D 1 False 350 350 1750 4ChoiceRANK1 3.0 D 1 350 1751 4ChoiceRANK2 3.0 B 1 350 1752 4ChoiceRANK3 3.0 A 1 350 1753 4ChoiceRANK4 3.0 C 1 351

351 1755 4ChoiceRANK1

2.0 C 1 False

```
351 1756 4ChoiceRANK2
2.0 D 1 False
351 1757 4ChoiceRANK3
2.0 A 1 False
351 1758 4ChoiceRANK4
2.0 B 1 False
352 1760 4ChoiceRANK1
3.0 D 1
352 1761 4ChoiceRANK2
3.0 C 1
352 1762 4ChoiceRANK3
3.0 B 1
352 1763 4ChoiceRANK4
3.0 A 1
353
353 1765 4ChoiceRANK1
3.0 B 1 False
353 1766 4ChoiceRANK2
3.0 A 1 False
353 1767 4ChoiceRANK3
3.0 C 1 False
353 1768 4ChoiceRANK4
3.0 D 1 False
354 1770 4ChoiceRANK1
3.0 D 1 False
354 1771 4ChoiceRANK2
3.0 B 1 False
354 1772 4ChoiceRANK3
3.0 A 1 False
354 1773 4ChoiceRANK4
3.0 C 1 False
355
355 1775 4ChoiceRANK1
2.0 B 1 False
355 1776 4ChoiceRANK2
2.0 A 1 False
355 1777 4ChoiceRANK3
2.0 D 1 False
355 1778 4ChoiceRANK4
2.0 C 1 False
356
356 1780 4ChoiceRANK1
2.0 C 1
356 1781 4ChoiceRANK2
2.0 B 1
356 1782 4ChoiceRANK3
2.0 A 1
356 1783 4ChoiceRANK4
error
357
357 1785 4ChoiceRANK1
357 1786 4ChoiceRANK2
2.0 B 1
357 1787 4ChoiceRANK3
2.0 A 1
357 1788 4ChoiceRANK4
error
358
358 1790 4ChoiceRANK1
4.0 B 1 False
358 1791 4ChoiceRANK2
4.0 A 1 False
358 1792 4ChoiceRANK3
4.0 D 1 False
358 1793 4ChoiceRANK4
4.0 C 1 False
359 1795 4ChoiceRANK1
2.0 C 1
359 1796 4ChoiceRANK2
2.0 B 1
359 1797 4ChoiceRANK3
2.0 A 1
359 1798 4ChoiceRANK4
error
360
360 1800 4ChoiceRANK1
2.0 A 1 False
360 1801 4ChoiceRANK2
2.0 B 1 False
360 1802 4ChoiceRANK3
2.0 D 1 False
360 1803 4ChoiceRANK4
2.0 C 1 False
361
361 1805 4ChoiceRANK1
2.0 D 1 False
361 1806 4ChoiceRANK2
2.0 C 1 False
361 1807 4ChoiceRANK3
```

```
2.0 B 1 False
361 1808 4ChoiceRANK4
2.0 A 1 False
362 1810 4ChoiceRANK1
3.0 D 1
362 1811 4ChoiceRANK2
3.0 B 1
362 1812 4ChoiceRANK3
3.0 A 1
362 1813 4ChoiceRANK4
3.0 C 1
363
363 1815 4ChoiceRANK1
2.0 B 1 False
363 1816 4ChoiceRANK2
2.0 A 1 False
363 1817 4ChoiceRANK3
2.0 D 1 False
363 1818 4ChoiceRANK4
2.0 C 1 False
364 1820 4ChoiceRANK1
2.0 B 1 False
364 1821 4ChoiceRANK2
2.0 D 1 False
364 1822 4ChoiceRANK3
2.0 A 1 False
364 1823 4ChoiceRANK4
2.0 C 1 False
365
365 1825 4ChoiceRANK1
2.0 D 1 False
365 1826 4ChoiceRANK2
2.0 C 1 False
365 1827 4ChoiceRANK3
2.0 B 1 False
365 1828 4ChoiceRANK4
2.0 A 1 False
366
366 1830 4ChoiceRANK1
2.0 B 1 False
366 1831 4ChoiceRANK2
2.0 A 1 False
366 1832 4ChoiceRANK3
2.0 D 1 False
366 1833 4ChoiceRANK4
2.0 C 1 False
367
367 1835 4ChoiceRANK1
2.0 B 1 False
367 1836 4ChoiceRANK2
2.0 D 1 False
367 1837 4ChoiceRANK3
2.0 A 1 False
367 1838 4ChoiceRANK4
2.0 C 1 False
368 1840 4ChoiceRANK1
3.0 A 1
368 1841 4ChoiceRANK2
3.0 B 1
368 1842 4ChoiceRANK3
3.0 D 1
368 1843 4ChoiceRANK4
3.0 C 1
369
369 1845 4ChoiceRANK1
2.0 B 1 False
369 1846 4ChoiceRANK2
2.0 A 1 False
369 1847 4ChoiceRANK3
2.0 D 1 False
369 1848 4ChoiceRANK4
2.0 C 1 False
370
370 1850 4ChoiceRANK1
2.0 C 1 False
370 1851 4ChoiceRANK2
2.0 D 1 False
370 1852 4ChoiceRANK3
2.0 B 1 False
370 1853 4ChoiceRANK4
2.0 A 1 False
371 1855 4ChoiceRANK1
2.0 A 1
371 1856 4ChoiceRANK2
2.0 B 1
371 1857 4ChoiceRANK3
```

2.0 C 1 371 1858 4ChoiceRANK4 error

```
372
372 1860 4ChoiceRANK1
2.0 A 1 False
372 1861 4ChoiceRANK2
2.0 D 1 False
372 1862 4ChoiceRANK3
2.0 B 1 False
372 1863 4ChoiceRANK4
2.0 C 1 False
373 1865 4ChoiceRANK1
2.0 D 1 False
373 1866 4ChoiceRANK2
2.0 B 1 False
373 1867 4ChoiceRANK3
2.0 C 1 False
373 1868 4ChoiceRANK4
2.0 A 1 False
374
374 1870 4ChoiceRANK1
2.0 B 1 False
374 1871 4ChoiceRANK2
2.0 D 1 False
374 1872 4ChoiceRANK3
2.0 A 1 False
374 1873 4ChoiceRANK4
2.0 C 1 False
375
375 1875 4ChoiceRANK1
2.0 C 1 False
375 1876 4ChoiceRANK2
2.0 A 1 False
375 1877 4ChoiceRANK3
2.0 D 1 False
375 1878 4ChoiceRANK4
2.0 B 1 False
376
376 1880 4ChoiceRANK1
2.0 C 1
376 1881 4ChoiceRANK2
2.0 A 1
376 1882 4ChoiceRANK3
2.0 B 1
376 1883 4ChoiceRANK4
error
377
377 1885 4ChoiceRANK1
3.0 A 1 False
377 1886 4ChoiceRANK2
3.0 B 1 False
377 1887 4ChoiceRANK3
3.0 C 1 False
377 1888 4ChoiceRANK4
3.0 D 1 False
378
378 1890 4ChoiceRANK1
2.0 C 1 False
378 1891 4ChoiceRANK2
2.0 B 1 False
378 1892 4ChoiceRANK3
2.0 A 1 False
378 1893 4ChoiceRANK4
2.0 D 1 False
379
379 1895 4ChoiceRANK1
2.0 A 1 False
379 1896 4ChoiceRANK2
2.0 B 1 False
379 1897 4ChoiceRANK3
2.0 D 1 False
379 1898 4ChoiceRANK4
2.0 C 1 False
380
380 1900 4ChoiceRANK1
2.0 B 1 False
380 1901 4ChoiceRANK2
2.0 D 1 False
380 1902 4ChoiceRANK3
2.0 A 1 False
380 1903 4ChoiceRANK4
2.0 C 1 False
381
381 1905 4ChoiceRANK1
2.0 A 1 False
381 1906 4ChoiceRANK2
2.0 B 1 False
381 1907 4ChoiceRANK3
2.0 C 1 False
381 1908 4ChoiceRANK4
2.0 D 1 False
382
382 1910 4ChoiceRANK1
```

3.0 C 1

```
382 1911 4ChoiceRANK2
3.0 A 1
382 1912 4ChoiceRANK3
3.0 B 1
382 1913 4ChoiceRANK4
3.0 D 1
383
383 1915 4ChoiceRANK1
2.0 B 1 False
383 1916 4ChoiceRANK2
2.0 C 1 False
383 1917 4ChoiceRANK3
2.0 D 1 False
383 1918 4ChoiceRANK4
2.0 A 1 False
384
384 1920 4ChoiceRANK1
2.0 D 1 False
384 1921 4ChoiceRANK2
2.0 A 1 False
384 1922 4ChoiceRANK3
2.0 B 1 False
384 1923 4ChoiceRANK4
2.0 C 1 False
385 1925 4ChoiceRANK1
2.0 A 1 False
385 1926 4ChoiceRANK2
2.0 C 1 False
385 1927 4ChoiceRANK3
2.0 B 1 False
385 1928 4ChoiceRANK4
2.0 D 1 False
386
386 1930 4ChoiceRANK1
3.0 B 1 False
386 1931 4ChoiceRANK2
3.0 A 1 False
386 1932 4ChoiceRANK3
3.0 C 1 False
386 1933 4ChoiceRANK4
3.0 D 1 False
387
387 1935 4ChoiceRANK1
2.0 B 1 False
387 1936 4ChoiceRANK2
2.0 D 1 False
387 1937 4ChoiceRANK3
2.0 A 1 False
387 1938 4ChoiceRANK4
2.0 C 1 False
388
388 1940 4ChoiceRANK1
1.0 B 1 False
388 1941 4ChoiceRANK2
1.0 A 1 False
388 1942 4ChoiceRANK3
1.0 C 1 False
388 1943 4ChoiceRANK4
1.0 D 1 False
389
389 1945 4ChoiceRANK1
2.0 A 1 False
389 1946 4ChoiceRANK2
2.0 C 1 False
389 1947 4ChoiceRANK3
2.0 D 1 False
389 1948 4ChoiceRANK4
2.0 B 1 False
390 1950 4ChoiceRANK1
2.0 B 1
390 1951 4ChoiceRANK2
2.0 C 1
390 1952 4ChoiceRANK3
2.0 A 1
390 1953 4ChoiceRANK4
error
391
391 1955 4ChoiceRANK1
3.0 B 1 False
391 1956 4ChoiceRANK2
3.0 A 1 False
391 1957 4ChoiceRANK3
3.0 D 1 False
391 1958 4ChoiceRANK4
3.0 C 1 False
392
392 1960 4ChoiceRANK1
1.0 C 1 False
392 1961 4ChoiceRANK2
1.0 A 1 False
392 1962 4ChoiceRANK3
```

```
1.0 D 1 False
392 1963 4ChoiceRANK4
1.0 B 1 False
393
393 1965 4ChoiceRANK1
2.0 D 1 False
393 1966 4ChoiceRANK2
2.0 B 1 False
393 1967 4ChoiceRANK3
2.0 C 1 False
393 1968 4ChoiceRANK4
2.0 A 1 False
394
394 1970 4ChoiceRANK1
2.0 B 1
394 1971 4ChoiceRANK2
2.0 C 1
394 1972 4ChoiceRANK3
2.0 A 1
394 1973 4ChoiceRANK4
error
395
395 1975 4ChoiceRANK1
3.0 D 1 False
395 1976 4ChoiceRANK2
3.0 C 1 False
395 1977 4ChoiceRANK3
3.0 B 1 False
395 1978 4ChoiceRANK4
3.0 A 1 False
396
396 1980 4ChoiceRANK1
2.0 B 1 False
396 1981 4ChoiceRANK2
2.0 D 1 False
396 1982 4ChoiceRANK3
2.0 A 1 False
396 1983 4ChoiceRANK4
2.0 C 1 False
397
397 1985 4ChoiceRANK1
2.0 A 1 False
397 1986 4ChoiceRANK2
2.0 D 1 False
397 1987 4ChoiceRANK3
2.0 B 1 False
397 1988 4ChoiceRANK4
2.0 C 1 False
398
398 1990 4ChoiceRANK1
2.0 A 1 False
398 1991 4ChoiceRANK2
2.0 B 1 False
398 1992 4ChoiceRANK3
2.0 D 1 False
398 1993 4ChoiceRANK4
2.0 C 1 False
399 1995 4ChoiceRANK1
2.0 A 1 False
399 1996 4ChoiceRANK2
2.0 B 1 False
399 1997 4ChoiceRANK3
2.0 C 1 False
399 1998 4ChoiceRANK4
2.0 D 1 False
400
400 2000 4ChoiceRANK1
3.0 C 1 False
400 2001 4ChoiceRANK2
3.0 D 1 False
400 2002 4ChoiceRANK3
3.0 A 1 False
400 2003 4ChoiceRANK4
3.0 B 1 False
401
401 2005 4ChoiceRANK1
3.0 A 1 False
401 2006 4ChoiceRANK2
3.0 B 1 False
401 2007 4ChoiceRANK3
3.0 D 1 False
401 2008 4ChoiceRANK4
3.0 C 1 False
402
402 2010 4ChoiceRANK1
2.0 B 1 False
402 2011 4ChoiceRANK2
2.0 A 1 False
402 2012 4ChoiceRANK3
2.0 D 1 False
```

402 2013 4ChoiceRANK4

2.0 C 1 False

```
403
403 2015 4ChoiceRANK1
3.0 D 1 False
403 2016 4ChoiceRANK2
3.0 B 1 False
403 2017 4ChoiceRANK3
3.0 A 1 False
403 2018 4ChoiceRANK4
3.0 C 1 False
404 2020 4ChoiceRANK1
3.0 A 1 False
404 2021 4ChoiceRANK2
3.0 D 1 False
404 2022 4ChoiceRANK3
3.0 B 1 False
404 2023 4ChoiceRANK4
3.0 C 1 False
405
405 2025 4ChoiceRANK1
2.0 B 1 False
405 2026 4ChoiceRANK2
2.0 D 1 False
405 2027 4ChoiceRANK3
2.0 A 1 False
405 2028 4ChoiceRANK4
2.0 C 1 False
406
406 2030 4ChoiceRANK1
3.0 A 1 False
406 2031 4ChoiceRANK2
3.0 B 1 False
406 2032 4ChoiceRANK3
3.0 D 1 False
406 2033 4ChoiceRANK4
3.0 C 1 False
407
407 2035 4ChoiceRANK1
2.0 D 1 False
407 2036 4ChoiceRANK2
2.0 A 1 False
407 2037 4ChoiceRANK3
2.0 B 1 False
407 2038 4ChoiceRANK4
2.0 C 1 False
408
408 2040 4ChoiceRANK1
1.0 B 1 False
408 2041 4ChoiceRANK2
1.0 A 1 False
408 2042 4ChoiceRANK3
1.0 C 1 False
408 2043 4ChoiceRANK4
1.0 D 1 False
409 2045 4ChoiceRANK1
3.0 B 1 False
409 2046 4ChoiceRANK2
3.0 D 1 False
409 2047 4ChoiceRANK3
3.0 A 1 False
409 2048 4ChoiceRANK4
3.0 C 1 False
410
410 2050 4ChoiceRANK1
3.0 C 1 False
410 2051 4ChoiceRANK2
3.0 B 1 False
410 2052 4ChoiceRANK3
3.0 D 1 False
410 2053 4ChoiceRANK4
3.0 A 1 False
411
411 2055 4ChoiceRANK1
3.0 B 1 False
411 2056 4ChoiceRANK2
3.0 A 1 False
411 2057 4ChoiceRANK3
3.0 D 1 False
411 2058 4ChoiceRANK4
3.0 C 1 False
412
412 2060 4ChoiceRANK1
3.0 D 1 False
412 2061 4ChoiceRANK2
3.0 B 1 False
412 2062 4ChoiceRANK3
3.0 C 1 False
412 2063 4ChoiceRANK4
3.0 A 1 False
413
```

413 2065 4ChoiceRANK1 2.0 C 1

```
413 2066 4ChoiceRANK2
2.0 A 1
413 2067 4ChoiceRANK3
2.0 B 1
413 2068 4ChoiceRANK4
error
414
414 2070 4ChoiceRANK1
1.0 D 1 False
414 2071 4ChoiceRANK2
1.0 B 1 False
414 2072 4ChoiceRANK3
1.0 A 1 False
414 2073 4ChoiceRANK4
1.0 C 1 False
415
415 2075 4ChoiceRANK1
3.0 C 1 False
415 2076 4ChoiceRANK2
3.0 A 1 False
415 2077 4ChoiceRANK3
3.0 D 1 False
415 2078 4ChoiceRANK4
3.0 B 1 False
416
416 2080 4ChoiceRANK1
3.0 A 1
416 2081 4ChoiceRANK2
3.0 B 1
416 2082 4ChoiceRANK3
3.0 C 1
416 2083 4ChoiceRANK4
3.0 D 1
417
417 2085 4ChoiceRANK1
3.0 B 1 False
417 2086 4ChoiceRANK2
3.0 D 1 False
417 2087 4ChoiceRANK3
3.0 A 1 False
417 2088 4ChoiceRANK4
3.0 C 1 False
418
418 2090 4ChoiceRANK1
2.0 B 1
418 2091 4ChoiceRANK2
2.0 A 1
418 2092 4ChoiceRANK3
2.0 C 1
418 2093 4ChoiceRANK4
error
419
419 2095 4ChoiceRANK1
1.0 B 1 False
419 2096 4ChoiceRANK2
1.0 D 1 False
419 2097 4ChoiceRANK3
1.0 C 1 False
419 2098 4ChoiceRANK4
1.0 A 1 False
420
420 2100 4ChoiceRANK1
3.0 D 1
420 2101 4ChoiceRANK2
3.0 B 1
420 2102 4ChoiceRANK3
3.0 A 1
420 2103 4ChoiceRANK4
3.0 C 1
421 2105 4ChoiceRANK1
2.0 D 1 False
421 2106 4ChoiceRANK2
2.0 B 1 False
421 2107 4ChoiceRANK3
2.0 A 1 False
421 2108 4ChoiceRANK4
2.0 C 1 False
422
422 2110 4ChoiceRANK1
2.0 A 1 False
422 2111 4ChoiceRANK2
2.0 B 1 False
422 2112 4ChoiceRANK3
2.0 C 1 False
422 2113 4ChoiceRANK4
2.0 D 1 False
423
423 2115 4ChoiceRANK1
1.0 C 1
423 2116 4ChoiceRANK2
```

1.0 A 1 423 2117 4ChoiceRANK3

1.0 B 1 423 2118 4ChoiceRANK4 424 424 2120 4ChoiceRANK1 2.0 B 1 False 424 2121 4ChoiceRANK2 2.0 C 1 False 424 2122 4ChoiceRANK3 2.0 D 1 False 424 2123 4ChoiceRANK4 2.0 A 1 False 425 425 2125 4ChoiceRANK1 2.0 D 1 False 425 2126 4ChoiceRANK2 2.0 B 1 False 425 2127 4ChoiceRANK3 2.0 A 1 False 425 2128 4ChoiceRANK4 2.0 C 1 False 426 426 2130 4ChoiceRANK1 2.0 B 1 False 426 2131 4ChoiceRANK2 2.0 D 1 False 426 2132 4ChoiceRANK3 2.0 A 1 False 426 2133 4ChoiceRANK4 2.0 C 1 False 427 427 2135 4ChoiceRANK1 3.0 D 1 False 427 2136 4ChoiceRANK2 3.0 C 1 False 427 2137 4ChoiceRANK3 3.0 B 1 False 427 2138 4ChoiceRANK4 3.0 A 1 False 428 428 2140 4ChoiceRANK1 2.0 C 1 False 428 2141 4ChoiceRANK2 2.0 D 1 False 428 2142 4ChoiceRANK3 2.0 A 1 False 428 2143 4ChoiceRANK4 2.0 B 1 False 429 429 2145 4ChoiceRANK1 2.0 B 1 429 2146 4ChoiceRANK2 2.0 A 1 429 2147 4ChoiceRANK3 2.0 C 1 429 2148 4ChoiceRANK4 error 430 430 2150 4ChoiceRANK1 2.0 B 1 430 2151 4ChoiceRANK2 2.0 A 1 430 2152 4ChoiceRANK3 2.0 C 1 430 2153 4ChoiceRANK4 error 431 2155 4ChoiceRANK1 2.0 A 1 False 431 2156 4ChoiceRANK2 2.0 B 1 False 431 2157 4ChoiceRANK3 2.0 C 1 False 431 2158 4ChoiceRANK4 2.0 D 1 False 432 432 2160 4ChoiceRANK1 2.0 D 1 False 432 2161 4ChoiceRANK2 2.0 B 1 False 432 2162 4ChoiceRANK3 2.0 C 1 False 432 2163 4ChoiceRANK4 2.0 A 1 False 433 433 2165 4ChoiceRANK1 1.0 A 1 False 433 2166 4ChoiceRANK2 1.0 B 1 False 433 2167 4ChoiceRANK3 1.0 C 1 False 433 2168 4ChoiceRANK4

1.0 D 1 False

434 434 2170 4ChoiceRANK1 1.0 A 1 False 434 2171 4ChoiceRANK2 1.0 C 1 False 434 2172 4ChoiceRANK3 1.0 B 1 False 434 2173 4ChoiceRANK4 1.0 D 1 False 435 2175 4ChoiceRANK1 3.0 A 1 False 435 2176 4ChoiceRANK2 3.0 B 1 False 435 2177 4ChoiceRANK3 3.0 D 1 False 435 2178 4ChoiceRANK4 3.0 C 1 False 436 436 2180 4ChoiceRANK1 2.0 A 1 False 436 2181 4ChoiceRANK2 2.0 B 1 False 436 2182 4ChoiceRANK3 2.0 C 1 False 436 2183 4ChoiceRANK4 2.0 D 1 False 437 437 2185 4ChoiceRANK1 3.0 B 1 False 437 2186 4ChoiceRANK2 3.0 A 1 False 437 2187 4ChoiceRANK3 3.0 D 1 False 437 2188 4ChoiceRANK4 3.0 C 1 False 438 438 2190 4ChoiceRANK1 2.0 A 1 False 438 2191 4ChoiceRANK2 2.0 B 1 False 438 2192 4ChoiceRANK3 2.0 C 1 False 438 2193 4ChoiceRANK4 2.0 D 1 False 439 439 2195 4ChoiceRANK1 2.0 D 1 False 439 2196 4ChoiceRANK2 2.0 C 1 False 439 2197 4ChoiceRANK3 2.0 A 1 False 439 2198 4ChoiceRANK4 2.0 B 1 False 440 440 2200 4ChoiceRANK1 2.0 A 1 False 440 2201 4ChoiceRANK2 2.0 B 1 False 440 2202 4ChoiceRANK3 2.0 C 1 False 440 2203 4ChoiceRANK4 2.0 D 1 False 441 441 2205 4ChoiceRANK1 2.0 D 1 False 441 2206 4ChoiceRANK2 2.0 B 1 False 441 2207 4ChoiceRANK3 2.0 C 1 False 441 2208 4ChoiceRANK4 2.0 A 1 False 442 442 2210 4ChoiceRANK1 3.0 D 1 False 442 2211 4ChoiceRANK2 3.0 A 1 False 442 2212 4ChoiceRANK3 3.0 C 1 False 442 2213 4ChoiceRANK4 3.0 B 1 False 443 443 2215 4ChoiceRANK1 2.0 B 1 False 443 2216 4ChoiceRANK2 2.0 A 1 False 443 2217 4ChoiceRANK3 2.0 D 1 False 443 2218 4ChoiceRANK4 2.0 C 1 False 444 444 2220 4ChoiceRANK1

2.0 B 1 False

```
444 2221 4ChoiceRANK2
2.0 D 1 False
444 2222 4ChoiceRANK3
2.0 C 1 False
444 2223 4ChoiceRANK4
2.0 A 1 False
445 2225 4ChoiceRANK1
2.0 C 1
445 2226 4ChoiceRANK2
2.0 A 1
445 2227 4ChoiceRANK3
2.0 B 1
445 2228 4ChoiceRANK4
error
446
446 2230 4ChoiceRANK1
2.0 A 1 False
446 2231 4ChoiceRANK2
2.0 B 1 False
446 2232 4ChoiceRANK3
2.0 D 1 False
446 2233 4ChoiceRANK4
2.0 C 1 False
447
447 2235 4ChoiceRANK1
3.0 A 1 False
447 2236 4ChoiceRANK2
3.0 B 1 False
447 2237 4ChoiceRANK3
3.0 D 1 False
447 2238 4ChoiceRANK4
3.0 C 1 False
448
448 2240 4ChoiceRANK1
2.0 C 1 False
448 2241 4ChoiceRANK2
2.0 D 1 False
448 2242 4ChoiceRANK3
2.0 A 1 False
448 2243 4ChoiceRANK4
2.0 B 1 False
449
449 2245 4ChoiceRANK1
4.0 C 1 False
449 2246 4ChoiceRANK2
4.0 B 1 False
449 2247 4ChoiceRANK3
4.0 D 1 False
449 2248 4ChoiceRANK4
4.0 A 1 False
450
450 2250 4ChoiceRANK1
2.0 C 1
450 2251 4ChoiceRANK2
2.0 B 1
450 2252 4ChoiceRANK3
2.0 A 1
450 2253 4ChoiceRANK4
error
451
451 2255 4ChoiceRANK1
3.0 D 1 False
451 2256 4ChoiceRANK2
3.0 C 1 False
451 2257 4ChoiceRANK3
3.0 B 1 False
451 2258 4ChoiceRANK4
3.0 A 1 False
452 2260 4ChoiceRANK1
3.0 D 1 False
452 2261 4ChoiceRANK2
3.0 B 1 False
452 2262 4ChoiceRANK3
3.0 C 1 False
452 2263 4ChoiceRANK4
3.0 A 1 False
453
453 2265 4ChoiceRANK1
4.0 C 1 False
453 2266 4ChoiceRANK2
4.0 D 1 False
453 2267 4ChoiceRANK3
4.0 B 1 False
453 2268 4ChoiceRANK4
4.0 A 1 False
454
454 2270 4ChoiceRANK1
2.0 B 1 False
454 2271 4ChoiceRANK2
2.0 A 1 False
454 2272 4ChoiceRANK3
```

```
2.0 D 1 False
454 2273 4ChoiceRANK4
2.0 C 1 False
455 2275 4ChoiceRANK1
3.0 C 1
455 2276 4ChoiceRANK2
3.0 D 1
455 2277 4ChoiceRANK3
3.0 B 1
455 2278 4ChoiceRANK4
3.0 A 1
456
456 2280 4ChoiceRANK1
2.0 B 1
456 2281 4ChoiceRANK2
2.0 A 1
456 2282 4ChoiceRANK3
2.0 C 1
456 2283 4ChoiceRANK4
error
457
457 2285 4ChoiceRANK1
2.0 C 1
457 2286 4ChoiceRANK2
2.0 B 1
457 2287 4ChoiceRANK3
2.0 A 1
457 2288 4ChoiceRANK4
error
458
458 2290 4ChoiceRANK1
4.0 C 1 False
458 2291 4ChoiceRANK2
4.0 D 1 False
458 2292 4ChoiceRANK3
4.0 A 1 False
458 2293 4ChoiceRANK4
4.0 B 1 False
459
459 2295 4ChoiceRANK1
1.0 B 1 False
459 2296 4ChoiceRANK2
1.0 A 1 False
459 2297 4ChoiceRANK3
1.0 D 1 False
459 2298 4ChoiceRANK4
1.0 C 1 False
460
460 2300 4ChoiceRANK1
2.0 A 1 False
460 2301 4ChoiceRANK2
2.0 B 1 False
460 2302 4ChoiceRANK3
2.0 D 1 False
460 2303 4ChoiceRANK4
2.0 C 1 False
461 2305 4ChoiceRANK1
3.0 B 1 False
461 2306 4ChoiceRANK2
3.0 A 1 False
461 2307 4ChoiceRANK3
3.0 D 1 False
461 2308 4ChoiceRANK4
3.0 C 1 False
462 2310 4ChoiceRANK1
2.0 B 1 False
462 2311 4ChoiceRANK2
2.0 D 1 False
462 2312 4ChoiceRANK3
2.0 A 1 False
462 2313 4ChoiceRANK4
2.0 C 1 False
463
463 2315 4ChoiceRANK1
2.0 B 1 False
463 2316 4ChoiceRANK2
2.0 A 1 False
463 2317 4ChoiceRANK3
2.0 D 1 False
463 2318 4ChoiceRANK4
2.0 C 1 False
464 2320 4ChoiceRANK1
3.0 D 1
464 2321 4ChoiceRANK2
3.0 B 1
464 2322 4ChoiceRANK3
3.0 A 1
```

464 2323 4ChoiceRANK4 3.0 C 1

```
465
465 2325 4ChoiceRANK1
3.0 C 1
465 2326 4ChoiceRANK2
3.0 A 1
465 2327 4ChoiceRANK3
3.0 D 1
465 2328 4ChoiceRANK4
3.0 B 1
466 2330 4ChoiceRANK1
2.0 D 1 False
466 2331 4ChoiceRANK2
2.0 A 1 False
466 2332 4ChoiceRANK3
2.0 B 1 False
466 2333 4ChoiceRANK4
2.0 C 1 False
467
467 2335 4ChoiceRANK1
3.0 A 1 False
467 2336 4ChoiceRANK2
3.0 D 1 False
467 2337 4ChoiceRANK3
3.0 B 1 False
467 2338 4ChoiceRANK4
3.0 C 1 False
468
468 2340 4ChoiceRANK1
1.0 B 1 False
468 2341 4ChoiceRANK2
1.0 D 1 False
468 2342 4ChoiceRANK3
1.0 A 1 False
468 2343 4ChoiceRANK4
1.0 C 1 False
469
469 2345 4ChoiceRANK1
2.0 D 1 False
469 2346 4ChoiceRANK2
2.0 B 1 False
469 2347 4ChoiceRANK3
2.0 A 1 False
469 2348 4ChoiceRANK4
2.0 C 1 False
470
470 2350 4ChoiceRANK1
3.0 B 1 False
470 2351 4ChoiceRANK2
3.0 A 1 False
470 2352 4ChoiceRANK3
3.0 D 1 False
470 2353 4ChoiceRANK4
3.0 C 1 False
471 2355 4ChoiceRANK1
3.0 D 1 False
471 2356 4ChoiceRANK2
3.0 C 1 False
471 2357 4ChoiceRANK3
3.0 B 1 False
471 2358 4ChoiceRANK4
3.0 A 1 False
472
472 2360 4ChoiceRANK1
1.0 B 1
472 2361 4ChoiceRANK2
472 2362 4ChoiceRANK3
472 2363 4ChoiceRANK4
error
473
473 2365 4ChoiceRANK1
2.0 A 1
473 2366 4ChoiceRANK2
2.0 C 1
473 2367 4ChoiceRANK3
2.0 B 1
473 2368 4ChoiceRANK4
error
474
474 2370 4ChoiceRANK1
4.0 C 1 False
474 2371 4ChoiceRANK2
4.0 D 1 False
474 2372 4ChoiceRANK3
4.0 B 1 False
474 2373 4ChoiceRANK4
4.0 A 1 False
475
475 2375 4ChoiceRANK1
```

2.0 D 1 False

```
475 2376 4ChoiceRANK2
2.0 A 1 False
475 2377 4ChoiceRANK3
2.0 C 1 False
475 2378 4ChoiceRANK4
2.0 B 1 False
476 2380 4ChoiceRANK1
1.0 C 1 False
476 2381 4ChoiceRANK2
1.0 D 1 False
476 2382 4ChoiceRANK3
1.0 A 1 False
476 2383 4ChoiceRANK4
1.0 B 1 False
477
477 2385 4ChoiceRANK1
4.0 D 1 False
477 2386 4ChoiceRANK2
4.0 B 1 False
477 2387 4ChoiceRANK3
4.0 C 1 False
477 2388 4ChoiceRANK4
4.0 A 1 False
478 2390 4ChoiceRANK1
1.0 B 1 False
478 2391 4ChoiceRANK2
1.0 A 1 False
478 2392 4ChoiceRANK3
1.0 D 1 False
478 2393 4ChoiceRANK4
1.0 C 1 False
479
479 2395 4ChoiceRANK1
3.0 B 1 False
479 2396 4ChoiceRANK2
3.0 A 1 False
479 2397 4ChoiceRANK3
3.0 D 1 False
479 2398 4ChoiceRANK4
3.0 C 1 False
480
480 2400 4ChoiceRANK1
1.0 B 1 False
480 2401 4ChoiceRANK2
1.0 D 1 False
480 2402 4ChoiceRANK3
1.0 A 1 False
480 2403 4ChoiceRANK4
1.0 C 1 False
481
481 2405 4ChoiceRANK1
3.0 A 1 False
481 2406 4ChoiceRANK2
3.0 B 1 False
481 2407 4ChoiceRANK3
3.0 D 1 False
481 2408 4ChoiceRANK4
3.0 C 1 False
482
482 2410 4ChoiceRANK1
1.0 A 1 False
482 2411 4ChoiceRANK2
1.0 B 1 False
482 2412 4ChoiceRANK3
1.0 D 1 False
482 2413 4ChoiceRANK4
1.0 C 1 False
483 2415 4ChoiceRANK1
2.0 A 1
483 2416 4ChoiceRANK2
2.0 B 1
483 2417 4ChoiceRANK3
2.0 C 1
483 2418 4ChoiceRANK4
error
484
484 2420 4ChoiceRANK1
3.0 B 1 False
484 2421 4ChoiceRANK2
3.0 A 1 False
484 2422 4ChoiceRANK3
3.0 D 1 False
484 2423 4ChoiceRANK4
3.0 C 1 False
485
485 2425 4ChoiceRANK1
3.0 D 1
485 2426 4ChoiceRANK2
3.0 B 1
```

485 2427 4ChoiceRANK3

```
3.0 A 1
485 2428 4ChoiceRANK4
3.0 C 1
486
486 2430 4ChoiceRANK1
4.0 A 1 False
486 2431 4ChoiceRANK2
4.0 B 1 False
486 2432 4ChoiceRANK3
4.0 C 1 False
486 2433 4ChoiceRANK4
4.0 D 1 False
487
487 2435 4ChoiceRANK1
2.0 D 1 False
487 2436 4ChoiceRANK2
2.0 A 1 False
487 2437 4ChoiceRANK3
2.0 B 1 False
487 2438 4ChoiceRANK4
2.0 C 1 False
488
488 2440 4ChoiceRANK1
2.0 A 1 False
488 2441 4ChoiceRANK2
2.0 C 1 False
488 2442 4ChoiceRANK3
2.0 B 1 False
488 2443 4ChoiceRANK4
2.0 D 1 False
489
489 2445 4ChoiceRANK1
3.0 D 1
489 2446 4ChoiceRANK2
3.0 B 1
489 2447 4ChoiceRANK3
3.0 A 1
489 2448 4ChoiceRANK4
3.0 C 1
490
490 2450 4ChoiceRANK1
1.0 A 1 False
490 2451 4ChoiceRANK2
1.0 B 1 False
490 2452 4ChoiceRANK3
1.0 D 1 False
490 2453 4ChoiceRANK4
1.0 C 1 False
491
491 2455 4ChoiceRANK1
2.0 A 1 False
491 2456 4ChoiceRANK2
2.0 C 1 False
491 2457 4ChoiceRANK3
2.0 D 1 False
491 2458 4ChoiceRANK4
2.0 B 1 False
492 2460 4ChoiceRANK1
3.0 A 1 False
492 2461 4ChoiceRANK2
3.0 B 1 False
492 2462 4ChoiceRANK3
3.0 D 1 False
492 2463 4ChoiceRANK4
3.0 C 1 False
493
493 2465 4ChoiceRANK1
3.0 D 1 False
493 2466 4ChoiceRANK2
3.0 C 1 False
493 2467 4ChoiceRANK3
3.0 B 1 False
493 2468 4ChoiceRANK4
3.0 A 1 False
494
494 2470 4ChoiceRANK1
2.0 C 1 False
494 2471 4ChoiceRANK2
2.0 D 1 False
494 2472 4ChoiceRANK3
2.0 B 1 False
494 2473 4ChoiceRANK4
2.0 A 1 False
495
495 2475 4ChoiceRANK1
2.0 A 1
495 2476 4ChoiceRANK2
2.0 C 1
495 2477 4ChoiceRANK3
2.0 B 1
```

495 2478 4ChoiceRANK4 error

```
496
496 2480 4ChoiceRANK1
3.0 B 1
496 2481 4ChoiceRANK2
3.0 A 1
496 2482 4ChoiceRANK3
3.0 C 1
496 2483 4ChoiceRANK4
3.0 D 1
497
497 2485 4ChoiceRANK1
3.0 C 1
497 2486 4ChoiceRANK2
3.0 D 1
497 2487 4ChoiceRANK3
3.0 B 1
497 2488 4ChoiceRANK4
3.0 A 1
498
498 2490 4ChoiceRANK1
2.0 D 1 False
498 2491 4ChoiceRANK2
2.0 C 1 False
498 2492 4ChoiceRANK3
2.0 B 1 False
498 2493 4ChoiceRANK4
2.0 A 1 False
499
499 2495 4ChoiceRANK1
2.0 A 1 False
499 2496 4ChoiceRANK2
2.0 B 1 False
499 2497 4ChoiceRANK3
2.0 C 1 False
499 2498 4ChoiceRANK4
2.0 D 1 False
500
500 2500 4ChoiceRANK1
3.0 D 1 False
500 2501 4ChoiceRANK2
3.0 B 1 False
500 2502 4ChoiceRANK3
3.0 C 1 False
500 2503 4ChoiceRANK4
3.0 A 1 False
501
501 2505 4ChoiceRANK1
2.0 A 1
501 2506 4ChoiceRANK2
2.0 C 1
501 2507 4ChoiceRANK3
2.0 B 1
501 2508 4ChoiceRANK4
502
502 2510 4ChoiceRANK1
2.0 D 1 False
502 2511 4ChoiceRANK2
2.0 C 1 False
502 2512 4ChoiceRANK3
2.0 B 1 False
502 2513 4ChoiceRANK4
2.0 A 1 False
503
503 2515 4ChoiceRANK1
2.0 D 1 False
503 2516 4ChoiceRANK2
2.0 B 1 False
503 2517 4ChoiceRANK3
2.0 C 1 False
503 2518 4ChoiceRANK4
2.0 A 1 False
504
504 2520 4ChoiceRANK1
2.0 A 1 False
504 2521 4ChoiceRANK2
2.0 B 1 False
504 2522 4ChoiceRANK3
2.0 C 1 False
504 2523 4ChoiceRANK4
2.0 D 1 False
505
505 2525 4ChoiceRANK1
3.0 B 1 False
505 2526 4ChoiceRANK2
3.0 C 1 False
505 2527 4ChoiceRANK3
3.0 A 1 False
505 2528 4ChoiceRANK4
3.0 D 1 False
506
506 2530 4ChoiceRANK1
```

3.0 D 1 False

506 2531 4ChoiceRANK2
3.0 A 1 False
506 2532 4ChoiceRANK3
3.0 B 1 False
506 2533 4ChoiceRANK4
3.0 C 1 False
0 out of 2028

```
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW 9\AppData\Local\Temp\ipykernel 22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW 9\AppData\Local\Temp\ipykernel 22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance.  Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df new.at[i,'var.'+cols] = df1 to copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
```

df\_new.at[i,'var.'+cols] = df1\_to\_copy.at[0,cols]

```
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW 9\AppData\Local\Temp\ipykernel 22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df new.at[i,'var.'+cols] = df1 to copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
```

C:\Users\THW\_9\AppData\Local\Temp\ipykernel\_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i

```
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance.  Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df new.at[i,'var.'+cols] = df1 to copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW 9\AppData\Local\Temp\ipykernel 22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW 9\AppData\Local\Temp\ipykernel 22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
```

C:\Users\THW\_9\AppData\Local\Temp\ipykernel\_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi

```
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW 9\AppData\Local\Temp\ipykernel 22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW 9\AppData\Local\Temp\ipykernel 22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
```

ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`

```
df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
```

```
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW 9\AppData\Local\Temp\ipykernel 22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_22728\3629021164.py:119: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
```

```
1 out of 2028
2 out of 2028
3 out of 2028
4 out of 2028
5 out of 2028
6 out of 2028
7 out of 2028
8 out of 2028
9 out of 2028
10 out of 2028
11 out of 2028
12 out of 2028
13 out of 2028
14 out of 2028
15 out of 2028
16 out of 2028
17 out of 2028
18 out of 2028
19 out of 2028
20 out of 2028
21 out of 2028
22 out of 2028
23 out of 2028
24 out of 2028
25 out of 2028
26 out of 2028
27 out of 2028
28 out of 2028
29 out of 2028
30 out of 2028
31 out of 2028
32 out of 2028
33 out of 2028
34 out of 2028
35 out of 2028
36 out of 2028
37 out of 2028
38 out of 2028
39 out of 2028
40 out of 2028
41 out of 2028
42 out of 2028
43 out of 2028
44 out of 2028
45 out of 2028
46 out of 2028
47 out of 2028
48 out of 2028
49 out of 2028
50 out of 2028
51 out of 2028
52 out of 2028
53 out of 2028
54 out of 2028
55 out of 2028
56 out of 2028
57 out of 2028
58 out of 2028
59 out of 2028
60 out of 2028
61 out of 2028
62 out of 2028
63 out of 2028
64 out of 2028
65 out of 2028
66 out of 2028
67 out of 2028
68 out of 2028
69 out of 2028
70 out of 2028
71 out of 2028
72 out of 2028
73 out of 2028
74 out of 2028
75 out of 2028
76 out of 2028
77 out of 2028
78 out of 2028
79 out of 2028
80 out of 2028
81 out of 2028
82 out of 2028
83 out of 2028
84 out of 2028
85 out of 2028
86 out of 2028
87 out of 2028
88 out of 2028
89 out of 2028
90 out of 2028
91 out of 2028
```

92 out of 2028 93 out of 2028

```
94 out of 2028
95 out of 2028
96 out of 2028
97 out of 2028
98 out of 2028
99 out of 2028
100 out of 2028
101 out of 2028
102 out of 2028
103 out of 2028
104 out of 2028
105 out of 2028
106 out of 2028
107 out of 2028
108 out of 2028
109 out of 2028
110 out of 2028
111 out of 2028
112 out of 2028
113 out of 2028
114 out of 2028
115 out of 2028
116 out of 2028
117 out of 2028
118 out of 2028
119 out of 2028
120 out of 2028
121 out of 2028
122 out of 2028
123 out of 2028
124 out of 2028
125 out of 2028
126 out of 2028
127 out of 2028
128 out of 2028
129 out of 2028
130 out of 2028
131 out of 2028
132 out of 2028
133 out of 2028
134 out of 2028
135 out of 2028
136 out of 2028
137 out of 2028
138 out of 2028
139 out of 2028
140 out of 2028
141 out of 2028
142 out of 2028
143 out of 2028
144 out of 2028
145 out of 2028
146 out of 2028
147 out of 2028
148 out of 2028
149 out of 2028
150 out of 2028
151 out of 2028
152 out of 2028
153 out of 2028
154 out of 2028
155 out of 2028
156 out of 2028
157 out of 2028
158 out of 2028
159 out of 2028
160 out of 2028
161 out of 2028
162 out of 2028
163 out of 2028
164 out of 2028
165 out of 2028
166 out of 2028
167 out of 2028
168 out of 2028
169 out of 2028
170 out of 2028
171 out of 2028
172 out of 2028
173 out of 2028
174 out of 2028
175 out of 2028
176 out of 2028
177 out of 2028
178 out of 2028
179 out of 2028
180 out of 2028
181 out of 2028
182 out of 2028
183 out of 2028
184 out of 2028
185 out of 2028
```

```
187 out of 2028
188 out of 2028
189 out of 2028
190 out of 2028
191 out of 2028
192 out of 2028
193 out of 2028
194 out of 2028
195 out of 2028
196 out of 2028
197 out of 2028
198 out of 2028
199 out of 2028
200 out of 2028
201 out of 2028
202 out of 2028
203 out of 2028
204 out of 2028
205 out of 2028
206 out of 2028
207 out of 2028
208 out of 2028
209 out of 2028
210 out of 2028
211 out of 2028
212 out of 2028
213 out of 2028
214 out of 2028
215 out of 2028
216 out of 2028
217 out of 2028
218 out of 2028
219 out of 2028
220 out of 2028
221 out of 2028
222 out of 2028
223 out of 2028
224 out of 2028
225 out of 2028
226 out of 2028
227 out of 2028
228 out of 2028
229 out of 2028
230 out of 2028
231 out of 2028
232 out of 2028
233 out of 2028
234 out of 2028
235 out of 2028
236 out of 2028
237 out of 2028
238 out of 2028
239 out of 2028
240 out of 2028
241 out of 2028
242 out of 2028
243 out of 2028
244 out of 2028
245 out of 2028
246 out of 2028
247 out of 2028
248 out of 2028
249 out of 2028
250 out of 2028
251 out of 2028
252 out of 2028
253 out of 2028
254 out of 2028
255 out of 2028
256 out of 2028
257 out of 2028
258 out of 2028
259 out of 2028
260 out of 2028
261 out of 2028
262 out of 2028
263 out of 2028
264 out of 2028
265 out of 2028
266 out of 2028
267 out of 2028
268 out of 2028
269 out of 2028
270 out of 2028
271 out of 2028
272 out of 2028
273 out of 2028
274 out of 2028
275 out of 2028
276 out of 2028
277 out of 2028
278 out of 2028
```

```
280 out of 2028
281 out of 2028
282 out of 2028
283 out of 2028
284 out of 2028
285 out of 2028
286 out of 2028
287 out of 2028
288 out of 2028
289 out of 2028
290 out of 2028
291 out of 2028
292 out of 2028
293 out of 2028
294 out of 2028
295 out of 2028
296 out of 2028
297 out of 2028
298 out of 2028
299 out of 2028
300 out of 2028
301 out of 2028
302 out of 2028
303 out of 2028
304 out of 2028
305 out of 2028
306 out of 2028
307 out of 2028
308 out of 2028
309 out of 2028
310 out of 2028
311 out of 2028
312 out of 2028
313 out of 2028
314 out of 2028
315 out of 2028
316 out of 2028
317 out of 2028
318 out of 2028
319 out of 2028
320 out of 2028
321 out of 2028
322 out of 2028
323 out of 2028
324 out of 2028
325 out of 2028
326 out of 2028
327 out of 2028
328 out of 2028
329 out of 2028
330 out of 2028
331 out of 2028
332 out of 2028
333 out of 2028
334 out of 2028
335 out of 2028
336 out of 2028
337 out of 2028
338 out of 2028
339 out of 2028
340 out of 2028
341 out of 2028
342 out of 2028
343 out of 2028
344 out of 2028
345 out of 2028
346 out of 2028
347 out of 2028
348 out of 2028
349 out of 2028
350 out of 2028
351 out of 2028
352 out of 2028
353 out of 2028
354 out of 2028
355 out of 2028
356 out of 2028
357 out of 2028
358 out of 2028
359 out of 2028
360 out of 2028
361 out of 2028
362 out of 2028
363 out of 2028
364 out of 2028
365 out of 2028
366 out of 2028
367 out of 2028
368 out of 2028
369 out of 2028
370 out of 2028
371 out of 2028
```

```
373 out of 2028
374 out of 2028
375 out of 2028
376 out of 2028
377 out of 2028
378 out of 2028
379 out of 2028
380 out of 2028
381 out of 2028
382 out of 2028
383 out of 2028
384 out of 2028
385 out of 2028
386 out of 2028
387 out of 2028
388 out of 2028
389 out of 2028
390 out of 2028
391 out of 2028
392 out of 2028
393 out of 2028
394 out of 2028
395 out of 2028
396 out of 2028
397 out of 2028
398 out of 2028
399 out of 2028
400 out of 2028
401 out of 2028
402 out of 2028
403 out of 2028
404 out of 2028
405 out of 2028
406 out of 2028
407 out of 2028
408 out of 2028
409 out of 2028
410 out of 2028
411 out of 2028
412 out of 2028
413 out of 2028
414 out of 2028
415 out of 2028
416 out of 2028
417 out of 2028
418 out of 2028
419 out of 2028
420 out of 2028
421 out of 2028
422 out of 2028
423 out of 2028
424 out of 2028
425 out of 2028
426 out of 2028
427 out of 2028
428 out of 2028
429 out of 2028
430 out of 2028
431 out of 2028
432 out of 2028
433 out of 2028
434 out of 2028
435 out of 2028
436 out of 2028
437 out of 2028
438 out of 2028
439 out of 2028
440 out of 2028
441 out of 2028
442 out of 2028
443 out of 2028
444 out of 2028
445 out of 2028
446 out of 2028
447 out of 2028
448 out of 2028
449 out of 2028
450 out of 2028
451 out of 2028
452 out of 2028
453 out of 2028
454 out of 2028
455 out of 2028
456 out of 2028
457 out of 2028
458 out of 2028
459 out of 2028
460 out of 2028
461 out of 2028
462 out of 2028
463 out of 2028
464 out of 2028
```

```
466 out of 2028
467 out of 2028
468 out of 2028
469 out of 2028
470 out of 2028
471 out of 2028
472 out of 2028
473 out of 2028
474 out of 2028
475 out of 2028
476 out of 2028
477 out of 2028
478 out of 2028
479 out of 2028
480 out of 2028
481 out of 2028
482 out of 2028
483 out of 2028
484 out of 2028
485 out of 2028
486 out of 2028
487 out of 2028
488 out of 2028
489 out of 2028
490 out of 2028
491 out of 2028
492 out of 2028
493 out of 2028
494 out of 2028
495 out of 2028
496 out of 2028
497 out of 2028
498 out of 2028
499 out of 2028
500 out of 2028
501 out of 2028
502 out of 2028
503 out of 2028
504 out of 2028
505 out of 2028
506 out of 2028
507 out of 2028
508 out of 2028
509 out of 2028
510 out of 2028
511 out of 2028
512 out of 2028
513 out of 2028
514 out of 2028
515 out of 2028
516 out of 2028
517 out of 2028
518 out of 2028
519 out of 2028
520 out of 2028
521 out of 2028
522 out of 2028
523 out of 2028
524 out of 2028
525 out of 2028
526 out of 2028
527 out of 2028
528 out of 2028
529 out of 2028
530 out of 2028
531 out of 2028
532 out of 2028
533 out of 2028
534 out of 2028
535 out of 2028
536 out of 2028
537 out of 2028
538 out of 2028
539 out of 2028
540 out of 2028
541 out of 2028
542 out of 2028
543 out of 2028
544 out of 2028
545 out of 2028
546 out of 2028
547 out of 2028
548 out of 2028
549 out of 2028
550 out of 2028
551 out of 2028
552 out of 2028
553 out of 2028
554 out of 2028
555 out of 2028
556 out of 2028
557 out of 2028
```

```
559 out of 2028
560 out of 2028
561 out of 2028
562 out of 2028
563 out of 2028
564 out of 2028
565 out of 2028
566 out of 2028
567 out of 2028
568 out of 2028
569 out of 2028
570 out of 2028
571 out of 2028
572 out of 2028
573 out of 2028
574 out of 2028
575 out of 2028
576 out of 2028
577 out of 2028
578 out of 2028
579 out of 2028
580 out of 2028
581 out of 2028
582 out of 2028
583 out of 2028
584 out of 2028
585 out of 2028
586 out of 2028
587 out of 2028
588 out of 2028
589 out of 2028
590 out of 2028
591 out of 2028
592 out of 2028
593 out of 2028
594 out of 2028
595 out of 2028
596 out of 2028
597 out of 2028
598 out of 2028
599 out of 2028
600 out of 2028
601 out of 2028
602 out of 2028
603 out of 2028
604 out of 2028
605 out of 2028
606 out of 2028
607 out of 2028
608 out of 2028
609 out of 2028
610 out of 2028
611 out of 2028
612 out of 2028
613 out of 2028
614 out of 2028
615 out of 2028
616 out of 2028
617 out of 2028
618 out of 2028
619 out of 2028
620 out of 2028
621 out of 2028
622 out of 2028
623 out of 2028
624 out of 2028
625 out of 2028
626 out of 2028
627 out of 2028
628 out of 2028
629 out of 2028
630 out of 2028
631 out of 2028
632 out of 2028
633 out of 2028
634 out of 2028
635 out of 2028
636 out of 2028
637 out of 2028
638 out of 2028
639 out of 2028
640 out of 2028
641 out of 2028
642 out of 2028
643 out of 2028
644 out of 2028
645 out of 2028
646 out of 2028
647 out of 2028
648 out of 2028
649 out of 2028
650 out of 2028
```

```
652 out of 2028
653 out of 2028
654 out of 2028
655 out of 2028
656 out of 2028
657 out of 2028
658 out of 2028
659 out of 2028
660 out of 2028
661 out of 2028
662 out of 2028
663 out of 2028
664 out of 2028
665 out of 2028
666 out of 2028
667 out of 2028
668 out of 2028
669 out of 2028
670 out of 2028
671 out of 2028
672 out of 2028
673 out of 2028
674 out of 2028
675 out of 2028
676 out of 2028
677 out of 2028
678 out of 2028
679 out of 2028
680 out of 2028
681 out of 2028
682 out of 2028
683 out of 2028
684 out of 2028
685 out of 2028
686 out of 2028
687 out of 2028
688 out of 2028
689 out of 2028
690 out of 2028
691 out of 2028
692 out of 2028
693 out of 2028
694 out of 2028
695 out of 2028
696 out of 2028
697 out of 2028
698 out of 2028
699 out of 2028
700 out of 2028
701 out of 2028
702 out of 2028
703 out of 2028
704 out of 2028
705 out of 2028
706 out of 2028
707 out of 2028
708 out of 2028
709 out of 2028
710 out of 2028
711 out of 2028
712 out of 2028
713 out of 2028
714 out of 2028
715 out of 2028
716 out of 2028
717 out of 2028
718 out of 2028
719 out of 2028
720 out of 2028
721 out of 2028
722 out of 2028
723 out of 2028
724 out of 2028
725 out of 2028
726 out of 2028
727 out of 2028
728 out of 2028
729 out of 2028
730 out of 2028
731 out of 2028
732 out of 2028
733 out of 2028
734 out of 2028
735 out of 2028
736 out of 2028
737 out of 2028
738 out of 2028
739 out of 2028
740 out of 2028
741 out of 2028
742 out of 2028
743 out of 2028
```

```
745 out of 2028
746 out of 2028
747 out of 2028
748 out of 2028
749 out of 2028
750 out of 2028
751 out of 2028
752 out of 2028
753 out of 2028
754 out of 2028
755 out of 2028
756 out of 2028
757 out of 2028
758 out of 2028
759 out of 2028
760 out of 2028
761 out of 2028
762 out of 2028
763 out of 2028
764 out of 2028
765 out of 2028
766 out of 2028
767 out of 2028
768 out of 2028
769 out of 2028
770 out of 2028
771 out of 2028
772 out of 2028
773 out of 2028
774 out of 2028
775 out of 2028
776 out of 2028
777 out of 2028
778 out of 2028
779 out of 2028
780 out of 2028
781 out of 2028
782 out of 2028
783 out of 2028
784 out of 2028
785 out of 2028
786 out of 2028
787 out of 2028
788 out of 2028
789 out of 2028
790 out of 2028
791 out of 2028
792 out of 2028
793 out of 2028
794 out of 2028
795 out of 2028
796 out of 2028
797 out of 2028
798 out of 2028
799 out of 2028
800 out of 2028
801 out of 2028
802 out of 2028
803 out of 2028
804 out of 2028
805 out of 2028
806 out of 2028
807 out of 2028
808 out of 2028
809 out of 2028
810 out of 2028
811 out of 2028
812 out of 2028
813 out of 2028
814 out of 2028
815 out of 2028
816 out of 2028
817 out of 2028
818 out of 2028
819 out of 2028
820 out of 2028
821 out of 2028
822 out of 2028
823 out of 2028
824 out of 2028
825 out of 2028
826 out of 2028
827 out of 2028
828 out of 2028
829 out of 2028
830 out of 2028
831 out of 2028
832 out of 2028
833 out of 2028
834 out of 2028
835 out of 2028
836 out of 2028
837 out of 2028
```

```
838 out of 2028
839 out of 2028
840 out of 2028
841 out of 2028
842 out of 2028
843 out of 2028
844 out of 2028
845 out of 2028
846 out of 2028
847 out of 2028
848 out of 2028
849 out of 2028
850 out of 2028
851 out of 2028
852 out of 2028
853 out of 2028
854 out of 2028
855 out of 2028
856 out of 2028
857 out of 2028
858 out of 2028
859 out of 2028
860 out of 2028
861 out of 2028
862 out of 2028
863 out of 2028
864 out of 2028
865 out of 2028
866 out of 2028
867 out of 2028
868 out of 2028
869 out of 2028
870 out of 2028
871 out of 2028
872 out of 2028
873 out of 2028
874 out of 2028
875 out of 2028
876 out of 2028
877 out of 2028
878 out of 2028
879 out of 2028
880 out of 2028
881 out of 2028
882 out of 2028
883 out of 2028
884 out of 2028
885 out of 2028
886 out of 2028
887 out of 2028
888 out of 2028
889 out of 2028
890 out of 2028
891 out of 2028
892 out of 2028
893 out of 2028
894 out of 2028
895 out of 2028
896 out of 2028
897 out of 2028
898 out of 2028
899 out of 2028
900 out of 2028
901 out of 2028
902 out of 2028
903 out of 2028
904 out of 2028
905 out of 2028
906 out of 2028
907 out of 2028
908 out of 2028
909 out of 2028
910 out of 2028
911 out of 2028
912 out of 2028
913 out of 2028
914 out of 2028
915 out of 2028
916 out of 2028
917 out of 2028
918 out of 2028
919 out of 2028
920 out of 2028
921 out of 2028
922 out of 2028
923 out of 2028
924 out of 2028
925 out of 2028
926 out of 2028
927 out of 2028
928 out of 2028
929 out of 2028
```

```
931 out of 2028
932 out of 2028
933 out of 2028
934 out of 2028
935 out of 2028
936 out of 2028
937 out of 2028
938 out of 2028
939 out of 2028
940 out of 2028
941 out of 2028
942 out of 2028
943 out of 2028
944 out of 2028
945 out of 2028
946 out of 2028
947 out of 2028
948 out of 2028
949 out of 2028
950 out of 2028
951 out of 2028
952 out of 2028
953 out of 2028
954 out of 2028
955 out of 2028
956 out of 2028
957 out of 2028
958 out of 2028
959 out of 2028
960 out of 2028
961 out of 2028
962 out of 2028
963 out of 2028
964 out of 2028
965 out of 2028
966 out of 2028
967 out of 2028
968 out of 2028
969 out of 2028
970 out of 2028
971 out of 2028
972 out of 2028
973 out of 2028
974 out of 2028
975 out of 2028
976 out of 2028
977 out of 2028
978 out of 2028
979 out of 2028
980 out of 2028
981 out of 2028
982 out of 2028
983 out of 2028
984 out of 2028
985 out of 2028
986 out of 2028
987 out of 2028
988 out of 2028
989 out of 2028
990 out of 2028
991 out of 2028
992 out of 2028
993 out of 2028
994 out of 2028
995 out of 2028
996 out of 2028
997 out of 2028
998 out of 2028
999 out of 2028
1000 out of 2028
1001 out of 2028
1002 out of 2028
1003 out of 2028
1004 out of 2028
1005 out of 2028
1006 out of 2028
1007 out of 2028
1008 out of 2028
1009 out of 2028
1010 out of 2028
1011 out of 2028
1012 out of 2028
1013 out of 2028
1014 out of 2028
1015 out of 2028
1016 out of 2028
1017 out of 2028
1018 out of 2028
1019 out of 2028
1020 out of 2028
1021 out of 2028
1022 out of 2028
1023 out of 2028
```

```
1024 out of 2028
1025 out of 2028
1026 out of 2028
1027 out of 2028
1028 out of 2028
1029 out of 2028
1030 out of 2028
1031 out of 2028
1032 out of 2028
1033 out of 2028
1034 out of 2028
1035 out of 2028
1036 out of 2028
1037 out of 2028
1038 out of 2028
1039 out of 2028
1040 out of 2028
1041 out of 2028
1042 out of 2028
1043 out of 2028
1044 out of 2028
1045 out of 2028
1046 out of 2028
1047 out of 2028
1048 out of 2028
1049 out of 2028
1050 out of 2028
1051 out of 2028
1052 out of 2028
1053 out of 2028
1054 out of 2028
1055 out of 2028
1056 out of 2028
1057 out of 2028
1058 out of 2028
1059 out of 2028
1060 out of 2028
1061 out of 2028
1062 out of 2028
1063 out of 2028
1064 out of 2028
1065 out of 2028
1066 out of 2028
1067 out of 2028
1068 out of 2028
1069 out of 2028
1070 out of 2028
1071 out of 2028
1072 out of 2028
1073 out of 2028
1074 out of 2028
1075 out of 2028
1076 out of 2028
1077 out of 2028
1078 out of 2028
1079 out of 2028
1080 out of 2028
1081 out of 2028
1082 out of 2028
1083 out of 2028
1084 out of 2028
1085 out of 2028
1086 out of 2028
1087 out of 2028
1088 out of 2028
1089 out of 2028
1090 out of 2028
1091 out of 2028
1092 out of 2028
1093 out of 2028
1094 out of 2028
1095 out of 2028
1096 out of 2028
1097 out of 2028
1098 out of 2028
1099 out of 2028
1100 out of 2028
1101 out of 2028
1102 out of 2028
1103 out of 2028
1104 out of 2028
1105 out of 2028
1106 out of 2028
1107 out of 2028
1108 out of 2028
1109 out of 2028
1110 out of 2028
1111 out of 2028
1112 out of 2028
1113 out of 2028
1114 out of 2028
1115 out of 2028
1116 out of 2028
```

```
1117 out of 2028
1118 out of 2028
1119 out of 2028
1120 out of 2028
1121 out of 2028
1122 out of 2028
1123 out of 2028
1124 out of 2028
1125 out of 2028
1126 out of 2028
1127 out of 2028
1128 out of 2028
1129 out of 2028
1130 out of 2028
1131 out of 2028
1132 out of 2028
1133 out of 2028
1134 out of 2028
1135 out of 2028
1136 out of 2028
1137 out of 2028
1138 out of 2028
1139 out of 2028
1140 out of 2028
1141 out of 2028
1142 out of 2028
1143 out of 2028
1144 out of 2028
1145 out of 2028
1146 out of 2028
1147 out of 2028
1148 out of 2028
1149 out of 2028
1150 out of 2028
1151 out of 2028
1152 out of 2028
1153 out of 2028
1154 out of 2028
1155 out of 2028
1156 out of 2028
1157 out of 2028
1158 out of 2028
1159 out of 2028
1160 out of 2028
1161 out of 2028
1162 out of 2028
1163 out of 2028
1164 out of 2028
1165 out of 2028
1166 out of 2028
1167 out of 2028
1168 out of 2028
1169 out of 2028
1170 out of 2028
1171 out of 2028
1172 out of 2028
1173 out of 2028
1174 out of 2028
1175 out of 2028
1176 out of 2028
1177 out of 2028
1178 out of 2028
1179 out of 2028
1180 out of 2028
1181 out of 2028
1182 out of 2028
1183 out of 2028
1184 out of 2028
1185 out of 2028
1186 out of 2028
1187 out of 2028
1188 out of 2028
1189 out of 2028
1190 out of 2028
1191 out of 2028
1192 out of 2028
1193 out of 2028
1194 out of 2028
1195 out of 2028
1196 out of 2028
1197 out of 2028
1198 out of 2028
1199 out of 2028
1200 out of 2028
1201 out of 2028
1202 out of 2028
1203 out of 2028
1204 out of 2028
1205 out of 2028
1206 out of 2028
1207 out of 2028
1208 out of 2028
```

```
1210 out of 2028
1211 out of 2028
1212 out of 2028
1213 out of 2028
1214 out of 2028
1215 out of 2028
1216 out of 2028
1217 out of 2028
1218 out of 2028
1219 out of 2028
1220 out of 2028
1221 out of 2028
1222 out of 2028
1223 out of 2028
1224 out of 2028
1225 out of 2028
1226 out of 2028
1227 out of 2028
1228 out of 2028
1229 out of 2028
1230 out of 2028
1231 out of 2028
1232 out of 2028
1233 out of 2028
1234 out of 2028
1235 out of 2028
1236 out of 2028
1237 out of 2028
1238 out of 2028
1239 out of 2028
1240 out of 2028
1241 out of 2028
1242 out of 2028
1243 out of 2028
1244 out of 2028
1245 out of 2028
1246 out of 2028
1247 out of 2028
1248 out of 2028
1249 out of 2028
1250 out of 2028
1251 out of 2028
1252 out of 2028
1253 out of 2028
1254 out of 2028
1255 out of 2028
1256 out of 2028
1257 out of 2028
1258 out of 2028
1259 out of 2028
1260 out of 2028
1261 out of 2028
1262 out of 2028
1263 out of 2028
1264 out of 2028
1265 out of 2028
1266 out of 2028
1267 out of 2028
1268 out of 2028
1269 out of 2028
1270 out of 2028
1271 out of 2028
1272 out of 2028
1273 out of 2028
1274 out of 2028
1275 out of 2028
1276 out of 2028
1277 out of 2028
1278 out of 2028
1279 out of 2028
1280 out of 2028
1281 out of 2028
1282 out of 2028
1283 out of 2028
1284 out of 2028
1285 out of 2028
1286 out of 2028
1287 out of 2028
1288 out of 2028
1289 out of 2028
1290 out of 2028
1291 out of 2028
1292 out of 2028
1293 out of 2028
1294 out of 2028
1295 out of 2028
1296 out of 2028
1297 out of 2028
1298 out of 2028
1299 out of 2028
1300 out of 2028
1301 out of 2028
1302 out of 2028
```

```
1303 out of 2028
1304 out of 2028
1305 out of 2028
1306 out of 2028
1307 out of 2028
1308 out of 2028
1309 out of 2028
1310 out of 2028
1311 out of 2028
1312 out of 2028
1313 out of 2028
1314 out of 2028
1315 out of 2028
1316 out of 2028
1317 out of 2028
1318 out of 2028
1319 out of 2028
1320 out of 2028
1321 out of 2028
1322 out of 2028
1323 out of 2028
1324 out of 2028
1325 out of 2028
1326 out of 2028
1327 out of 2028
1328 out of 2028
1329 out of 2028
1330 out of 2028
1331 out of 2028
1332 out of 2028
1333 out of 2028
1334 out of 2028
1335 out of 2028
1336 out of 2028
1337 out of 2028
1338 out of 2028
1339 out of 2028
1340 out of 2028
1341 out of 2028
1342 out of 2028
1343 out of 2028
1344 out of 2028
1345 out of 2028
1346 out of 2028
1347 out of 2028
1348 out of 2028
1349 out of 2028
1350 out of 2028
1351 out of 2028
1352 out of 2028
1353 out of 2028
1354 out of 2028
1355 out of 2028
1356 out of 2028
1357 out of 2028
1358 out of 2028
1359 out of 2028
1360 out of 2028
1361 out of 2028
1362 out of 2028
1363 out of 2028
1364 out of 2028
1365 out of 2028
1366 out of 2028
1367 out of 2028
1368 out of 2028
1369 out of 2028
1370 out of 2028
1371 out of 2028
1372 out of 2028
1373 out of 2028
1374 out of 2028
1375 out of 2028
1376 out of 2028
1377 out of 2028
1378 out of 2028
1379 out of 2028
1380 out of 2028
1381 out of 2028
1382 out of 2028
1383 out of 2028
1384 out of 2028
1385 out of 2028
1386 out of 2028
1387 out of 2028
1388 out of 2028
1389 out of 2028
1390 out of 2028
1391 out of 2028
1392 out of 2028
1393 out of 2028
1394 out of 2028
```

```
1396 out of 2028
1397 out of 2028
1398 out of 2028
1399 out of 2028
1400 out of 2028
1401 out of 2028
1402 out of 2028
1403 out of 2028
1404 out of 2028
1405 out of 2028
1406 out of 2028
1407 out of 2028
1408 out of 2028
1409 out of 2028
1410 out of 2028
1411 out of 2028
1412 out of 2028
1413 out of 2028
1414 out of 2028
1415 out of 2028
1416 out of 2028
1417 out of 2028
1418 out of 2028
1419 out of 2028
1420 out of 2028
1421 out of 2028
1422 out of 2028
1423 out of 2028
1424 out of 2028
1425 out of 2028
1426 out of 2028
1427 out of 2028
1428 out of 2028
1429 out of 2028
1430 out of 2028
1431 out of 2028
1432 out of 2028
1433 out of 2028
1434 out of 2028
1435 out of 2028
1436 out of 2028
1437 out of 2028
1438 out of 2028
1439 out of 2028
1440 out of 2028
1441 out of 2028
1442 out of 2028
1443 out of 2028
1444 out of 2028
1445 out of 2028
1446 out of 2028
1447 out of 2028
1448 out of 2028
1449 out of 2028
1450 out of 2028
1451 out of 2028
1452 out of 2028
1453 out of 2028
1454 out of 2028
1455 out of 2028
1456 out of 2028
1457 out of 2028
1458 out of 2028
1459 out of 2028
1460 out of 2028
1461 out of 2028
1462 out of 2028
1463 out of 2028
1464 out of 2028
1465 out of 2028
1466 out of 2028
1467 out of 2028
1468 out of 2028
1469 out of 2028
1470 out of 2028
1471 out of 2028
1472 out of 2028
1473 out of 2028
1474 out of 2028
1475 out of 2028
1476 out of 2028
1477 out of 2028
1478 out of 2028
1479 out of 2028
1480 out of 2028
1481 out of 2028
1482 out of 2028
1483 out of 2028
1484 out of 2028
1485 out of 2028
1486 out of 2028
1487 out of 2028
```

```
1489 out of 2028
1490 out of 2028
1491 out of 2028
1492 out of 2028
1493 out of 2028
1494 out of 2028
1495 out of 2028
1496 out of 2028
1497 out of 2028
1498 out of 2028
1499 out of 2028
1500 out of 2028
1501 out of 2028
1502 out of 2028
1503 out of 2028
1504 out of 2028
1505 out of 2028
1506 out of 2028
1507 out of 2028
1508 out of 2028
1509 out of 2028
1510 out of 2028
1511 out of 2028
1512 out of 2028
1513 out of 2028
1514 out of 2028
1515 out of 2028
1516 out of 2028
1517 out of 2028
1518 out of 2028
1519 out of 2028
1520 out of 2028
1521 out of 2028
1522 out of 2028
1523 out of 2028
1524 out of 2028
1525 out of 2028
1526 out of 2028
1527 out of 2028
1528 out of 2028
1529 out of 2028
1530 out of 2028
1531 out of 2028
1532 out of 2028
1533 out of 2028
1534 out of 2028
1535 out of 2028
1536 out of 2028
1537 out of 2028
1538 out of 2028
1539 out of 2028
1540 out of 2028
1541 out of 2028
1542 out of 2028
1543 out of 2028
1544 out of 2028
1545 out of 2028
1546 out of 2028
1547 out of 2028
1548 out of 2028
1549 out of 2028
1550 out of 2028
1551 out of 2028
1552 out of 2028
1553 out of 2028
1554 out of 2028
1555 out of 2028
1556 out of 2028
1557 out of 2028
1558 out of 2028
1559 out of 2028
1560 out of 2028
1561 out of 2028
1562 out of 2028
1563 out of 2028
1564 out of 2028
1565 out of 2028
1566 out of 2028
1567 out of 2028
1568 out of 2028
1569 out of 2028
1570 out of 2028
1571 out of 2028
1572 out of 2028
1573 out of 2028
1574 out of 2028
1575 out of 2028
1576 out of 2028
1577 out of 2028
1578 out of 2028
1579 out of 2028
1580 out of 2028
```

```
1582 out of 2028
1583 out of 2028
1584 out of 2028
1585 out of 2028
1586 out of 2028
1587 out of 2028
1588 out of 2028
1589 out of 2028
1590 out of 2028
1591 out of 2028
1592 out of 2028
1593 out of 2028
1594 out of 2028
1595 out of 2028
1596 out of 2028
1597 out of 2028
1598 out of 2028
1599 out of 2028
1600 out of 2028
1601 out of 2028
1602 out of 2028
1603 out of 2028
1604 out of 2028
1605 out of 2028
1606 out of 2028
1607 out of 2028
1608 out of 2028
1609 out of 2028
1610 out of 2028
1611 out of 2028
1612 out of 2028
1613 out of 2028
1614 out of 2028
1615 out of 2028
1616 out of 2028
1617 out of 2028
1618 out of 2028
1619 out of 2028
1620 out of 2028
1621 out of 2028
1622 out of 2028
1623 out of 2028
1624 out of 2028
1625 out of 2028
1626 out of 2028
1627 out of 2028
1628 out of 2028
1629 out of 2028
1630 out of 2028
1631 out of 2028
1632 out of 2028
1633 out of 2028
1634 out of 2028
1635 out of 2028
1636 out of 2028
1637 out of 2028
1638 out of 2028
1639 out of 2028
1640 out of 2028
1641 out of 2028
1642 out of 2028
1643 out of 2028
1644 out of 2028
1645 out of 2028
1646 out of 2028
1647 out of 2028
1648 out of 2028
1649 out of 2028
1650 out of 2028
1651 out of 2028
1652 out of 2028
1653 out of 2028
1654 out of 2028
1655 out of 2028
1656 out of 2028
1657 out of 2028
1658 out of 2028
1659 out of 2028
1660 out of 2028
1661 out of 2028
1662 out of 2028
1663 out of 2028
1664 out of 2028
1665 out of 2028
1666 out of 2028
1667 out of 2028
1668 out of 2028
1669 out of 2028
1670 out of 2028
1671 out of 2028
1672 out of 2028
1673 out of 2028
```

```
1675 out of 2028
1676 out of 2028
1677 out of 2028
1678 out of 2028
1679 out of 2028
1680 out of 2028
1681 out of 2028
1682 out of 2028
1683 out of 2028
1684 out of 2028
1685 out of 2028
1686 out of 2028
1687 out of 2028
1688 out of 2028
1689 out of 2028
1690 out of 2028
1691 out of 2028
1692 out of 2028
1693 out of 2028
1694 out of 2028
1695 out of 2028
1696 out of 2028
1697 out of 2028
1698 out of 2028
1699 out of 2028
1700 out of 2028
1701 out of 2028
1702 out of 2028
1703 out of 2028
1704 out of 2028
1705 out of 2028
1706 out of 2028
1707 out of 2028
1708 out of 2028
1709 out of 2028
1710 out of 2028
1711 out of 2028
1712 out of 2028
1713 out of 2028
1714 out of 2028
1715 out of 2028
1716 out of 2028
1717 out of 2028
1718 out of 2028
1719 out of 2028
1720 out of 2028
1721 out of 2028
1722 out of 2028
1723 out of 2028
1724 out of 2028
1725 out of 2028
1726 out of 2028
1727 out of 2028
1728 out of 2028
1729 out of 2028
1730 out of 2028
1731 out of 2028
1732 out of 2028
1733 out of 2028
1734 out of 2028
1735 out of 2028
1736 out of 2028
1737 out of 2028
1738 out of 2028
1739 out of 2028
1740 out of 2028
1741 out of 2028
1742 out of 2028
1743 out of 2028
1744 out of 2028
1745 out of 2028
1746 out of 2028
1747 out of 2028
1748 out of 2028
1749 out of 2028
1750 out of 2028
1751 out of 2028
1752 out of 2028
1753 out of 2028
1754 out of 2028
1755 out of 2028
1756 out of 2028
1757 out of 2028
1758 out of 2028
1759 out of 2028
1760 out of 2028
1761 out of 2028
1762 out of 2028
1763 out of 2028
1764 out of 2028
1765 out of 2028
1766 out of 2028
```

```
1768 out of 2028
1769 out of 2028
1770 out of 2028
1771 out of 2028
1772 out of 2028
1773 out of 2028
1774 out of 2028
1775 out of 2028
1776 out of 2028
1777 out of 2028
1778 out of 2028
1779 out of 2028
1780 out of 2028
1781 out of 2028
1782 out of 2028
1783 out of 2028
1784 out of 2028
1785 out of 2028
1786 out of 2028
1787 out of 2028
1788 out of 2028
1789 out of 2028
1790 out of 2028
1791 out of 2028
1792 out of 2028
1793 out of 2028
1794 out of 2028
1795 out of 2028
1796 out of 2028
1797 out of 2028
1798 out of 2028
1799 out of 2028
1800 out of 2028
1801 out of 2028
1802 out of 2028
1803 out of 2028
1804 out of 2028
1805 out of 2028
1806 out of 2028
1807 out of 2028
1808 out of 2028
1809 out of 2028
1810 out of 2028
1811 out of 2028
1812 out of 2028
1813 out of 2028
1814 out of 2028
1815 out of 2028
1816 out of 2028
1817 out of 2028
1818 out of 2028
1819 out of 2028
1820 out of 2028
1821 out of 2028
1822 out of 2028
1823 out of 2028
1824 out of 2028
1825 out of 2028
1826 out of 2028
1827 out of 2028
1828 out of 2028
1829 out of 2028
1830 out of 2028
1831 out of 2028
1832 out of 2028
1833 out of 2028
1834 out of 2028
1835 out of 2028
1836 out of 2028
1837 out of 2028
1838 out of 2028
1839 out of 2028
1840 out of 2028
1841 out of 2028
1842 out of 2028
1843 out of 2028
1844 out of 2028
1845 out of 2028
1846 out of 2028
1847 out of 2028
1848 out of 2028
1849 out of 2028
1850 out of 2028
1851 out of 2028
1852 out of 2028
1853 out of 2028
1854 out of 2028
1855 out of 2028
1856 out of 2028
1857 out of 2028
1858 out of 2028
1859 out of 2028
```

```
1861 out of 2028
1862 out of 2028
1863 out of 2028
1864 out of 2028
1865 out of 2028
1866 out of 2028
1867 out of 2028
1868 out of 2028
1869 out of 2028
1870 out of 2028
1871 out of 2028
1872 out of 2028
1873 out of 2028
1874 out of 2028
1875 out of 2028
1876 out of 2028
1877 out of 2028
1878 out of 2028
1879 out of 2028
1880 out of 2028
1881 out of 2028
1882 out of 2028
1883 out of 2028
1884 out of 2028
1885 out of 2028
1886 out of 2028
1887 out of 2028
1888 out of 2028
1889 out of 2028
1890 out of 2028
1891 out of 2028
1892 out of 2028
1893 out of 2028
1894 out of 2028
1895 out of 2028
1896 out of 2028
1897 out of 2028
1898 out of 2028
1899 out of 2028
1900 out of 2028
1901 out of 2028
1902 out of 2028
1903 out of 2028
1904 out of 2028
1905 out of 2028
1906 out of 2028
1907 out of 2028
1908 out of 2028
1909 out of 2028
1910 out of 2028
1911 out of 2028
1912 out of 2028
1913 out of 2028
1914 out of 2028
1915 out of 2028
1916 out of 2028
1917 out of 2028
1918 out of 2028
1919 out of 2028
1920 out of 2028
1921 out of 2028
1922 out of 2028
1923 out of 2028
1924 out of 2028
1925 out of 2028
1926 out of 2028
1927 out of 2028
1928 out of 2028
1929 out of 2028
1930 out of 2028
1931 out of 2028
1932 out of 2028
1933 out of 2028
1934 out of 2028
1935 out of 2028
1936 out of 2028
1937 out of 2028
1938 out of 2028
1939 out of 2028
1940 out of 2028
1941 out of 2028
1942 out of 2028
1943 out of 2028
1944 out of 2028
1945 out of 2028
1946 out of 2028
1947 out of 2028
1948 out of 2028
1949 out of 2028
1950 out of 2028
1951 out of 2028
1952 out of 2028
```

```
1954 out of 2028
1955 out of 2028
1956 out of 2028
1957 out of 2028
1958 out of 2028
1959 out of 2028
1960 out of 2028
1961 out of 2028
1962 out of 2028
1963 out of 2028
1964 out of 2028
1965 out of 2028
1966 out of 2028
1967 out of 2028
1968 out of 2028
1969 out of 2028
1970 out of 2028
1971 out of 2028
1972 out of 2028
1973 out of 2028
1974 out of 2028
1975 out of 2028
1976 out of 2028
1977 out of 2028
1978 out of 2028
1979 out of 2028
1980 out of 2028
1981 out of 2028
1982 out of 2028
1983 out of 2028
1984 out of 2028
1985 out of 2028
1986 out of 2028
1987 out of 2028
1988 out of 2028
1989 out of 2028
1990 out of 2028
1991 out of 2028
1992 out of 2028
1993 out of 2028
1994 out of 2028
1995 out of 2028
1996 out of 2028
1997 out of 2028
1998 out of 2028
1999 out of 2028
2000 out of 2028
2001 out of 2028
2002 out of 2028
2003 out of 2028
2004 out of 2028
2005 out of 2028
2006 out of 2028
2007 out of 2028
2008 out of 2028
2009 out of 2028
2010 out of 2028
2011 out of 2028
2012 out of 2028
2013 out of 2028
2014 out of 2028
2015 out of 2028
2016 out of 2028
2017 out of 2028
2018 out of 2028
2019 out of 2028
2020 out of 2028
2021 out of 2028
2022 out of 2028
2023 out of 2028
2024 out of 2028
2025 out of 2028
2026 out of 2028
```

## **Running for Pre-policy**

```
In [4]: #This is the variable of interest.
y_column = '4ChoiceRANK'

#this is for post-policy changes
post = 0

#for option choices
abcd = ['A','B','C','D']
abc = ['A','','B','C']
rankt = [1,2,3,4]

df1_to_run = df1

df_new = pd.DataFrame()
```

```
x = 0
for i in range(0,len(df1_to_run)):
    print(i)
   for rank in rankt:
        j = i + x #to index the dataframe
        x += 1
        df_new.at[j,'SN'] = df1_to_run.at[i,'SN']
       y_column_choice = y_column + str(rank)
        print(i, j, y_column_choice)
        doub = df1_to_run.at[i, 'DualIncome'] > 0
        sing = df1_to_run.at[i,'SingleIncome'] > 0
        ineli = df1_to_run.at[i,'SingleIncome'] == 3
        df_new.at[j,'Rank'] = rank
        #Processing for doubles
        if (doub and not sing):
            income = df1_to_run.at[i,'DualIncome']
            choice = abcd[
                int(df1_to_run.at[i,y_column_choice]) - 1 #To identify chocies by alphabets
            print(income, choice, post, ineli)
            #to get attributes from df2 which has the attributes
            tempdf2 = df2.loc[
                df2['DualIncome'] == income
                ].loc[
                    df2['Option'] == choice
                    ].loc[
                        df2['Post'] == post
                    ].reset_index()
            for cols in list(df2.columns)[:8]:
                df_new.at[j,cols] = tempdf2.at[0,cols] #copies and pastes the housing attribute into dataset
            df_new.at[j,y_column_choice] = 1
        #Repeated process for singles
        elif sing and not doub:
            income = df1_to_run.at[i,'SingleIncome']
            if not (df1_to_run.at[i,y_column_choice] > 0):
                print('error')
                continue
            if not ineli:
                choice = abc[
                    int(df1_to_run.at[i,y_column_choice]) - 1
            else:
                choice = abcd[
                    int(df1_to_run.at[i,y_column_choice]) - 1
            print(income,choice,post)
            tempdf2 = df2.loc[
                df2['SingleIncome'] == income
                ].loc[
                    df2['Option'] == choice
                    ].loc[
                        df2['Post'] == post
                    ].reset_index()
            for cols in list(df2.columns)[:8]:
                df_new.at[j,cols] = tempdf2.at[0,cols]
            df_new.at[j,y_column_choice] = 1
        else:
            print('error')
```

```
df_new = df_new.reset_index()

for i in range(0,len(df_new)):
    print(i, 'out of', len(df_new))

    df1_to_copy = df1_to_run.loc[df1_to_run['SN'] == df_new.at[i,'SN']].reset_index()

    for cols in df1_to_copy.columns[22:]:
        df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
```

```
2.0 D 0 False
0 2 4ChoiceRANK3
2.0 A 0 False
0 3 4ChoiceRANK4
2.0 C 0 False
1 5 4ChoiceRANK1
3.0 D 0
1 6 4ChoiceRANK2
3.0 B 0
1 7 4ChoiceRANK3
3.0 C 0
1 8 4ChoiceRANK4
3.0 A 0
2
2 10 4ChoiceRANK1
4.0 C 0 False
2 11 4ChoiceRANK2
4.0 A 0 False
2 12 4ChoiceRANK3
4.0 D 0 False
2 13 4ChoiceRANK4
4.0 B 0 False
3 15 4ChoiceRANK1
4.0 D 0 False
3 16 4ChoiceRANK2
4.0 C 0 False
3 17 4ChoiceRANK3
4.0 B 0 False
3 18 4ChoiceRANK4
4.0 A 0 False
4 20 4ChoiceRANK1
1.0 C 0 False
4 21 4ChoiceRANK2
1.0 A 0 False
4 22 4ChoiceRANK3
1.0 B 0 False
4 23 4ChoiceRANK4
1.0 D 0 False
5
5 25 4ChoiceRANK1
1.0 B 0 False
5 26 4ChoiceRANK2
1.0 D 0 False
5 27 4ChoiceRANK3
1.0 A 0 False
5 28 4ChoiceRANK4
1.0 C 0 False
6
6 30 4ChoiceRANK1
1.0 A 0 False
6 31 4ChoiceRANK2
1.0 C 0 False
6 32 4ChoiceRANK3
1.0 B 0 False
6 33 4ChoiceRANK4
1.0 D 0 False
7 35 4ChoiceRANK1
2.0 B 0
7 36 4ChoiceRANK2
2.0 C 0
7 37 4ChoiceRANK3
2.0 A 0
7 38 4ChoiceRANK4
error
8
8 40 4ChoiceRANK1
4.0 D 0 False
8 41 4ChoiceRANK2
4.0 C 0 False
8 42 4ChoiceRANK3
4.0 B 0 False
8 43 4ChoiceRANK4
4.0 A 0 False
9 45 4ChoiceRANK1
2.0 D 0 False
9 46 4ChoiceRANK2
2.0 C 0 False
9 47 4ChoiceRANK3
2.0 B 0 False
9 48 4ChoiceRANK4
2.0 A 0 False
10
10 50 4ChoiceRANK1
2.0 A 0 False
```

0

0 0 4ChoiceRANK1
2.0 B 0 False
0 1 4ChoiceRANK2

10 51 4ChoiceRANK2 2.0 B 0 False 10 52 4ChoiceRANK3 2.0 D 0 False 10 53 4ChoiceRANK4 2.0 C 0 False 11 11 55 4ChoiceRANK1 2.0 B 0 11 56 4ChoiceRANK2 2.0 A 0 11 57 4ChoiceRANK3 2.0 C 0 11 58 4ChoiceRANK4 error 12 12 60 4ChoiceRANK1 2.0 A 0 False 12 61 4ChoiceRANK2 2.0 C 0 False 12 62 4ChoiceRANK3 2.0 B 0 False 12 63 4ChoiceRANK4 2.0 D 0 False 13 65 4ChoiceRANK1 2.0 A 0 False 13 66 4ChoiceRANK2 2.0 C 0 False 13 67 4ChoiceRANK3 2.0 D 0 False 13 68 4ChoiceRANK4 2.0 B 0 False 14 14 70 4ChoiceRANK1 2.0 A 0 14 71 4ChoiceRANK2 2.0 C 0 14 72 4ChoiceRANK3 2.0 B 0 14 73 4ChoiceRANK4 error 15 15 75 4ChoiceRANK1 4.0 C 0 False 15 76 4ChoiceRANK2 4.0 D 0 False 15 77 4ChoiceRANK3 4.0 A 0 False 15 78 4ChoiceRANK4 4.0 B 0 False 16 16 80 4ChoiceRANK1 2.0 B 0 False 16 81 4ChoiceRANK2 2.0 D 0 False 16 82 4ChoiceRANK3 2.0 C 0 False 16 83 4ChoiceRANK4 2.0 A 0 False 17 17 85 4ChoiceRANK1 2.0 A 0 False 17 86 4ChoiceRANK2 2.0 B 0 False 17 87 4ChoiceRANK3 2.0 D 0 False 17 88 4ChoiceRANK4 2.0 C 0 False 18 90 4ChoiceRANK1 2.0 C 0 False 18 91 4ChoiceRANK2 2.0 D 0 False 18 92 4ChoiceRANK3 2.0 A 0 False 18 93 4ChoiceRANK4 2.0 B 0 False 19 19 95 4ChoiceRANK1 2.0 B 0 False 19 96 4ChoiceRANK2 2.0 A 0 False 19 97 4ChoiceRANK3 2.0 D 0 False 19 98 4ChoiceRANK4 2.0 C 0 False 20 20 100 4ChoiceRANK1 2.0 B 0 False 20 101 4ChoiceRANK2

2.0 D 0 False

20 102 4ChoiceRANK3

```
2.0 A 0 False
20 103 4ChoiceRANK4
2.0 C 0 False
21 105 4ChoiceRANK1
3.0 C 0
21 106 4ChoiceRANK2
3.0 B 0
21 107 4ChoiceRANK3
3.0 D 0
21 108 4ChoiceRANK4
3.0 A 0
22
22 110 4ChoiceRANK1
2.0 D 0 False
22 111 4ChoiceRANK2
2.0 B 0 False
22 112 4ChoiceRANK3
2.0 A 0 False
22 113 4ChoiceRANK4
2.0 C 0 False
23
23 115 4ChoiceRANK1
2.0 D 0 False
23 116 4ChoiceRANK2
2.0 C 0 False
23 117 4ChoiceRANK3
2.0 A 0 False
23 118 4ChoiceRANK4
2.0 B 0 False
24 120 4ChoiceRANK1
2.0 D 0 False
24 121 4ChoiceRANK2
2.0 B 0 False
24 122 4ChoiceRANK3
2.0 A 0 False
24 123 4ChoiceRANK4
2.0 C 0 False
25
25 125 4ChoiceRANK1
3.0 B 0
25 126 4ChoiceRANK2
3.0 C 0
25 127 4ChoiceRANK3
3.0 D 0
25 128 4ChoiceRANK4
3.0 A 0
26
26 130 4ChoiceRANK1
4.0 D 0 False
26 131 4ChoiceRANK2
4.0 B 0 False
26 132 4ChoiceRANK3
4.0 C 0 False
26 133 4ChoiceRANK4
4.0 A 0 False
27
27 135 4ChoiceRANK1
2.0 A 0 False
27 136 4ChoiceRANK2
2.0 C 0 False
27 137 4ChoiceRANK3
2.0 D 0 False
27 138 4ChoiceRANK4
2.0 B 0 False
28 140 4ChoiceRANK1
2.0 B 0
28 141 4ChoiceRANK2
2.0 A 0
28 142 4ChoiceRANK3
2.0 C 0
28 143 4ChoiceRANK4
error
29
29 145 4ChoiceRANK1
2.0 C 0
29 146 4ChoiceRANK2
2.0 B 0
29 147 4ChoiceRANK3
2.0 A 0
29 148 4ChoiceRANK4
error
30
30 150 4ChoiceRANK1
2.0 B 0
30 151 4ChoiceRANK2
2.0 C 0
30 152 4ChoiceRANK3
2.0 A 0
30 153 4ChoiceRANK4
```

error

```
31
31 155 4ChoiceRANK1
4.0 B 0 False
31 156 4ChoiceRANK2
4.0 A 0 False
31 157 4ChoiceRANK3
4.0 C 0 False
31 158 4ChoiceRANK4
4.0 D 0 False
32 160 4ChoiceRANK1
2.0 A 0 False
32 161 4ChoiceRANK2
2.0 B 0 False
32 162 4ChoiceRANK3
2.0 C 0 False
32 163 4ChoiceRANK4
2.0 D 0 False
33
33 165 4ChoiceRANK1
2.0 D 0 False
33 166 4ChoiceRANK2
2.0 C 0 False
33 167 4ChoiceRANK3
2.0 B 0 False
33 168 4ChoiceRANK4
2.0 A 0 False
34
34 170 4ChoiceRANK1
2.0 A 0 False
34 171 4ChoiceRANK2
2.0 B 0 False
34 172 4ChoiceRANK3
2.0 D 0 False
34 173 4ChoiceRANK4
2.0 C 0 False
35
35 175 4ChoiceRANK1
2.0 B 0
35 176 4ChoiceRANK2
2.0 A 0
35 177 4ChoiceRANK3
2.0 C 0
35 178 4ChoiceRANK4
error
36
36 180 4ChoiceRANK1
1.0 C 0 False
36 181 4ChoiceRANK2
1.0 D 0 False
36 182 4ChoiceRANK3
1.0 A 0 False
36 183 4ChoiceRANK4
1.0 B 0 False
37
37 185 4ChoiceRANK1
2.0 A 0
37 186 4ChoiceRANK2
2.0 B 0
37 187 4ChoiceRANK3
2.0 C 0
37 188 4ChoiceRANK4
error
38
38 190 4ChoiceRANK1
2.0 D 0 False
38 191 4ChoiceRANK2
2.0 C 0 False
38 192 4ChoiceRANK3
2.0 B 0 False
38 193 4ChoiceRANK4
2.0 A 0 False
39
39 195 4ChoiceRANK1
2.0 B 0 False
39 196 4ChoiceRANK2
2.0 A 0 False
39 197 4ChoiceRANK3
2.0 D 0 False
39 198 4ChoiceRANK4
2.0 C 0 False
40
40 200 4ChoiceRANK1
2.0 A 0 False
40 201 4ChoiceRANK2
2.0 B 0 False
40 202 4ChoiceRANK3
2.0 C 0 False
40 203 4ChoiceRANK4
2.0 D 0 False
41 205 4ChoiceRANK1
```

4.0 B 0 False

```
41 206 4ChoiceRANK2
4.0 A 0 False
41 207 4ChoiceRANK3
4.0 C 0 False
41 208 4ChoiceRANK4
4.0 D 0 False
42 210 4ChoiceRANK1
4.0 D 0 False
42 211 4ChoiceRANK2
4.0 B 0 False
42 212 4ChoiceRANK3
4.0 C 0 False
42 213 4ChoiceRANK4
4.0 A 0 False
43
43 215 4ChoiceRANK1
2.0 B 0 False
43 216 4ChoiceRANK2
2.0 A 0 False
43 217 4ChoiceRANK3
2.0 D 0 False
43 218 4ChoiceRANK4
2.0 C 0 False
44 220 4ChoiceRANK1
2.0 C 0 False
44 221 4ChoiceRANK2
2.0 A 0 False
44 222 4ChoiceRANK3
2.0 B 0 False
44 223 4ChoiceRANK4
2.0 D 0 False
45
45 225 4ChoiceRANK1
3.0 C 0
45 226 4ChoiceRANK2
3.0 D 0
45 227 4ChoiceRANK3
3.0 A 0
45 228 4ChoiceRANK4
3.0 B 0
46
46 230 4ChoiceRANK1
2.0 A 0 False
46 231 4ChoiceRANK2
2.0 D 0 False
46 232 4ChoiceRANK3
2.0 B 0 False
46 233 4ChoiceRANK4
2.0 C 0 False
47
47 235 4ChoiceRANK1
4.0 B 0 False
47 236 4ChoiceRANK2
4.0 D 0 False
47 237 4ChoiceRANK3
4.0 C 0 False
47 238 4ChoiceRANK4
4.0 A 0 False
48
48 240 4ChoiceRANK1
2.0 B 0 False
48 241 4ChoiceRANK2
2.0 A 0 False
48 242 4ChoiceRANK3
2.0 D 0 False
48 243 4ChoiceRANK4
2.0 C 0 False
49 245 4ChoiceRANK1
2.0 A 0 False
49 246 4ChoiceRANK2
2.0 C 0 False
49 247 4ChoiceRANK3
2.0 D 0 False
49 248 4ChoiceRANK4
2.0 B 0 False
50
50 250 4ChoiceRANK1
2.0 B 0 False
50 251 4ChoiceRANK2
2.0 A 0 False
50 252 4ChoiceRANK3
2.0 D 0 False
50 253 4ChoiceRANK4
2.0 C 0 False
51
51 255 4ChoiceRANK1
2.0 A 0 False
51 256 4ChoiceRANK2
2.0 C 0 False
```

51 257 4ChoiceRANK3

```
2.0 B 0 False
51 258 4ChoiceRANK4
2.0 D 0 False
52 260 4ChoiceRANK1
2.0 C 0
52 261 4ChoiceRANK2
2.0 B 0
52 262 4ChoiceRANK3
2.0 A 0
52 263 4ChoiceRANK4
error
53
53 265 4ChoiceRANK1
2.0 D 0 False
53 266 4ChoiceRANK2
2.0 B 0 False
53 267 4ChoiceRANK3
2.0 A 0 False
53 268 4ChoiceRANK4
2.0 C 0 False
54
54 270 4ChoiceRANK1
2.0 C 0 False
54 271 4ChoiceRANK2
2.0 A 0 False
54 272 4ChoiceRANK3
2.0 D 0 False
54 273 4ChoiceRANK4
2.0 B 0 False
55 275 4ChoiceRANK1
3.0 D 0
55 276 4ChoiceRANK2
3.0 B 0
55 277 4ChoiceRANK3
3.0 C 0
55 278 4ChoiceRANK4
3.0 A 0
56
56 280 4ChoiceRANK1
4.0 D 0 False
56 281 4ChoiceRANK2
4.0 B 0 False
56 282 4ChoiceRANK3
4.0 C 0 False
56 283 4ChoiceRANK4
4.0 A 0 False
57
57 285 4ChoiceRANK1
2.0 B 0 False
57 286 4ChoiceRANK2
2.0 A 0 False
57 287 4ChoiceRANK3
2.0 D 0 False
57 288 4ChoiceRANK4
2.0 C 0 False
58 290 4ChoiceRANK1
1.0 B 0 False
58 291 4ChoiceRANK2
1.0 D 0 False
58 292 4ChoiceRANK3
1.0 A 0 False
58 293 4ChoiceRANK4
1.0 C 0 False
59
59 295 4ChoiceRANK1
2.0 B 0 False
59 296 4ChoiceRANK2
2.0 D 0 False
59 297 4ChoiceRANK3
2.0 A 0 False
59 298 4ChoiceRANK4
2.0 C 0 False
60 300 4ChoiceRANK1
2.0 C 0 False
60 301 4ChoiceRANK2
2.0 B 0 False
60 302 4ChoiceRANK3
2.0 A 0 False
60 303 4ChoiceRANK4
2.0 D 0 False
61 305 4ChoiceRANK1
2.0 D 0 False
61 306 4ChoiceRANK2
2.0 B 0 False
61 307 4ChoiceRANK3
2.0 C 0 False
61 308 4ChoiceRANK4
```

2.0 A 0 False

```
62
62 310 4ChoiceRANK1
2.0 C 0
62 311 4ChoiceRANK2
2.0 B 0
62 312 4ChoiceRANK3
2.0 A 0
62 313 4ChoiceRANK4
63 315 4ChoiceRANK1
2.0 D 0 False
63 316 4ChoiceRANK2
2.0 C 0 False
63 317 4ChoiceRANK3
2.0 B 0 False
63 318 4ChoiceRANK4
2.0 A 0 False
64
64 320 4ChoiceRANK1
3.0 D 0
64 321 4ChoiceRANK2
3.0 B 0
64 322 4ChoiceRANK3
3.0 A 0
64 323 4ChoiceRANK4
3.0 C 0
65
65 325 4ChoiceRANK1
1.0 D 0 False
65 326 4ChoiceRANK2
1.0 A 0 False
65 327 4ChoiceRANK3
1.0 B 0 False
65 328 4ChoiceRANK4
1.0 C 0 False
66 330 4ChoiceRANK1
4.0 D 0 False
66 331 4ChoiceRANK2
4.0 C 0 False
66 332 4ChoiceRANK3
4.0 B 0 False
66 333 4ChoiceRANK4
4.0 A 0 False
67
67 335 4ChoiceRANK1
2.0 A 0 False
67 336 4ChoiceRANK2
2.0 C 0 False
67 337 4ChoiceRANK3
2.0 D 0 False
67 338 4ChoiceRANK4
2.0 B 0 False
68
68 340 4ChoiceRANK1
2.0 A 0
68 341 4ChoiceRANK2
2.0 B 0
68 342 4ChoiceRANK3
2.0 C 0
68 343 4ChoiceRANK4
error
69
69 345 4ChoiceRANK1
2.0 C 0
69 346 4ChoiceRANK2
2.0 B 0
69 347 4ChoiceRANK3
2.0 A 0
69 348 4ChoiceRANK4
error
70
70 350 4ChoiceRANK1
1.0 A 0 False
70 351 4ChoiceRANK2
1.0 C 0 False
70 352 4ChoiceRANK3
1.0 B 0 False
70 353 4ChoiceRANK4
1.0 D 0 False
71
71 355 4ChoiceRANK1
2.0 B 0 False
71 356 4ChoiceRANK2
2.0 D 0 False
71 357 4ChoiceRANK3
2.0 A 0 False
71 358 4ChoiceRANK4
2.0 C 0 False
72
72 360 4ChoiceRANK1
```

2.0 C 0 False

```
72 361 4ChoiceRANK2
2.0 A 0 False
72 362 4ChoiceRANK3
2.0 B 0 False
72 363 4ChoiceRANK4
2.0 D 0 False
73 365 4ChoiceRANK1
2.0 D 0 False
73 366 4ChoiceRANK2
2.0 C 0 False
73 367 4ChoiceRANK3
2.0 B 0 False
73 368 4ChoiceRANK4
2.0 A 0 False
74
74 370 4ChoiceRANK1
2.0 B 0 False
74 371 4ChoiceRANK2
2.0 D 0 False
74 372 4ChoiceRANK3
2.0 A 0 False
74 373 4ChoiceRANK4
2.0 C 0 False
75 375 4ChoiceRANK1
3.0 B 0
75 376 4ChoiceRANK2
3.0 D 0
75 377 4ChoiceRANK3
3.0 C 0
75 378 4ChoiceRANK4
3.0 A 0
76
76 380 4ChoiceRANK1
1.0 B 0
76 381 4ChoiceRANK2
1.0 A 0
76 382 4ChoiceRANK3
1.0 C 0
76 383 4ChoiceRANK4
error
77
77 385 4ChoiceRANK1
2.0 B 0 False
77 386 4ChoiceRANK2
2.0 A 0 False
77 387 4ChoiceRANK3
2.0 C 0 False
77 388 4ChoiceRANK4
2.0 D 0 False
78
78 390 4ChoiceRANK1
3.0 C 0
78 391 4ChoiceRANK2
3.0 B 0
78 392 4ChoiceRANK3
3.0 A 0
78 393 4ChoiceRANK4
3.0 D 0
79
79 395 4ChoiceRANK1
2.0 A 0
79 396 4ChoiceRANK2
2.0 B 0
79 397 4ChoiceRANK3
2.0 C 0
79 398 4ChoiceRANK4
error
80 400 4ChoiceRANK1
2.0 A 0 False
80 401 4ChoiceRANK2
2.0 B 0 False
80 402 4ChoiceRANK3
2.0 C 0 False
80 403 4ChoiceRANK4
2.0 D 0 False
81
81 405 4ChoiceRANK1
2.0 B 0 False
81 406 4ChoiceRANK2
2.0 A 0 False
81 407 4ChoiceRANK3
2.0 D 0 False
81 408 4ChoiceRANK4
2.0 C 0 False
82
82 410 4ChoiceRANK1
3.0 C 0
82 411 4ChoiceRANK2
3.0 A 0
```

82 412 4ChoiceRANK3

```
3.0 B 0
82 413 4ChoiceRANK4
3.0 D 0
83 415 4ChoiceRANK1
2.0 A 0 False
83 416 4ChoiceRANK2
2.0 B 0 False
83 417 4ChoiceRANK3
2.0 D 0 False
83 418 4ChoiceRANK4
2.0 C 0 False
84
84 420 4ChoiceRANK1
4.0 C 0 False
84 421 4ChoiceRANK2
4.0 A 0 False
84 422 4ChoiceRANK3
4.0 B 0 False
84 423 4ChoiceRANK4
4.0 D 0 False
85
85 425 4ChoiceRANK1
1.0 C 0 False
85 426 4ChoiceRANK2
1.0 A 0 False
85 427 4ChoiceRANK3
1.0 D 0 False
85 428 4ChoiceRANK4
1.0 B 0 False
86 430 4ChoiceRANK1
2.0 C 0
86 431 4ChoiceRANK2
2.0 A 0
86 432 4ChoiceRANK3
2.0 B 0
86 433 4ChoiceRANK4
error
87
87 435 4ChoiceRANK1
2.0 A 0
87 436 4ChoiceRANK2
2.0 C 0
87 437 4ChoiceRANK3
2.0 B 0
87 438 4ChoiceRANK4
error
88
88 440 4ChoiceRANK1
2.0 B 0 False
88 441 4ChoiceRANK2
2.0 A 0 False
88 442 4ChoiceRANK3
2.0 D 0 False
88 443 4ChoiceRANK4
2.0 C 0 False
89 445 4ChoiceRANK1
2.0 D 0 False
89 446 4ChoiceRANK2
2.0 C 0 False
89 447 4ChoiceRANK3
2.0 A 0 False
89 448 4ChoiceRANK4
2.0 B 0 False
90
90 450 4ChoiceRANK1
2.0 B 0 False
90 451 4ChoiceRANK2
2.0 D 0 False
90 452 4ChoiceRANK3
2.0 A 0 False
90 453 4ChoiceRANK4
2.0 C 0 False
91
91 455 4ChoiceRANK1
2.0 A 0 False
91 456 4ChoiceRANK2
2.0 B 0 False
91 457 4ChoiceRANK3
2.0 C 0 False
91 458 4ChoiceRANK4
2.0 D 0 False
92 460 4ChoiceRANK1
2.0 A 0
92 461 4ChoiceRANK2
2.0 B 0
92 462 4ChoiceRANK3
2.0 C 0
92 463 4ChoiceRANK4
```

error

```
93
93 465 4ChoiceRANK1
2.0 B 0 False
93 466 4ChoiceRANK2
2.0 A 0 False
93 467 4ChoiceRANK3
2.0 D 0 False
93 468 4ChoiceRANK4
2.0 C 0 False
94 470 4ChoiceRANK1
3.0 B 0
94 471 4ChoiceRANK2
3.0 A 0
94 472 4ChoiceRANK3
3.0 C 0
94 473 4ChoiceRANK4
3.0 D 0
95
95 475 4ChoiceRANK1
2.0 C 0 False
95 476 4ChoiceRANK2
2.0 A 0 False
95 477 4ChoiceRANK3
2.0 B 0 False
95 478 4ChoiceRANK4
2.0 D 0 False
96
96 480 4ChoiceRANK1
2.0 B 0 False
96 481 4ChoiceRANK2
2.0 D 0 False
96 482 4ChoiceRANK3
2.0 A 0 False
96 483 4ChoiceRANK4
2.0 C 0 False
97
97 485 4ChoiceRANK1
1.0 A 0
97 486 4ChoiceRANK2
1.0 B 0
97 487 4ChoiceRANK3
1.0 C 0
97 488 4ChoiceRANK4
error
98
98 490 4ChoiceRANK1
2.0 A 0 False
98 491 4ChoiceRANK2
2.0 B 0 False
98 492 4ChoiceRANK3
2.0 C 0 False
98 493 4ChoiceRANK4
2.0 D 0 False
99
99 495 4ChoiceRANK1
1.0 B 0 False
99 496 4ChoiceRANK2
1.0 A 0 False
99 497 4ChoiceRANK3
1.0 D 0 False
99 498 4ChoiceRANK4
1.0 C 0 False
100
100 500 4ChoiceRANK1
2.0 D 0 False
100 501 4ChoiceRANK2
2.0 B 0 False
100 502 4ChoiceRANK3
2.0 C 0 False
100 503 4ChoiceRANK4
2.0 A 0 False
101
101 505 4ChoiceRANK1
4.0 D 0 False
101 506 4ChoiceRANK2
4.0 B 0 False
101 507 4ChoiceRANK3
4.0 C 0 False
101 508 4ChoiceRANK4
4.0 A 0 False
102
102 510 4ChoiceRANK1
4.0 D 0 False
102 511 4ChoiceRANK2
4.0 C 0 False
102 512 4ChoiceRANK3
4.0 B 0 False
102 513 4ChoiceRANK4
4.0 A 0 False
103
103 515 4ChoiceRANK1
1.0 A 0 False
```

```
103 516 4ChoiceRANK2
1.0 B 0 False
103 517 4ChoiceRANK3
1.0 C 0 False
103 518 4ChoiceRANK4
1.0 D 0 False
104 520 4ChoiceRANK1
2.0 C 0 False
104 521 4ChoiceRANK2
2.0 A 0 False
104 522 4ChoiceRANK3
2.0 D 0 False
104 523 4ChoiceRANK4
2.0 B 0 False
105
105 525 4ChoiceRANK1
2.0 B 0 False
105 526 4ChoiceRANK2
2.0 D 0 False
105 527 4ChoiceRANK3
2.0 C 0 False
105 528 4ChoiceRANK4
2.0 A 0 False
106
106 530 4ChoiceRANK1
2.0 B 0
106 531 4ChoiceRANK2
2.0 A 0
106 532 4ChoiceRANK3
2.0 C 0
106 533 4ChoiceRANK4
error
107
107 535 4ChoiceRANK1
2.0 B 0 False
107 536 4ChoiceRANK2
2.0 A 0 False
107 537 4ChoiceRANK3
2.0 D 0 False
107 538 4ChoiceRANK4
2.0 C 0 False
108
108 540 4ChoiceRANK1
2.0 A 0
108 541 4ChoiceRANK2
2.0 B 0
108 542 4ChoiceRANK3
2.0 C 0
108 543 4ChoiceRANK4
error
109
109 545 4ChoiceRANK1
2.0 C 0 False
109 546 4ChoiceRANK2
2.0 D 0 False
109 547 4ChoiceRANK3
2.0 A 0 False
109 548 4ChoiceRANK4
2.0 B 0 False
110
110 550 4ChoiceRANK1
2.0 C 0
110 551 4ChoiceRANK2
2.0 B 0
110 552 4ChoiceRANK3
2.0 A 0
110 553 4ChoiceRANK4
error
111 555 4ChoiceRANK1
2.0 A 0 False
111 556 4ChoiceRANK2
2.0 C 0 False
111 557 4ChoiceRANK3
2.0 B 0 False
111 558 4ChoiceRANK4
2.0 D 0 False
112
112 560 4ChoiceRANK1
3.0 A 0
112 561 4ChoiceRANK2
3.0 B 0
112 562 4ChoiceRANK3
3.0 C 0
112 563 4ChoiceRANK4
3.0 D 0
113
113 565 4ChoiceRANK1
2.0 A 0 False
113 566 4ChoiceRANK2
2.0 B 0 False
113 567 4ChoiceRANK3
```

```
2.0 D 0 False
113 568 4ChoiceRANK4
2.0 C 0 False
114
114 570 4ChoiceRANK1
2.0 A 0 False
114 571 4ChoiceRANK2
2.0 B 0 False
114 572 4ChoiceRANK3
2.0 C 0 False
114 573 4ChoiceRANK4
2.0 D 0 False
115
115 575 4ChoiceRANK1
2.0 C 0
115 576 4ChoiceRANK2
2.0 B 0
115 577 4ChoiceRANK3
2.0 A 0
115 578 4ChoiceRANK4
error
116
116 580 4ChoiceRANK1
2.0 B 0 False
116 581 4ChoiceRANK2
2.0 C 0 False
116 582 4ChoiceRANK3
2.0 A 0 False
116 583 4ChoiceRANK4
2.0 D 0 False
117 585 4ChoiceRANK1
3.0 B 0 False
117 586 4ChoiceRANK2
3.0 D 0 False
117 587 4ChoiceRANK3
3.0 A 0 False
117 588 4ChoiceRANK4
3.0 C 0 False
118
118 590 4ChoiceRANK1
3.0 D 0 False
118 591 4ChoiceRANK2
3.0 B 0 False
118 592 4ChoiceRANK3
3.0 C 0 False
118 593 4ChoiceRANK4
3.0 A 0 False
119
119 595 4ChoiceRANK1
2.0 A 0 False
119 596 4ChoiceRANK2
2.0 C 0 False
119 597 4ChoiceRANK3
2.0 B 0 False
119 598 4ChoiceRANK4
2.0 D 0 False
120
120 600 4ChoiceRANK1
3.0 B 0 False
120 601 4ChoiceRANK2
3.0 A 0 False
120 602 4ChoiceRANK3
3.0 D 0 False
120 603 4ChoiceRANK4
3.0 C 0 False
121
121 605 4ChoiceRANK1
2.0 C 0
121 606 4ChoiceRANK2
2.0 B 0
121 607 4ChoiceRANK3
2.0 A 0
121 608 4ChoiceRANK4
error
122
122 610 4ChoiceRANK1
3.0 C 0
122 611 4ChoiceRANK2
3.0 D 0
122 612 4ChoiceRANK3
3.0 A 0
122 613 4ChoiceRANK4
3.0 B 0
123
123 615 4ChoiceRANK1
2.0 B 0 False
123 616 4ChoiceRANK2
2.0 A 0 False
123 617 4ChoiceRANK3
2.0 D 0 False
```

123 618 4ChoiceRANK4 2.0 C 0 False

```
124
124 620 4ChoiceRANK1
2.0 B 0
124 621 4ChoiceRANK2
2.0 C 0
124 622 4ChoiceRANK3
2.0 A 0
124 623 4ChoiceRANK4
error
125
125 625 4ChoiceRANK1
3.0 A 0 False
125 626 4ChoiceRANK2
3.0 B 0 False
125 627 4ChoiceRANK3
3.0 D 0 False
125 628 4ChoiceRANK4
3.0 C 0 False
126
126 630 4ChoiceRANK1
3.0 A 0 False
126 631 4ChoiceRANK2
3.0 D 0 False
126 632 4ChoiceRANK3
3.0 C 0 False
126 633 4ChoiceRANK4
3.0 B 0 False
127
127 635 4ChoiceRANK1
3.0 B 0 False
127 636 4ChoiceRANK2
3.0 A 0 False
127 637 4ChoiceRANK3
3.0 D 0 False
127 638 4ChoiceRANK4
3.0 C 0 False
128
128 640 4ChoiceRANK1
2.0 A 0 False
128 641 4ChoiceRANK2
2.0 D 0 False
128 642 4ChoiceRANK3
2.0 C 0 False
128 643 4ChoiceRANK4
2.0 B 0 False
129
129 645 4ChoiceRANK1
2.0 C 0
129 646 4ChoiceRANK2
2.0 B 0
129 647 4ChoiceRANK3
2.0 A 0
129 648 4ChoiceRANK4
error
130
130 650 4ChoiceRANK1
2.0 B 0 False
130 651 4ChoiceRANK2
2.0 D 0 False
130 652 4ChoiceRANK3
2.0 A 0 False
130 653 4ChoiceRANK4
2.0 C 0 False
131
131 655 4ChoiceRANK1
3.0 C 0 False
131 656 4ChoiceRANK2
3.0 D 0 False
131 657 4ChoiceRANK3
3.0 B 0 False
131 658 4ChoiceRANK4
3.0 A 0 False
132
132 660 4ChoiceRANK1
3.0 A 0 False
132 661 4ChoiceRANK2
3.0 B 0 False
132 662 4ChoiceRANK3
3.0 C 0 False
132 663 4ChoiceRANK4
3.0 D 0 False
133
133 665 4ChoiceRANK1
2.0 D 0 False
133 666 4ChoiceRANK2
2.0 A 0 False
133 667 4ChoiceRANK3
2.0 C 0 False
133 668 4ChoiceRANK4
2.0 B 0 False
134
134 670 4ChoiceRANK1
```

1.0 A 0

```
134 671 4ChoiceRANK2
1.0 C 0
134 672 4ChoiceRANK3
1.0 B 0
134 673 4ChoiceRANK4
error
135
135 675 4ChoiceRANK1
2.0 D 0 False
135 676 4ChoiceRANK2
2.0 A 0 False
135 677 4ChoiceRANK3
2.0 B 0 False
135 678 4ChoiceRANK4
2.0 C 0 False
136
136 680 4ChoiceRANK1
4.0 A 0 False
136 681 4ChoiceRANK2
4.0 B 0 False
136 682 4ChoiceRANK3
4.0 C 0 False
136 683 4ChoiceRANK4
4.0 D 0 False
137
137 685 4ChoiceRANK1
3.0 A 0 False
137 686 4ChoiceRANK2
3.0 C 0 False
137 687 4ChoiceRANK3
3.0 D 0 False
137 688 4ChoiceRANK4
3.0 B 0 False
138
138 690 4ChoiceRANK1
2.0 B 0 False
138 691 4ChoiceRANK2
2.0 D 0 False
138 692 4ChoiceRANK3
2.0 A 0 False
138 693 4ChoiceRANK4
2.0 C 0 False
139
139 695 4ChoiceRANK1
3.0 B 0 False
139 696 4ChoiceRANK2
3.0 D 0 False
139 697 4ChoiceRANK3
3.0 A 0 False
139 698 4ChoiceRANK4
3.0 C 0 False
140
140 700 4ChoiceRANK1
3.0 D 0
140 701 4ChoiceRANK2
3.0 C 0
140 702 4ChoiceRANK3
3.0 B 0
140 703 4ChoiceRANK4
3.0 A 0
141
141 705 4ChoiceRANK1
2.0 A 0 False
141 706 4ChoiceRANK2
2.0 B 0 False
141 707 4ChoiceRANK3
2.0 C 0 False
141 708 4ChoiceRANK4
2.0 D 0 False
142 710 4ChoiceRANK1
2.0 D 0 False
142 711 4ChoiceRANK2
2.0 B 0 False
142 712 4ChoiceRANK3
2.0 A 0 False
142 713 4ChoiceRANK4
2.0 C 0 False
143
143 715 4ChoiceRANK1
3.0 A 0 False
143 716 4ChoiceRANK2
3.0 B 0 False
143 717 4ChoiceRANK3
3.0 C 0 False
143 718 4ChoiceRANK4
3.0 D 0 False
144
144 720 4ChoiceRANK1
3.0 D 0 False
144 721 4ChoiceRANK2
3.0 B 0 False
144 722 4ChoiceRANK3
```

```
3.0 A 0 False
144 723 4ChoiceRANK4
3.0 C 0 False
145 725 4ChoiceRANK1
3.0 A 0 False
145 726 4ChoiceRANK2
3.0 B 0 False
145 727 4ChoiceRANK3
3.0 D 0 False
145 728 4ChoiceRANK4
3.0 C 0 False
146
146 730 4ChoiceRANK1
3.0 D 0
146 731 4ChoiceRANK2
3.0 B 0
146 732 4ChoiceRANK3
3.0 C 0
146 733 4ChoiceRANK4
3.0 A 0
147
147 735 4ChoiceRANK1
3.0 B 0 False
147 736 4ChoiceRANK2
3.0 D 0 False
147 737 4ChoiceRANK3
3.0 A 0 False
147 738 4ChoiceRANK4
3.0 C 0 False
148
148 740 4ChoiceRANK1
2.0 B 0 False
148 741 4ChoiceRANK2
2.0 A 0 False
148 742 4ChoiceRANK3
2.0 D 0 False
148 743 4ChoiceRANK4
2.0 C 0 False
149
149 745 4ChoiceRANK1
4.0 D 0 False
149 746 4ChoiceRANK2
4.0 C 0 False
149 747 4ChoiceRANK3
4.0 B 0 False
149 748 4ChoiceRANK4
4.0 A 0 False
150
150 750 4ChoiceRANK1
4.0 D 0 False
150 751 4ChoiceRANK2
4.0 C 0 False
150 752 4ChoiceRANK3
4.0 B 0 False
150 753 4ChoiceRANK4
4.0 A 0 False
151
151 755 4ChoiceRANK1
2.0 C 0
151 756 4ChoiceRANK2
2.0 B 0
151 757 4ChoiceRANK3
2.0 A 0
151 758 4ChoiceRANK4
error
152
152 760 4ChoiceRANK1
2.0 B 0 False
152 761 4ChoiceRANK2
2.0 D 0 False
152 762 4ChoiceRANK3
2.0 A 0 False
152 763 4ChoiceRANK4
2.0 C 0 False
153
153 765 4ChoiceRANK1
1.0 A 0 False
153 766 4ChoiceRANK2
1.0 D 0 False
153 767 4ChoiceRANK3
1.0 B 0 False
153 768 4ChoiceRANK4
1.0 C 0 False
154
154 770 4ChoiceRANK1
2.0 B 0 False
154 771 4ChoiceRANK2
2.0 A 0 False
154 772 4ChoiceRANK3
2.0 D 0 False
```

2.0 D 0 False 154 773 4ChoiceRANK4 2.0 C 0 False

```
155
155 775 4ChoiceRANK1
2.0 B 0 False
155 776 4ChoiceRANK2
2.0 A 0 False
155 777 4ChoiceRANK3
2.0 D 0 False
155 778 4ChoiceRANK4
2.0 C 0 False
156
156 780 4ChoiceRANK1
3.0 C 0
156 781 4ChoiceRANK2
3.0 D 0
156 782 4ChoiceRANK3
3.0 A 0
156 783 4ChoiceRANK4
3.0 B 0
157
157 785 4ChoiceRANK1
3.0 B 0 False
157 786 4ChoiceRANK2
3.0 D 0 False
157 787 4ChoiceRANK3
3.0 A 0 False
157 788 4ChoiceRANK4
3.0 C 0 False
158
158 790 4ChoiceRANK1
3.0 A 0 False
158 791 4ChoiceRANK2
3.0 B 0 False
158 792 4ChoiceRANK3
3.0 C 0 False
158 793 4ChoiceRANK4
3.0 D 0 False
159
159 795 4ChoiceRANK1
3.0 A 0 False
159 796 4ChoiceRANK2
3.0 B 0 False
159 797 4ChoiceRANK3
3.0 D 0 False
159 798 4ChoiceRANK4
3.0 C 0 False
160
160 800 4ChoiceRANK1
2.0 A 0 False
160 801 4ChoiceRANK2
2.0 D 0 False
160 802 4ChoiceRANK3
2.0 B 0 False
160 803 4ChoiceRANK4
2.0 C 0 False
161
161 805 4ChoiceRANK1
2.0 B 0 False
161 806 4ChoiceRANK2
2.0 A 0 False
161 807 4ChoiceRANK3
2.0 C 0 False
161 808 4ChoiceRANK4
2.0 D 0 False
162
162 810 4ChoiceRANK1
1.0 B 0
162 811 4ChoiceRANK2
1.0 A 0
162 812 4ChoiceRANK3
1.0 C 0
162 813 4ChoiceRANK4
error
163
163 815 4ChoiceRANK1
3.0 A 0 False
163 816 4ChoiceRANK2
3.0 B 0 False
163 817 4ChoiceRANK3
3.0 D 0 False
163 818 4ChoiceRANK4
3.0 C 0 False
164
164 820 4ChoiceRANK1
2.0 A 0 False
164 821 4ChoiceRANK2
2.0 C 0 False
164 822 4ChoiceRANK3
2.0 B 0 False
164 823 4ChoiceRANK4
2.0 D 0 False
165
165 825 4ChoiceRANK1
```

3.0 A 0

```
165 826 4ChoiceRANK2
3.0 D 0
165 827 4ChoiceRANK3
3.0 B 0
165 828 4ChoiceRANK4
3.0 C 0
166
166 830 4ChoiceRANK1
2.0 B 0 False
166 831 4ChoiceRANK2
2.0 A 0 False
166 832 4ChoiceRANK3
2.0 D 0 False
166 833 4ChoiceRANK4
2.0 C 0 False
167
167 835 4ChoiceRANK1
2.0 A 0 False
167 836 4ChoiceRANK2
2.0 B 0 False
167 837 4ChoiceRANK3
2.0 C 0 False
167 838 4ChoiceRANK4
2.0 D 0 False
168
168 840 4ChoiceRANK1
1.0 B 0 False
168 841 4ChoiceRANK2
1.0 A 0 False
168 842 4ChoiceRANK3
1.0 C 0 False
168 843 4ChoiceRANK4
1.0 D 0 False
169
169 845 4ChoiceRANK1
3.0 A 0 False
169 846 4ChoiceRANK2
3.0 B 0 False
169 847 4ChoiceRANK3
3.0 D 0 False
169 848 4ChoiceRANK4
3.0 C 0 False
170
170 850 4ChoiceRANK1
2.0 B 0 False
170 851 4ChoiceRANK2
2.0 A 0 False
170 852 4ChoiceRANK3
2.0 C 0 False
170 853 4ChoiceRANK4
2.0 D 0 False
171
171 855 4ChoiceRANK1
3.0 A 0 False
171 856 4ChoiceRANK2
3.0 B 0 False
171 857 4ChoiceRANK3
3.0 C 0 False
171 858 4ChoiceRANK4
3.0 D 0 False
172
172 860 4ChoiceRANK1
2.0 C 0
172 861 4ChoiceRANK2
2.0 A 0
172 862 4ChoiceRANK3
2.0 B 0
172 863 4ChoiceRANK4
error
173 865 4ChoiceRANK1
4.0 A 0 False
173 866 4ChoiceRANK2
4.0 B 0 False
173 867 4ChoiceRANK3
4.0 C 0 False
173 868 4ChoiceRANK4
4.0 D 0 False
174
174 870 4ChoiceRANK1
2.0 A 0 False
174 871 4ChoiceRANK2
2.0 B 0 False
174 872 4ChoiceRANK3
2.0 D 0 False
174 873 4ChoiceRANK4
2.0 C 0 False
175
175 875 4ChoiceRANK1
3.0 A 0 False
175 876 4ChoiceRANK2
3.0 B 0 False
175 877 4ChoiceRANK3
```

```
3.0 D 0 False
175 878 4ChoiceRANK4
3.0 C 0 False
176
176 880 4ChoiceRANK1
3.0 A 0 False
176 881 4ChoiceRANK2
3.0 C 0 False
176 882 4ChoiceRANK3
3.0 D 0 False
176 883 4ChoiceRANK4
3.0 B 0 False
177
177 885 4ChoiceRANK1
2.0 D 0 False
177 886 4ChoiceRANK2
2.0 C 0 False
177 887 4ChoiceRANK3
2.0 A 0 False
177 888 4ChoiceRANK4
2.0 B 0 False
178
178 890 4ChoiceRANK1
2.0 C 0
178 891 4ChoiceRANK2
2.0 B 0
178 892 4ChoiceRANK3
2.0 A 0
178 893 4ChoiceRANK4
error
179
179 895 4ChoiceRANK1
1.0 B 0
179 896 4ChoiceRANK2
1.0 C 0
179 897 4ChoiceRANK3
1.0 A 0
179 898 4ChoiceRANK4
error
180
180 900 4ChoiceRANK1
2.0 A 0
180 901 4ChoiceRANK2
2.0 B 0
180 902 4ChoiceRANK3
2.0 C 0
180 903 4ChoiceRANK4
error
181
181 905 4ChoiceRANK1
2.0 A 0 False
181 906 4ChoiceRANK2
2.0 B 0 False
181 907 4ChoiceRANK3
2.0 C 0 False
181 908 4ChoiceRANK4
2.0 D 0 False
182
182 910 4ChoiceRANK1
2.0 A 0 False
182 911 4ChoiceRANK2
2.0 B 0 False
182 912 4ChoiceRANK3
2.0 D 0 False
182 913 4ChoiceRANK4
2.0 C 0 False
183
183 915 4ChoiceRANK1
2.0 A 0 False
183 916 4ChoiceRANK2
2.0 D 0 False
183 917 4ChoiceRANK3
2.0 C 0 False
183 918 4ChoiceRANK4
2.0 B 0 False
184
184 920 4ChoiceRANK1
3.0 B 0
184 921 4ChoiceRANK2
3.0 A 0
184 922 4ChoiceRANK3
3.0 D 0
184 923 4ChoiceRANK4
3.0 C 0
185
185 925 4ChoiceRANK1
2.0 B 0 False
185 926 4ChoiceRANK2
2.0 A 0 False
185 927 4ChoiceRANK3
2.0 C 0 False
```

2.0 C 0 False 185 928 4ChoiceRANK4 2.0 D 0 False

```
186
186 930 4ChoiceRANK1
2.0 B 0 False
186 931 4ChoiceRANK2
2.0 A 0 False
186 932 4ChoiceRANK3
2.0 D 0 False
186 933 4ChoiceRANK4
2.0 C 0 False
187 935 4ChoiceRANK1
2.0 C 0
187 936 4ChoiceRANK2
2.0 B 0
187 937 4ChoiceRANK3
2.0 A 0
187 938 4ChoiceRANK4
error
188
188 940 4ChoiceRANK1
3.0 B 0 False
188 941 4ChoiceRANK2
3.0 A 0 False
188 942 4ChoiceRANK3
3.0 D 0 False
188 943 4ChoiceRANK4
3.0 C 0 False
189
189 945 4ChoiceRANK1
2.0 D 0 False
189 946 4ChoiceRANK2
2.0 B 0 False
189 947 4ChoiceRANK3
2.0 A 0 False
189 948 4ChoiceRANK4
2.0 C 0 False
190
190 950 4ChoiceRANK1
2.0 A 0 False
190 951 4ChoiceRANK2
2.0 B 0 False
190 952 4ChoiceRANK3
2.0 C 0 False
190 953 4ChoiceRANK4
2.0 D 0 False
191
191 955 4ChoiceRANK1
3.0 C 0 False
191 956 4ChoiceRANK2
3.0 A 0 False
191 957 4ChoiceRANK3
3.0 B 0 False
191 958 4ChoiceRANK4
3.0 D 0 False
192
192 960 4ChoiceRANK1
2.0 A 0 False
192 961 4ChoiceRANK2
2.0 B 0 False
192 962 4ChoiceRANK3
2.0 C 0 False
192 963 4ChoiceRANK4
2.0 D 0 False
193
193 965 4ChoiceRANK1
2.0 C 0
193 966 4ChoiceRANK2
2.0 B 0
193 967 4ChoiceRANK3
2.0 A 0
193 968 4ChoiceRANK4
error
194
194 970 4ChoiceRANK1
3.0 A 0 False
194 971 4ChoiceRANK2
3.0 C 0 False
194 972 4ChoiceRANK3
3.0 B 0 False
194 973 4ChoiceRANK4
3.0 D 0 False
195
195 975 4ChoiceRANK1
3.0 B 0 False
195 976 4ChoiceRANK2
3.0 D 0 False
195 977 4ChoiceRANK3
3.0 A 0 False
195 978 4ChoiceRANK4
3.0 C 0 False
196
196 980 4ChoiceRANK1
```

1.0 B 0 False

```
196 981 4ChoiceRANK2
1.0 A 0 False
196 982 4ChoiceRANK3
1.0 D 0 False
196 983 4ChoiceRANK4
1.0 C 0 False
197
197 985 4ChoiceRANK1
2.0 D 0 False
197 986 4ChoiceRANK2
2.0 B 0 False
197 987 4ChoiceRANK3
2.0 A 0 False
197 988 4ChoiceRANK4
2.0 C 0 False
198
198 990 4ChoiceRANK1
4.0 D 0 False
198 991 4ChoiceRANK2
4.0 C 0 False
198 992 4ChoiceRANK3
4.0 B 0 False
198 993 4ChoiceRANK4
4.0 A 0 False
199 995 4ChoiceRANK1
3.0 A 0 False
199 996 4ChoiceRANK2
3.0 B 0 False
199 997 4ChoiceRANK3
3.0 C 0 False
199 998 4ChoiceRANK4
3.0 D 0 False
200
200 1000 4ChoiceRANK1
3.0 B 0 False
200 1001 4ChoiceRANK2
3.0 A 0 False
200 1002 4ChoiceRANK3
3.0 D 0 False
200 1003 4ChoiceRANK4
3.0 C 0 False
201
201 1005 4ChoiceRANK1
2.0 C 0 False
201 1006 4ChoiceRANK2
2.0 D 0 False
201 1007 4ChoiceRANK3
2.0 B 0 False
201 1008 4ChoiceRANK4
2.0 A 0 False
202
202 1010 4ChoiceRANK1
3.0 A 0 False
202 1011 4ChoiceRANK2
3.0 B 0 False
202 1012 4ChoiceRANK3
3.0 C 0 False
202 1013 4ChoiceRANK4
3.0 D 0 False
203
203 1015 4ChoiceRANK1
2.0 A 0
203 1016 4ChoiceRANK2
2.0 B 0
203 1017 4ChoiceRANK3
2.0 C 0
203 1018 4ChoiceRANK4
error
204 1020 4ChoiceRANK1
1.0 D 0 False
204 1021 4ChoiceRANK2
1.0 B 0 False
204 1022 4ChoiceRANK3
1.0 C 0 False
204 1023 4ChoiceRANK4
1.0 A 0 False
205
205 1025 4ChoiceRANK1
2.0 B 0 False
205 1026 4ChoiceRANK2
2.0 D 0 False
205 1027 4ChoiceRANK3
2.0 C 0 False
205 1028 4ChoiceRANK4
2.0 A 0 False
206
206 1030 4ChoiceRANK1
2.0 A 0 False
206 1031 4ChoiceRANK2
2.0 C 0 False
```

206 1032 4ChoiceRANK3

```
2.0 B 0 False
206 1033 4ChoiceRANK4
2.0 D 0 False
207 1035 4ChoiceRANK1
3.0 C 0 False
207 1036 4ChoiceRANK2
3.0 D 0 False
207 1037 4ChoiceRANK3
3.0 B 0 False
207 1038 4ChoiceRANK4
3.0 A 0 False
208
208 1040 4ChoiceRANK1
4.0 A 0 False
208 1041 4ChoiceRANK2
4.0 B 0 False
208 1042 4ChoiceRANK3
4.0 C 0 False
208 1043 4ChoiceRANK4
4.0 D 0 False
209
209 1045 4ChoiceRANK1
3.0 B 0 False
209 1046 4ChoiceRANK2
3.0 D 0 False
209 1047 4ChoiceRANK3
3.0 C 0 False
209 1048 4ChoiceRANK4
3.0 A 0 False
210
210 1050 4ChoiceRANK1
2.0 A 0 False
210 1051 4ChoiceRANK2
2.0 C 0 False
210 1052 4ChoiceRANK3
2.0 B 0 False
210 1053 4ChoiceRANK4
2.0 D 0 False
211
211 1055 4ChoiceRANK1
3.0 B 0 False
211 1056 4ChoiceRANK2
3.0 C 0 False
211 1057 4ChoiceRANK3
3.0 D 0 False
211 1058 4ChoiceRANK4
3.0 A 0 False
212
212 1060 4ChoiceRANK1
2.0 A 0
212 1061 4ChoiceRANK2
2.0 B 0
212 1062 4ChoiceRANK3
2.0 C 0
212 1063 4ChoiceRANK4
error
213
213 1065 4ChoiceRANK1
1.0 B 0 False
213 1066 4ChoiceRANK2
1.0 D 0 False
213 1067 4ChoiceRANK3
1.0 C 0 False
213 1068 4ChoiceRANK4
1.0 A 0 False
214
214 1070 4ChoiceRANK1
3.0 D 0
214 1071 4ChoiceRANK2
3.0 B 0
214 1072 4ChoiceRANK3
3.0 C 0
214 1073 4ChoiceRANK4
3.0 A 0
215
215 1075 4ChoiceRANK1
3.0 A 0 False
215 1076 4ChoiceRANK2
3.0 B 0 False
215 1077 4ChoiceRANK3
3.0 D 0 False
215 1078 4ChoiceRANK4
3.0 C 0 False
216
216 1080 4ChoiceRANK1
3.0 D 0 False
216 1081 4ChoiceRANK2
3.0 B 0 False
216 1082 4ChoiceRANK3
3.0 C 0 False
216 1083 4ChoiceRANK4
```

3.0 A 0 False

217 217 1085 4ChoiceRANK1 3.0 B 0 False 217 1086 4ChoiceRANK2 3.0 C 0 False 217 1087 4ChoiceRANK3 3.0 A 0 False 217 1088 4ChoiceRANK4 3.0 D 0 False 218 218 1090 4ChoiceRANK1 2.0 D 0 False 218 1091 4ChoiceRANK2 2.0 B 0 False 218 1092 4ChoiceRANK3 2.0 A 0 False 218 1093 4ChoiceRANK4 2.0 C 0 False 219 219 1095 4ChoiceRANK1 2.0 B 0 False 219 1096 4ChoiceRANK2 2.0 D 0 False 219 1097 4ChoiceRANK3 2.0 A 0 False 219 1098 4ChoiceRANK4 2.0 C 0 False 220 220 1100 4ChoiceRANK1 1.0 B 0 False 220 1101 4ChoiceRANK2 1.0 D 0 False 220 1102 4ChoiceRANK3 1.0 A 0 False 220 1103 4ChoiceRANK4 1.0 C 0 False 221 221 1105 4ChoiceRANK1 1.0 A 0 False 221 1106 4ChoiceRANK2 1.0 B 0 False 221 1107 4ChoiceRANK3 1.0 D 0 False 221 1108 4ChoiceRANK4 1.0 C 0 False 222 222 1110 4ChoiceRANK1 3.0 B 0 False 222 1111 4ChoiceRANK2 3.0 A 0 False 222 1112 4ChoiceRANK3 3.0 D 0 False 222 1113 4ChoiceRANK4 3.0 C 0 False 223 223 1115 4ChoiceRANK1 2.0 B 0 False 223 1116 4ChoiceRANK2 2.0 D 0 False 223 1117 4ChoiceRANK3 2.0 C 0 False 223 1118 4ChoiceRANK4 2.0 A 0 False 224 224 1120 4ChoiceRANK1 2.0 B 0 False 224 1121 4ChoiceRANK2 2.0 A 0 False 224 1122 4ChoiceRANK3 2.0 D 0 False 224 1123 4ChoiceRANK4 2.0 C 0 False 225 225 1125 4ChoiceRANK1 2.0 A 0 False 225 1126 4ChoiceRANK2 2.0 D 0 False 225 1127 4ChoiceRANK3 2.0 B 0 False 225 1128 4ChoiceRANK4 2.0 C 0 False 226 226 1130 4ChoiceRANK1 3.0 A 0 False 226 1131 4ChoiceRANK2 3.0 B 0 False 226 1132 4ChoiceRANK3 3.0 D 0 False 226 1133 4ChoiceRANK4 3.0 C 0 False 227 227 1135 4ChoiceRANK1

2.0 D 0 False

```
227 1136 4ChoiceRANK2
2.0 C 0 False
227 1137 4ChoiceRANK3
2.0 B 0 False
227 1138 4ChoiceRANK4
2.0 A 0 False
228 1140 4ChoiceRANK1
2.0 D 0 False
228 1141 4ChoiceRANK2
2.0 C 0 False
228 1142 4ChoiceRANK3
2.0 B 0 False
228 1143 4ChoiceRANK4
2.0 A 0 False
229
229 1145 4ChoiceRANK1
2.0 B 0 False
229 1146 4ChoiceRANK2
2.0 A 0 False
229 1147 4ChoiceRANK3
2.0 D 0 False
229 1148 4ChoiceRANK4
2.0 C 0 False
230 1150 4ChoiceRANK1
3.0 A 0
230 1151 4ChoiceRANK2
3.0 B 0
230 1152 4ChoiceRANK3
3.0 D 0
230 1153 4ChoiceRANK4
3.0 C 0
231
231 1155 4ChoiceRANK1
1.0 B 0 False
231 1156 4ChoiceRANK2
1.0 A 0 False
231 1157 4ChoiceRANK3
1.0 D 0 False
231 1158 4ChoiceRANK4
1.0 C 0 False
232
232 1160 4ChoiceRANK1
3.0 B 0 False
232 1161 4ChoiceRANK2
3.0 A 0 False
232 1162 4ChoiceRANK3
3.0 D 0 False
232 1163 4ChoiceRANK4
3.0 C 0 False
233
233 1165 4ChoiceRANK1
3.0 D 0
233 1166 4ChoiceRANK2
3.0 C 0
233 1167 4ChoiceRANK3
3.0 B 0
233 1168 4ChoiceRANK4
3.0 A 0
234
234 1170 4ChoiceRANK1
1.0 A 0 False
234 1171 4ChoiceRANK2
1.0 B 0 False
234 1172 4ChoiceRANK3
1.0 C 0 False
234 1173 4ChoiceRANK4
1.0 D 0 False
235 1175 4ChoiceRANK1
2.0 A 0 False
235 1176 4ChoiceRANK2
2.0 D 0 False
235 1177 4ChoiceRANK3
2.0 B 0 False
235 1178 4ChoiceRANK4
2.0 C 0 False
236
236 1180 4ChoiceRANK1
2.0 C 0
236 1181 4ChoiceRANK2
2.0 B 0
236 1182 4ChoiceRANK3
2.0 A 0
236 1183 4ChoiceRANK4
error
237
237 1185 4ChoiceRANK1
2.0 B 0 False
237 1186 4ChoiceRANK2
2.0 C 0 False
237 1187 4ChoiceRANK3
```

```
2.0 D 0 False
237 1188 4ChoiceRANK4
2.0 A 0 False
238
238 1190 4ChoiceRANK1
1.0 A 0 False
238 1191 4ChoiceRANK2
1.0 B 0 False
238 1192 4ChoiceRANK3
1.0 D 0 False
238 1193 4ChoiceRANK4
1.0 C 0 False
239
239 1195 4ChoiceRANK1
2.0 B 0 False
239 1196 4ChoiceRANK2
2.0 A 0 False
239 1197 4ChoiceRANK3
2.0 C 0 False
239 1198 4ChoiceRANK4
2.0 D 0 False
240
240 1200 4ChoiceRANK1
2.0 D 0 False
240 1201 4ChoiceRANK2
2.0 B 0 False
240 1202 4ChoiceRANK3
2.0 A 0 False
240 1203 4ChoiceRANK4
2.0 C 0 False
241
241 1205 4ChoiceRANK1
2.0 B 0
241 1206 4ChoiceRANK2
2.0 C 0
241 1207 4ChoiceRANK3
2.0 A 0
241 1208 4ChoiceRANK4
error
242
242 1210 4ChoiceRANK1
3.0 D 0 False
242 1211 4ChoiceRANK2
3.0 C 0 False
242 1212 4ChoiceRANK3
3.0 B 0 False
242 1213 4ChoiceRANK4
3.0 A 0 False
243
243 1215 4ChoiceRANK1
3.0 D 0 False
243 1216 4ChoiceRANK2
3.0 B 0 False
243 1217 4ChoiceRANK3
3.0 A 0 False
243 1218 4ChoiceRANK4
3.0 C 0 False
244
244 1220 4ChoiceRANK1
3.0 A 0 False
244 1221 4ChoiceRANK2
3.0 D 0 False
244 1222 4ChoiceRANK3
3.0 C 0 False
244 1223 4ChoiceRANK4
3.0 B 0 False
245
245 1225 4ChoiceRANK1
2.0 D 0 False
245 1226 4ChoiceRANK2
2.0 C 0 False
245 122/ 4ChoiceRANK3
2.0 A 0 False
245 1228 4ChoiceRANK4
2.0 B 0 False
246
246 1230 4ChoiceRANK1
2.0 C 0
246 1231 4ChoiceRANK2
2.0 B 0
246 1232 4ChoiceRANK3
2.0 A 0
246 1233 4ChoiceRANK4
error
247
247 1235 4ChoiceRANK1
1.0 A 0 False
247 1236 4ChoiceRANK2
1.0 D 0 False
247 1237 4ChoiceRANK3
1.0 C 0 False
```

247 1238 4ChoiceRANK4 1.0 B 0 False

248 248 1240 4ChoiceRANK1 2.0 A 0 False 248 1241 4ChoiceRANK2 2.0 B 0 False 248 1242 4ChoiceRANK3 2.0 C 0 False 248 1243 4ChoiceRANK4 2.0 D 0 False 249 1245 4ChoiceRANK1 2.0 C 0 False 249 1246 4ChoiceRANK2 2.0 A 0 False 249 1247 4ChoiceRANK3 2.0 D 0 False 249 1248 4ChoiceRANK4 2.0 B 0 False 250 250 1250 4ChoiceRANK1 3.0 D 0 250 1251 4ChoiceRANK2 3.0 B 0 250 1252 4ChoiceRANK3 3.0 C 0 250 1253 4ChoiceRANK4 3.0 A 0 251 251 1255 4ChoiceRANK1 2.0 B 0 False 251 1256 4ChoiceRANK2 2.0 C 0 False 251 1257 4ChoiceRANK3 2.0 A 0 False 251 1258 4ChoiceRANK4 2.0 D 0 False 252 252 1260 4ChoiceRANK1 2.0 B 0 False 252 1261 4ChoiceRANK2 2.0 A 0 False 252 1262 4ChoiceRANK3 2.0 C 0 False 252 1263 4ChoiceRANK4 2.0 D 0 False 253 253 1265 4ChoiceRANK1 2.0 A 0 253 1266 4ChoiceRANK2 2.0 C 0 253 1267 4ChoiceRANK3 2.0 B 0 253 1268 4ChoiceRANK4 254 254 1270 4ChoiceRANK1 2.0 B 0 254 1271 4ChoiceRANK2 2.0 C 0 254 1272 4ChoiceRANK3 2.0 A 0 254 1273 4ChoiceRANK4 error 255 255 1275 4ChoiceRANK1 4.0 A 0 False 255 1276 4ChoiceRANK2 4.0 B 0 False 255 1277 4ChoiceRANK3 4.0 C 0 False 255 1278 4ChoiceRANK4 4.0 D 0 False 256 256 1280 4ChoiceRANK1 2.0 B 0 256 1281 4ChoiceRANK2 2.0 A 0 256 1282 4ChoiceRANK3 2.0 C 0 256 1283 4ChoiceRANK4 error 257 257 1285 4ChoiceRANK1 2.0 C 0 False 257 1286 4ChoiceRANK2 2.0 A 0 False 257 1287 4ChoiceRANK3 2.0 D 0 False 257 1288 4ChoiceRANK4 2.0 B 0 False 258 258 1290 4ChoiceRANK1

3.0 A 0 False

```
258 1291 4ChoiceRANK2
3.0 C 0 False
258 1292 4ChoiceRANK3
3.0 B 0 False
258 1293 4ChoiceRANK4
3.0 D 0 False
259 1295 4ChoiceRANK1
2.0 B 0 False
259 1296 4ChoiceRANK2
2.0 D 0 False
259 1297 4ChoiceRANK3
2.0 A 0 False
259 1298 4ChoiceRANK4
2.0 C 0 False
260
260 1300 4ChoiceRANK1
2.0 D 0 False
260 1301 4ChoiceRANK2
2.0 B 0 False
260 1302 4ChoiceRANK3
2.0 A 0 False
260 1303 4ChoiceRANK4
2.0 C 0 False
261 1305 4ChoiceRANK1
2.0 D 0 False
261 1306 4ChoiceRANK2
2.0 B 0 False
261 1307 4ChoiceRANK3
2.0 A 0 False
261 1308 4ChoiceRANK4
2.0 C 0 False
262
262 1310 4ChoiceRANK1
2.0 A 0
262 1311 4ChoiceRANK2
2.0 C 0
262 1312 4ChoiceRANK3
2.0 B 0
262 1313 4ChoiceRANK4
error
263
263 1315 4ChoiceRANK1
3.0 C 0 False
263 1316 4ChoiceRANK2
3.0 D 0 False
263 1317 4ChoiceRANK3
3.0 A 0 False
263 1318 4ChoiceRANK4
3.0 B 0 False
264
264 1320 4ChoiceRANK1
2.0 D 0 False
264 1321 4ChoiceRANK2
2.0 B 0 False
264 1322 4ChoiceRANK3
2.0 A 0 False
264 1323 4ChoiceRANK4
2.0 C 0 False
265
265 1325 4ChoiceRANK1
2.0 A 0
265 1326 4ChoiceRANK2
2.0 B 0
265 1327 4ChoiceRANK3
2.0 C 0
265 1328 4ChoiceRANK4
error
266 1330 4ChoiceRANK1
1.0 B 0 False
266 1331 4ChoiceRANK2
1.0 D 0 False
266 1332 4ChoiceRANK3
1.0 A 0 False
266 1333 4ChoiceRANK4
1.0 C 0 False
267
267 1335 4ChoiceRANK1
1.0 A 0 False
267 1336 4ChoiceRANK2
1.0 C 0 False
267 1337 4ChoiceRANK3
1.0 B 0 False
267 1338 4ChoiceRANK4
1.0 D 0 False
268
268 1340 4ChoiceRANK1
2.0 C 0
268 1341 4ChoiceRANK2
2.0 B 0
268 1342 4ChoiceRANK3
```

```
2.0 A 0
268 1343 4ChoiceRANK4
269 1345 4ChoiceRANK1
3.0 A 0 False
269 1346 4ChoiceRANK2
3.0 B 0 False
269 1347 4ChoiceRANK3
3.0 D 0 False
269 1348 4ChoiceRANK4
3.0 C 0 False
270
270 1350 4ChoiceRANK1
2.0 D 0 False
270 1351 4ChoiceRANK2
2.0 C 0 False
270 1352 4ChoiceRANK3
2.0 B 0 False
270 1353 4ChoiceRANK4
2.0 A 0 False
271
271 1355 4ChoiceRANK1
3.0 B 0 False
271 1356 4ChoiceRANK2
3.0 D 0 False
271 1357 4ChoiceRANK3
3.0 A 0 False
271 1358 4ChoiceRANK4
3.0 C 0 False
272 1360 4ChoiceRANK1
3.0 B 0 False
272 1361 4ChoiceRANK2
3.0 D 0 False
272 1362 4ChoiceRANK3
3.0 C 0 False
272 1363 4ChoiceRANK4
3.0 A 0 False
273
273 1365 4ChoiceRANK1
2.0 A 0
273 1366 4ChoiceRANK2
2.0 B 0
273 1367 4ChoiceRANK3
2.0 C 0
273 1368 4ChoiceRANK4
error
274
274 1370 4ChoiceRANK1
2.0 B 0 False
274 1371 4ChoiceRANK2
2.0 D 0 False
274 1372 4ChoiceRANK3
2.0 C 0 False
274 1373 4ChoiceRANK4
2.0 A 0 False
275
275 1375 4ChoiceRANK1
2.0 D 0 False
275 1376 4ChoiceRANK2
2.0 C 0 False
275 1377 4ChoiceRANK3
2.0 B 0 False
275 1378 4ChoiceRANK4
2.0 A 0 False
276
276 1380 4ChoiceRANK1
2.0 B 0
276 1381 4ChoiceRANK2
2.0 A 0
276 1382 4ChoiceRANK3
2.0 C 0
276 1383 4ChoiceRANK4
error
277
277 1385 4ChoiceRANK1
2.0 C 0
277 1386 4ChoiceRANK2
2.0 B 0
277 1387 4ChoiceRANK3
2.0 A 0
277 1388 4ChoiceRANK4
error
278
278 1390 4ChoiceRANK1
2.0 B 0 False
278 1391 4ChoiceRANK2
2.0 D 0 False
278 1392 4ChoiceRANK3
2.0 A 0 False
278 1393 4ChoiceRANK4
2.0 C 0 False
```

```
279
279 1395 4ChoiceRANK1
4.0 B 0 False
279 1396 4ChoiceRANK2
4.0 A 0 False
279 1397 4ChoiceRANK3
4.0 C 0 False
279 1398 4ChoiceRANK4
4.0 D 0 False
280 1400 4ChoiceRANK1
3.0 D 0
280 1401 4ChoiceRANK2
3.0 B 0
280 1402 4ChoiceRANK3
3.0 C 0
280 1403 4ChoiceRANK4
3.0 A 0
281
281 1405 4ChoiceRANK1
2.0 D 0 False
281 1406 4ChoiceRANK2
2.0 B 0 False
281 1407 4ChoiceRANK3
2.0 A 0 False
281 1408 4ChoiceRANK4
2.0 C 0 False
282
282 1410 4ChoiceRANK1
2.0 A 0 False
282 1411 4ChoiceRANK2
2.0 B 0 False
282 1412 4ChoiceRANK3
2.0 C 0 False
282 1413 4ChoiceRANK4
2.0 D 0 False
283
283 1415 4ChoiceRANK1
1.0 A 0 False
283 1416 4ChoiceRANK2
1.0 B 0 False
283 1417 4ChoiceRANK3
1.0 C 0 False
283 1418 4ChoiceRANK4
1.0 D 0 False
284
284 1420 4ChoiceRANK1
2.0 A 0
284 1421 4ChoiceRANK2
2.0 B 0
284 1422 4ChoiceRANK3
2.0 C 0
284 1423 4ChoiceRANK4
error
285
285 1425 4ChoiceRANK1
3.0 B 0 False
285 1426 4ChoiceRANK2
3.0 D 0 False
285 1427 4ChoiceRANK3
3.0 A 0 False
285 1428 4ChoiceRANK4
3.0 C 0 False
286
286 1430 4ChoiceRANK1
3.0 B 0 False
286 1431 4ChoiceRANK2
3.0 A 0 False
286 1432 4ChoiceRANK3
3.0 D 0 False
286 1433 4ChoiceRANK4
3.0 C 0 False
287
287 1435 4ChoiceRANK1
2.0 B 0 False
287 1436 4ChoiceRANK2
2.0 A 0 False
287 1437 4ChoiceRANK3
2.0 D 0 False
287 1438 4ChoiceRANK4
2.0 C 0 False
288
288 1440 4ChoiceRANK1
2.0 C 0
288 1441 4ChoiceRANK2
2.0 B 0
288 1442 4ChoiceRANK3
2.0 A 0
288 1443 4ChoiceRANK4
error
289
289 1445 4ChoiceRANK1
3.0 C 0
```

```
289 1446 4ChoiceRANK2
3.0 B 0
289 1447 4ChoiceRANK3
3.0 A 0
289 1448 4ChoiceRANK4
3.0 D 0
290
290 1450 4ChoiceRANK1
3.0 A 0 False
290 1451 4ChoiceRANK2
3.0 B 0 False
290 1452 4ChoiceRANK3
3.0 C 0 False
290 1453 4ChoiceRANK4
3.0 D 0 False
291
291 1455 4ChoiceRANK1
2.0 C 0 False
291 1456 4ChoiceRANK2
2.0 B 0 False
291 1457 4ChoiceRANK3
2.0 A 0 False
291 1458 4ChoiceRANK4
2.0 D 0 False
292 1460 4ChoiceRANK1
2.0 C 0
292 1461 4ChoiceRANK2
2.0 B 0
292 1462 4ChoiceRANK3
2.0 A 0
292 1463 4ChoiceRANK4
error
293
293 1465 4ChoiceRANK1
2.0 A 0 False
293 1466 4ChoiceRANK2
2.0 D 0 False
293 1467 4ChoiceRANK3
2.0 C 0 False
293 1468 4ChoiceRANK4
2.0 B 0 False
294
294 1470 4ChoiceRANK1
3.0 A 0 False
294 1471 4ChoiceRANK2
3.0 B 0 False
294 1472 4ChoiceRANK3
3.0 C 0 False
294 1473 4ChoiceRANK4
3.0 D 0 False
295
295 1475 4ChoiceRANK1
2.0 A 0 False
295 1476 4ChoiceRANK2
2.0 B 0 False
295 1477 4ChoiceRANK3
2.0 D 0 False
295 1478 4ChoiceRANK4
2.0 C 0 False
296
296 1480 4ChoiceRANK1
3.0 B 0 False
296 1481 4ChoiceRANK2
3.0 A 0 False
296 1482 4ChoiceRANK3
3.0 D 0 False
296 1483 4ChoiceRANK4
3.0 C 0 False
297 1485 4ChoiceRANK1
3.0 A 0 False
297 1486 4ChoiceRANK2
3.0 B 0 False
297 1487 4ChoiceRANK3
3.0 C 0 False
297 1488 4ChoiceRANK4
3.0 D 0 False
298
298 1490 4ChoiceRANK1
3.0 B 0 False
298 1491 4ChoiceRANK2
3.0 A 0 False
298 1492 4ChoiceRANK3
3.0 D 0 False
298 1493 4ChoiceRANK4
3.0 C 0 False
299
299 1495 4ChoiceRANK1
3.0 B 0 False
299 1496 4ChoiceRANK2
3.0 A 0 False
299 1497 4ChoiceRANK3
```

```
3.0 D 0 False
299 1498 4ChoiceRANK4
3.0 C 0 False
300 1500 4ChoiceRANK1
1.0 D 0 False
300 1501 4ChoiceRANK2
1.0 C 0 False
300 1502 4ChoiceRANK3
1.0 B 0 False
300 1503 4ChoiceRANK4
1.0 A 0 False
301
301 1505 4ChoiceRANK1
1.0 B 0 False
301 1506 4ChoiceRANK2
1.0 D 0 False
301 1507 4ChoiceRANK3
1.0 A 0 False
301 1508 4ChoiceRANK4
1.0 C 0 False
302
302 1510 4ChoiceRANK1
3.0 A 0 False
302 1511 4ChoiceRANK2
3.0 B 0 False
302 1512 4ChoiceRANK3
3.0 C 0 False
302 1513 4ChoiceRANK4
3.0 D 0 False
303 1515 4ChoiceRANK1
2.0 B 0 False
303 1516 4ChoiceRANK2
2.0 A 0 False
303 1517 4ChoiceRANK3
2.0 D 0 False
303 1518 4ChoiceRANK4
2.0 C 0 False
304
304 1520 4ChoiceRANK1
2.0 B 0 False
304 1521 4ChoiceRANK2
2.0 A 0 False
304 1522 4ChoiceRANK3
2.0 D 0 False
304 1523 4ChoiceRANK4
2.0 C 0 False
305
305 1525 4ChoiceRANK1
3.0 A 0 False
305 1526 4ChoiceRANK2
3.0 D 0 False
305 1527 4ChoiceRANK3
3.0 B 0 False
305 1528 4ChoiceRANK4
3.0 C 0 False
306 1530 4ChoiceRANK1
3.0 B 0 False
306 1531 4ChoiceRANK2
3.0 D 0 False
306 1532 4ChoiceRANK3
3.0 A 0 False
306 1533 4ChoiceRANK4
3.0 C 0 False
307
307 1535 4ChoiceRANK1
3.0 A 0 False
307 1536 4ChoiceRANK2
3.0 B 0 False
307 1537 4ChoiceRANK3
3.0 D 0 False
307 1538 4ChoiceRANK4
3.0 C 0 False
308
308 1540 4ChoiceRANK1
2.0 B 0
308 1541 4ChoiceRANK2
2.0 A 0
308 1542 4ChoiceRANK3
2.0 C 0
308 1543 4ChoiceRANK4
error
309
309 1545 4ChoiceRANK1
3.0 B 0 False
309 1546 4ChoiceRANK2
3.0 A 0 False
309 1547 4ChoiceRANK3
3.0 D 0 False
```

309 1548 4ChoiceRANK4 3.0 C 0 False

```
310
310 1550 4ChoiceRANK1
2.0 C 0
310 1551 4ChoiceRANK2
2.0 B 0
310 1552 4ChoiceRANK3
2.0 A 0
310 1553 4ChoiceRANK4
error
311
311 1555 4ChoiceRANK1
2.0 B 0
311 1556 4ChoiceRANK2
2.0 C 0
311 1557 4ChoiceRANK3
2.0 A 0
311 1558 4ChoiceRANK4
error
312
312 1560 4ChoiceRANK1
1.0 D 0 False
312 1561 4ChoiceRANK2
1.0 C 0 False
312 1562 4ChoiceRANK3
1.0 B 0 False
312 1563 4ChoiceRANK4
1.0 A 0 False
313
313 1565 4ChoiceRANK1
2.0 B 0 False
313 1566 4ChoiceRANK2
2.0 D 0 False
313 1567 4ChoiceRANK3
2.0 A 0 False
313 1568 4ChoiceRANK4
2.0 C 0 False
314
314 1570 4ChoiceRANK1
2.0 C 0
314 1571 4ChoiceRANK2
2.0 A 0
314 1572 4ChoiceRANK3
2.0 B 0
314 1573 4ChoiceRANK4
error
315
315 1575 4ChoiceRANK1
4.0 A 0 False
315 1576 4ChoiceRANK2
4.0 B 0 False
315 1577 4ChoiceRANK3
4.0 C 0 False
315 1578 4ChoiceRANK4
4.0 D 0 False
316
316 1580 4ChoiceRANK1
2.0 A 0 False
316 1581 4ChoiceRANK2
2.0 D 0 False
316 1582 4ChoiceRANK3
2.0 C 0 False
316 1583 4ChoiceRANK4
2.0 B 0 False
317
317 1585 4ChoiceRANK1
1.0 B 0 False
317 1586 4ChoiceRANK2
1.0 A 0 False
317 1587 4ChoiceRANK3
1.0 D 0 False
317 1588 4ChoiceRANK4
1.0 C 0 False
318
318 1590 4ChoiceRANK1
2.0 B 0 False
318 1591 4ChoiceRANK2
2.0 A 0 False
318 1592 4ChoiceRANK3
2.0 D 0 False
318 1593 4ChoiceRANK4
2.0 C 0 False
319
319 1595 4ChoiceRANK1
2.0 A 0 False
319 1596 4ChoiceRANK2
2.0 C 0 False
319 1597 4ChoiceRANK3
2.0 B 0 False
319 1598 4ChoiceRANK4
2.0 D 0 False
320
```

320 1600 4ChoiceRANK1

2.0 C 0

```
320 1601 4ChoiceRANK2
2.0 B 0
320 1602 4ChoiceRANK3
2.0 A 0
320 1603 4ChoiceRANK4
error
321
321 1605 4ChoiceRANK1
3.0 B 0 False
321 1606 4ChoiceRANK2
3.0 A 0 False
321 1607 4ChoiceRANK3
3.0 D 0 False
321 1608 4ChoiceRANK4
3.0 C 0 False
322
322 1610 4ChoiceRANK1
3.0 B 0 False
322 1611 4ChoiceRANK2
3.0 A 0 False
322 1612 4ChoiceRANK3
3.0 D 0 False
322 1613 4ChoiceRANK4
3.0 C 0 False
323 1615 4ChoiceRANK1
2.0 B 0 False
323 1616 4ChoiceRANK2
2.0 C 0 False
323 1617 4ChoiceRANK3
2.0 A 0 False
323 1618 4ChoiceRANK4
2.0 D 0 False
324
324 1620 4ChoiceRANK1
2.0 C 0
324 1621 4ChoiceRANK2
2.0 B 0
324 1622 4ChoiceRANK3
2.0 A 0
324 1623 4ChoiceRANK4
error
325
325 1625 4ChoiceRANK1
2.0 B 0 False
325 1626 4ChoiceRANK2
2.0 C 0 False
325 1627 4ChoiceRANK3
2.0 A 0 False
325 1628 4ChoiceRANK4
2.0 D 0 False
326
326 1630 4ChoiceRANK1
3.0 B 0 False
326 1631 4ChoiceRANK2
3.0 D 0 False
326 1632 4ChoiceRANK3
3.0 A 0 False
326 1633 4ChoiceRANK4
3.0 C 0 False
327
327 1635 4ChoiceRANK1
3.0 A 0 False
327 1636 4ChoiceRANK2
3.0 B 0 False
327 1637 4ChoiceRANK3
3.0 D 0 False
327 1638 4ChoiceRANK4
3.0 C 0 False
328 1640 4ChoiceRANK1
3.0 D 0 False
328 1641 4ChoiceRANK2
3.0 B 0 False
328 1642 4ChoiceRANK3
3.0 A 0 False
328 1643 4ChoiceRANK4
3.0 C 0 False
329
329 1645 4ChoiceRANK1
2.0 D 0 False
329 1646 4ChoiceRANK2
2.0 A 0 False
329 1647 4ChoiceRANK3
2.0 B 0 False
329 1648 4ChoiceRANK4
2.0 C 0 False
330
330 1650 4ChoiceRANK1
1.0 B 0 False
330 1651 4ChoiceRANK2
1.0 D 0 False
```

330 1652 4ChoiceRANK3

```
1.0 C 0 False
330 1653 4ChoiceRANK4
1.0 A 0 False
331
331 1655 4ChoiceRANK1
3.0 B 0 False
331 1656 4ChoiceRANK2
3.0 D 0 False
331 1657 4ChoiceRANK3
3.0 A 0 False
331 1658 4ChoiceRANK4
3.0 C 0 False
332
332 1660 4ChoiceRANK1
3.0 D 0
332 1661 4ChoiceRANK2
3.0 B 0
332 1662 4ChoiceRANK3
3.0 C 0
332 1663 4ChoiceRANK4
3.0 A 0
333
333 1665 4ChoiceRANK1
2.0 D 0 False
333 1666 4ChoiceRANK2
2.0 B 0 False
333 1667 4ChoiceRANK3
2.0 C 0 False
333 1668 4ChoiceRANK4
2.0 A 0 False
334
334 1670 4ChoiceRANK1
4.0 A 0 False
334 1671 4ChoiceRANK2
4.0 B 0 False
334 1672 4ChoiceRANK3
4.0 C 0 False
334 1673 4ChoiceRANK4
4.0 D 0 False
335
335 1675 4ChoiceRANK1
3.0 D 0
335 1676 4ChoiceRANK2
3.0 B 0
335 1677 4ChoiceRANK3
3.0 C 0
335 1678 4ChoiceRANK4
3.0 A 0
336
336 1680 4ChoiceRANK1
3.0 B 0 False
336 1681 4ChoiceRANK2
3.0 A 0 False
336 1682 4ChoiceRANK3
3.0 C 0 False
336 1683 4ChoiceRANK4
3.0 D 0 False
337
337 1685 4ChoiceRANK1
3.0 B 0 False
337 1686 4ChoiceRANK2
3.0 D 0 False
337 1687 4ChoiceRANK3
3.0 C 0 False
337 1688 4ChoiceRANK4
3.0 A 0 False
338
338 1690 4ChoiceRANK1
2.0 A 0
338 1691 4ChoiceRANK2
2.0 B 0
338 1692 4ChoiceRANK3
2.0 C 0
338 1693 4ChoiceRANK4
error
339
339 1695 4ChoiceRANK1
2.0 B 0 False
339 1696 4ChoiceRANK2
2.0 A 0 False
339 1697 4ChoiceRANK3
2.0 D 0 False
339 1698 4ChoiceRANK4
2.0 C 0 False
340
340 1700 4ChoiceRANK1
1.0 B 0 False
340 1701 4ChoiceRANK2
1.0 D 0 False
340 1702 4ChoiceRANK3
1.0 A 0 False
```

340 1703 4ChoiceRANK4

1.0 C 0 False

341 341 1705 4ChoiceRANK1 2.0 C 0 341 1706 4ChoiceRANK2 2.0 A 0 341 1707 4ChoiceRANK3 2.0 B 0 341 1708 4ChoiceRANK4 error 342 1710 4ChoiceRANK1 3.0 C 0 False 342 1711 4ChoiceRANK2 3.0 D 0 False 342 1712 4ChoiceRANK3 3.0 A 0 False 342 1713 4ChoiceRANK4 3.0 B 0 False 343 343 1715 4ChoiceRANK1 3.0 D 0 False 343 1716 4ChoiceRANK2 3.0 C 0 False 343 1717 4ChoiceRANK3 3.0 B 0 False 343 1718 4ChoiceRANK4 3.0 A 0 False 344 344 1720 4ChoiceRANK1 2.0 B 0 False 344 1721 4ChoiceRANK2 2.0 D 0 False 344 1722 4ChoiceRANK3 2.0 A 0 False 344 1723 4ChoiceRANK4 2.0 C 0 False 345 345 1725 4ChoiceRANK1 1.0 B 0 False 345 1726 4ChoiceRANK2 1.0 D 0 False 345 1727 4ChoiceRANK3 1.0 A 0 False 345 1728 4ChoiceRANK4 1.0 C 0 False 346 346 1730 4ChoiceRANK1 1.0 A 0 False 346 1731 4ChoiceRANK2 1.0 D 0 False 346 1732 4ChoiceRANK3 1.0 C 0 False 346 1733 4ChoiceRANK4 1.0 B 0 False 347 347 1735 4ChoiceRANK1 2.0 A 0 347 1736 4ChoiceRANK2 2.0 C 0 347 1737 4ChoiceRANK3 2.0 B 0 347 1738 4ChoiceRANK4 error 348 348 1740 4ChoiceRANK1 2.0 A 0 False 348 1741 4ChoiceRANK2 2.0 B 0 False 348 1742 4ChoiceRANK3 2.0 C 0 False 348 1743 4ChoiceRANK4 2.0 D 0 False 349 349 1745 4ChoiceRANK1 2.0 A 0 False 349 1746 4ChoiceRANK2 2.0 C 0 False 349 1747 4ChoiceRANK3 2.0 B 0 False 349 1748 4ChoiceRANK4 2.0 D 0 False 350 350 1750 4ChoiceRANK1 3.0 D 0 350 1751 4ChoiceRANK2 3.0 B 0 350 1752 4ChoiceRANK3 3.0 A 0 350 1753 4ChoiceRANK4 3.0 C 0 351 351 1755 4ChoiceRANK1

2.0 C 0 False

```
351 1756 4ChoiceRANK2
2.0 D 0 False
351 1757 4ChoiceRANK3
2.0 A 0 False
351 1758 4ChoiceRANK4
2.0 B 0 False
352 1760 4ChoiceRANK1
3.0 D 0
352 1761 4ChoiceRANK2
3.0 C 0
352 1762 4ChoiceRANK3
3.0 B 0
352 1763 4ChoiceRANK4
3.0 A 0
353
353 1765 4ChoiceRANK1
3.0 B 0 False
353 1766 4ChoiceRANK2
3.0 A 0 False
353 1767 4ChoiceRANK3
3.0 C 0 False
353 1768 4ChoiceRANK4
3.0 D 0 False
354 1770 4ChoiceRANK1
3.0 D 0 False
354 1771 4ChoiceRANK2
3.0 B 0 False
354 1772 4ChoiceRANK3
3.0 A 0 False
354 1773 4ChoiceRANK4
3.0 C 0 False
355
355 1775 4ChoiceRANK1
2.0 B 0 False
355 1776 4ChoiceRANK2
2.0 A 0 False
355 1777 4ChoiceRANK3
2.0 D 0 False
355 1778 4ChoiceRANK4
2.0 C 0 False
356
356 1780 4ChoiceRANK1
2.0 C 0
356 1781 4ChoiceRANK2
2.0 B 0
356 1782 4ChoiceRANK3
2.0 A 0
356 1783 4ChoiceRANK4
error
357
357 1785 4ChoiceRANK1
357 1786 4ChoiceRANK2
2.0 B 0
357 1787 4ChoiceRANK3
2.0 A 0
357 1788 4ChoiceRANK4
error
358
358 1790 4ChoiceRANK1
4.0 B 0 False
358 1791 4ChoiceRANK2
4.0 A 0 False
358 1792 4ChoiceRANK3
4.0 D 0 False
358 1793 4ChoiceRANK4
4.0 C 0 False
359 1795 4ChoiceRANK1
2.0 C 0
359 1796 4ChoiceRANK2
2.0 B 0
359 1797 4ChoiceRANK3
2.0 A 0
359 1798 4ChoiceRANK4
error
360
360 1800 4ChoiceRANK1
2.0 A 0 False
360 1801 4ChoiceRANK2
2.0 B 0 False
360 1802 4ChoiceRANK3
2.0 D 0 False
360 1803 4ChoiceRANK4
2.0 C 0 False
361
361 1805 4ChoiceRANK1
2.0 D 0 False
361 1806 4ChoiceRANK2
2.0 C 0 False
361 1807 4ChoiceRANK3
```

```
2.0 B 0 False
361 1808 4ChoiceRANK4
2.0 A 0 False
362 1810 4ChoiceRANK1
3.0 D 0
362 1811 4ChoiceRANK2
3.0 B 0
362 1812 4ChoiceRANK3
3.0 A 0
362 1813 4ChoiceRANK4
3.0 C 0
363
363 1815 4ChoiceRANK1
2.0 B 0 False
363 1816 4ChoiceRANK2
2.0 A 0 False
363 1817 4ChoiceRANK3
2.0 D 0 False
363 1818 4ChoiceRANK4
2.0 C 0 False
364 1820 4ChoiceRANK1
2.0 B 0 False
364 1821 4ChoiceRANK2
2.0 D 0 False
364 1822 4ChoiceRANK3
2.0 A 0 False
364 1823 4ChoiceRANK4
2.0 C 0 False
365
365 1825 4ChoiceRANK1
2.0 D 0 False
365 1826 4ChoiceRANK2
2.0 C 0 False
365 1827 4ChoiceRANK3
2.0 B 0 False
365 1828 4ChoiceRANK4
2.0 A 0 False
366
366 1830 4ChoiceRANK1
2.0 B 0 False
366 1831 4ChoiceRANK2
2.0 A 0 False
366 1832 4ChoiceRANK3
2.0 D 0 False
366 1833 4ChoiceRANK4
2.0 C 0 False
367
367 1835 4ChoiceRANK1
2.0 B 0 False
367 1836 4ChoiceRANK2
2.0 D 0 False
367 1837 4ChoiceRANK3
2.0 A 0 False
367 1838 4ChoiceRANK4
2.0 C 0 False
368
368 1840 4ChoiceRANK1
3.0 A 0
368 1841 4ChoiceRANK2
3.0 B 0
368 1842 4ChoiceRANK3
3.0 D 0
368 1843 4ChoiceRANK4
3.0 C 0
369
369 1845 4ChoiceRANK1
2.0 B 0 False
369 1846 4ChoiceRANK2
2.0 A 0 False
369 1847 4ChoiceRANK3
2.0 D 0 False
369 1848 4ChoiceRANK4
2.0 C 0 False
370
370 1850 4ChoiceRANK1
2.0 C 0 False
370 1851 4ChoiceRANK2
2.0 D 0 False
370 1852 4ChoiceRANK3
2.0 B 0 False
370 1853 4ChoiceRANK4
2.0 A 0 False
371 1855 4ChoiceRANK1
2.0 A 0
371 1856 4ChoiceRANK2
2.0 B 0
371 1857 4ChoiceRANK3
2.0 C 0
```

371 1858 4ChoiceRANK4

error

```
372
372 1860 4ChoiceRANK1
2.0 A 0 False
372 1861 4ChoiceRANK2
2.0 D 0 False
372 1862 4ChoiceRANK3
2.0 B 0 False
372 1863 4ChoiceRANK4
2.0 C 0 False
373 1865 4ChoiceRANK1
2.0 D 0 False
373 1866 4ChoiceRANK2
2.0 B 0 False
373 1867 4ChoiceRANK3
2.0 C 0 False
373 1868 4ChoiceRANK4
2.0 A 0 False
374
374 1870 4ChoiceRANK1
2.0 B 0 False
374 1871 4ChoiceRANK2
2.0 D 0 False
374 1872 4ChoiceRANK3
2.0 A 0 False
374 1873 4ChoiceRANK4
2.0 C 0 False
375
375 1875 4ChoiceRANK1
2.0 C 0 False
375 1876 4ChoiceRANK2
2.0 A 0 False
375 1877 4ChoiceRANK3
2.0 D 0 False
375 1878 4ChoiceRANK4
2.0 B 0 False
376
376 1880 4ChoiceRANK1
2.0 C 0
376 1881 4ChoiceRANK2
2.0 A 0
376 1882 4ChoiceRANK3
2.0 B 0
376 1883 4ChoiceRANK4
error
377
377 1885 4ChoiceRANK1
3.0 A 0 False
377 1886 4ChoiceRANK2
3.0 B 0 False
377 1887 4ChoiceRANK3
3.0 C 0 False
377 1888 4ChoiceRANK4
3.0 D 0 False
378
378 1890 4ChoiceRANK1
2.0 C 0 False
378 1891 4ChoiceRANK2
2.0 B 0 False
378 1892 4ChoiceRANK3
2.0 A 0 False
378 1893 4ChoiceRANK4
2.0 D 0 False
379
379 1895 4ChoiceRANK1
2.0 A 0 False
379 1896 4ChoiceRANK2
2.0 B 0 False
379 1897 4ChoiceRANK3
2.0 D 0 False
379 1898 4ChoiceRANK4
2.0 C 0 False
380
380 1900 4ChoiceRANK1
2.0 B 0 False
380 1901 4ChoiceRANK2
2.0 D 0 False
380 1902 4ChoiceRANK3
2.0 A 0 False
380 1903 4ChoiceRANK4
2.0 C 0 False
381
381 1905 4ChoiceRANK1
2.0 A 0 False
381 1906 4ChoiceRANK2
2.0 B 0 False
381 1907 4ChoiceRANK3
2.0 C 0 False
381 1908 4ChoiceRANK4
2.0 D 0 False
382
382 1910 4ChoiceRANK1
```

3.0 C 0

```
382 1911 4ChoiceRANK2
3.0 A 0
382 1912 4ChoiceRANK3
3.0 B 0
382 1913 4ChoiceRANK4
3.0 D 0
383 1915 4ChoiceRANK1
2.0 B 0 False
383 1916 4ChoiceRANK2
2.0 C 0 False
383 1917 4ChoiceRANK3
2.0 D 0 False
383 1918 4ChoiceRANK4
2.0 A 0 False
384
384 1920 4ChoiceRANK1
2.0 D 0 False
384 1921 4ChoiceRANK2
2.0 A 0 False
384 1922 4ChoiceRANK3
2.0 B 0 False
384 1923 4ChoiceRANK4
2.0 C 0 False
385 1925 4ChoiceRANK1
2.0 A 0 False
385 1926 4ChoiceRANK2
2.0 C 0 False
385 1927 4ChoiceRANK3
2.0 B 0 False
385 1928 4ChoiceRANK4
2.0 D 0 False
386
386 1930 4ChoiceRANK1
3.0 B 0 False
386 1931 4ChoiceRANK2
3.0 A 0 False
386 1932 4ChoiceRANK3
3.0 C 0 False
386 1933 4ChoiceRANK4
3.0 D 0 False
387
387 1935 4ChoiceRANK1
2.0 B 0 False
387 1936 4ChoiceRANK2
2.0 D 0 False
387 1937 4ChoiceRANK3
2.0 A 0 False
387 1938 4ChoiceRANK4
2.0 C 0 False
388
388 1940 4ChoiceRANK1
1.0 B 0 False
388 1941 4ChoiceRANK2
1.0 A 0 False
388 1942 4ChoiceRANK3
1.0 C 0 False
388 1943 4ChoiceRANK4
1.0 D 0 False
389
389 1945 4ChoiceRANK1
2.0 A 0 False
389 1946 4ChoiceRANK2
2.0 C 0 False
389 1947 4ChoiceRANK3
2.0 D 0 False
389 1948 4ChoiceRANK4
2.0 B 0 False
390 1950 4ChoiceRANK1
2.0 B 0
390 1951 4ChoiceRANK2
2.0 C 0
390 1952 4ChoiceRANK3
2.0 A 0
390 1953 4ChoiceRANK4
error
391
391 1955 4ChoiceRANK1
3.0 B 0 False
391 1956 4ChoiceRANK2
3.0 A 0 False
391 1957 4ChoiceRANK3
3.0 D 0 False
391 1958 4ChoiceRANK4
3.0 C 0 False
392
392 1960 4ChoiceRANK1
1.0 C 0 False
392 1961 4ChoiceRANK2
1.0 A 0 False
392 1962 4ChoiceRANK3
```

```
1.0 D 0 False
392 1963 4ChoiceRANK4
1.0 B 0 False
393
393 1965 4ChoiceRANK1
2.0 D 0 False
393 1966 4ChoiceRANK2
2.0 B 0 False
393 1967 4ChoiceRANK3
2.0 C 0 False
393 1968 4ChoiceRANK4
2.0 A 0 False
394
394 1970 4ChoiceRANK1
2.0 B 0
394 1971 4ChoiceRANK2
2.0 C 0
394 1972 4ChoiceRANK3
2.0 A 0
394 1973 4ChoiceRANK4
error
395
395 1975 4ChoiceRANK1
3.0 D 0 False
395 1976 4ChoiceRANK2
3.0 C 0 False
395 1977 4ChoiceRANK3
3.0 B 0 False
395 1978 4ChoiceRANK4
3.0 A 0 False
396
396 1980 4ChoiceRANK1
2.0 B 0 False
396 1981 4ChoiceRANK2
2.0 D 0 False
396 1982 4ChoiceRANK3
2.0 A 0 False
396 1983 4ChoiceRANK4
2.0 C 0 False
397
397 1985 4ChoiceRANK1
2.0 A 0 False
397 1986 4ChoiceRANK2
2.0 D 0 False
397 1987 4ChoiceRANK3
2.0 B 0 False
397 1988 4ChoiceRANK4
2.0 C 0 False
398
398 1990 4ChoiceRANK1
2.0 A 0 False
398 1991 4ChoiceRANK2
2.0 B 0 False
398 1992 4ChoiceRANK3
2.0 D 0 False
398 1993 4ChoiceRANK4
2.0 C 0 False
399 1995 4ChoiceRANK1
2.0 A 0 False
399 1996 4ChoiceRANK2
2.0 B 0 False
399 1997 4ChoiceRANK3
2.0 C 0 False
399 1998 4ChoiceRANK4
2.0 D 0 False
400
400 2000 4ChoiceRANK1
3.0 C 0 False
400 2001 4ChoiceRANK2
3.0 D 0 False
400 2002 4ChoiceRANK3
3.0 A 0 False
400 2003 4ChoiceRANK4
3.0 B 0 False
401
401 2005 4ChoiceRANK1
3.0 A 0 False
401 2006 4ChoiceRANK2
3.0 B 0 False
401 2007 4ChoiceRANK3
3.0 D 0 False
401 2008 4ChoiceRANK4
3.0 C 0 False
402
402 2010 4ChoiceRANK1
2.0 B 0 False
402 2011 4ChoiceRANK2
2.0 A 0 False
402 2012 4ChoiceRANK3
2.0 D 0 False
```

402 2013 4ChoiceRANK4 2.0 C 0 False

```
403
403 2015 4ChoiceRANK1
3.0 D 0 False
403 2016 4ChoiceRANK2
3.0 B 0 False
403 2017 4ChoiceRANK3
3.0 A 0 False
403 2018 4ChoiceRANK4
3.0 C 0 False
404 2020 4ChoiceRANK1
3.0 A 0 False
404 2021 4ChoiceRANK2
3.0 D 0 False
404 2022 4ChoiceRANK3
3.0 B 0 False
404 2023 4ChoiceRANK4
3.0 C 0 False
405
405 2025 4ChoiceRANK1
2.0 B 0 False
405 2026 4ChoiceRANK2
2.0 D 0 False
405 2027 4ChoiceRANK3
2.0 A 0 False
405 2028 4ChoiceRANK4
2.0 C 0 False
406
406 2030 4ChoiceRANK1
3.0 A 0 False
406 2031 4ChoiceRANK2
3.0 B 0 False
406 2032 4ChoiceRANK3
3.0 D 0 False
406 2033 4ChoiceRANK4
3.0 C 0 False
407
407 2035 4ChoiceRANK1
2.0 D 0 False
407 2036 4ChoiceRANK2
2.0 A 0 False
407 2037 4ChoiceRANK3
2.0 B 0 False
407 2038 4ChoiceRANK4
2.0 C 0 False
408
408 2040 4ChoiceRANK1
1.0 B 0 False
408 2041 4ChoiceRANK2
1.0 A 0 False
408 2042 4ChoiceRANK3
1.0 C 0 False
408 2043 4ChoiceRANK4
1.0 D 0 False
409 2045 4ChoiceRANK1
3.0 B 0 False
409 2046 4ChoiceRANK2
3.0 D 0 False
409 2047 4ChoiceRANK3
3.0 A 0 False
409 2048 4ChoiceRANK4
3.0 C 0 False
410
410 2050 4ChoiceRANK1
3.0 C 0 False
410 2051 4ChoiceRANK2
3.0 B 0 False
410 2052 4ChoiceRANK3
3.0 D 0 False
410 2053 4ChoiceRANK4
3.0 A 0 False
411
411 2055 4ChoiceRANK1
3.0 B 0 False
411 2056 4ChoiceRANK2
3.0 A 0 False
411 2057 4ChoiceRANK3
3.0 D 0 False
411 2058 4ChoiceRANK4
3.0 C 0 False
412
412 2060 4ChoiceRANK1
3.0 D 0 False
412 2061 4ChoiceRANK2
3.0 B 0 False
412 2062 4ChoiceRANK3
3.0 C 0 False
412 2063 4ChoiceRANK4
3.0 A 0 False
413
413 2065 4ChoiceRANK1
```

2.0 C 0

```
413 2066 4ChoiceRANK2
2.0 A 0
413 2067 4ChoiceRANK3
2.0 B 0
413 2068 4ChoiceRANK4
error
414
414 2070 4ChoiceRANK1
1.0 D 0 False
414 2071 4ChoiceRANK2
1.0 B 0 False
414 2072 4ChoiceRANK3
1.0 A 0 False
414 2073 4ChoiceRANK4
1.0 C 0 False
415
415 2075 4ChoiceRANK1
3.0 C 0 False
415 2076 4ChoiceRANK2
3.0 A 0 False
415 2077 4ChoiceRANK3
3.0 D 0 False
415 2078 4ChoiceRANK4
3.0 B 0 False
416
416 2080 4ChoiceRANK1
3.0 A 0
416 2081 4ChoiceRANK2
3.0 B 0
416 2082 4ChoiceRANK3
3.0 C 0
416 2083 4ChoiceRANK4
3.0 D 0
417
417 2085 4ChoiceRANK1
3.0 B 0 False
417 2086 4ChoiceRANK2
3.0 D 0 False
417 2087 4ChoiceRANK3
3.0 A 0 False
417 2088 4ChoiceRANK4
3.0 C 0 False
418
418 2090 4ChoiceRANK1
2.0 B 0
418 2091 4ChoiceRANK2
2.0 A 0
418 2092 4ChoiceRANK3
2.0 C 0
418 2093 4ChoiceRANK4
error
419
419 2095 4ChoiceRANK1
1.0 B 0 False
419 2096 4ChoiceRANK2
1.0 D 0 False
419 2097 4ChoiceRANK3
1.0 C 0 False
419 2098 4ChoiceRANK4
1.0 A 0 False
420
420 2100 4ChoiceRANK1
3.0 D 0
420 2101 4ChoiceRANK2
3.0 B 0
420 2102 4ChoiceRANK3
3.0 A 0
420 2103 4ChoiceRANK4
3.0 C 0
421 2105 4ChoiceRANK1
2.0 D 0 False
421 2106 4ChoiceRANK2
2.0 B 0 False
421 2107 4ChoiceRANK3
2.0 A 0 False
421 2108 4ChoiceRANK4
2.0 C 0 False
422
422 2110 4ChoiceRANK1
2.0 A 0 False
422 2111 4ChoiceRANK2
2.0 B 0 False
422 2112 4ChoiceRANK3
2.0 C 0 False
422 2113 4ChoiceRANK4
2.0 D 0 False
423
423 2115 4ChoiceRANK1
1.0 C 0
423 2116 4ChoiceRANK2
1.0 A 0
```

423 2117 4ChoiceRANK3

1.0 B 0 423 2118 4ChoiceRANK4 424 424 2120 4ChoiceRANK1 2.0 B 0 False 424 2121 4ChoiceRANK2 2.0 C 0 False 424 2122 4ChoiceRANK3 2.0 D 0 False 424 2123 4ChoiceRANK4 2.0 A 0 False 425 425 2125 4ChoiceRANK1 2.0 D 0 False 425 2126 4ChoiceRANK2 2.0 B 0 False 425 2127 4ChoiceRANK3 2.0 A 0 False 425 2128 4ChoiceRANK4 2.0 C 0 False 426 426 2130 4ChoiceRANK1 2.0 B 0 False 426 2131 4ChoiceRANK2 2.0 D 0 False 426 2132 4ChoiceRANK3 2.0 A 0 False 426 2133 4ChoiceRANK4 2.0 C 0 False 427 2135 4ChoiceRANK1 3.0 D 0 False 427 2136 4ChoiceRANK2 3.0 C 0 False 427 2137 4ChoiceRANK3 3.0 B 0 False 427 2138 4ChoiceRANK4 3.0 A 0 False 428 428 2140 4ChoiceRANK1 2.0 C 0 False 428 2141 4ChoiceRANK2 2.0 D 0 False 428 2142 4ChoiceRANK3 2.0 A 0 False 428 2143 4ChoiceRANK4 2.0 B 0 False 429 429 2145 4ChoiceRANK1 2.0 B 0 429 2146 4ChoiceRANK2 2.0 A 0 429 2147 4ChoiceRANK3 2.0 C 0 429 2148 4ChoiceRANK4 error 430 430 2150 4ChoiceRANK1 2.0 B 0 430 2151 4ChoiceRANK2 2.0 A 0 430 2152 4ChoiceRANK3 2.0 C 0 430 2153 4ChoiceRANK4 error 431 431 2155 4ChoiceRANK1 2.0 A 0 False 431 2156 4ChoiceRANK2 2.0 B 0 False 431 2157 4ChoiceRANK3 2.0 C 0 False 431 2158 4ChoiceRANK4 2.0 D 0 False 432 432 2160 4ChoiceRANK1 2.0 D 0 False 432 2161 4ChoiceRANK2 2.0 B 0 False 432 2162 4ChoiceRANK3 2.0 C 0 False 432 2163 4ChoiceRANK4 2.0 A 0 False 433 433 2165 4ChoiceRANK1 1.0 A 0 False 433 2166 4ChoiceRANK2 1.0 B 0 False 433 2167 4ChoiceRANK3 1.0 C 0 False 433 2168 4ChoiceRANK4 1.0 D 0 False

434 434 2170 4ChoiceRANK1 1.0 A 0 False 434 2171 4ChoiceRANK2 1.0 C 0 False 434 2172 4ChoiceRANK3 1.0 B 0 False 434 2173 4ChoiceRANK4 1.0 D 0 False 435 2175 4ChoiceRANK1 3.0 A 0 False 435 2176 4ChoiceRANK2 3.0 B 0 False 435 2177 4ChoiceRANK3 3.0 D 0 False 435 2178 4ChoiceRANK4 3.0 C 0 False 436 436 2180 4ChoiceRANK1 2.0 A 0 False 436 2181 4ChoiceRANK2 2.0 B 0 False 436 2182 4ChoiceRANK3 2.0 C 0 False 436 2183 4ChoiceRANK4 2.0 D 0 False 437 437 2185 4ChoiceRANK1 3.0 B 0 False 437 2186 4ChoiceRANK2 3.0 A 0 False 437 2187 4ChoiceRANK3 3.0 D 0 False 437 2188 4ChoiceRANK4 3.0 C 0 False 438 438 2190 4ChoiceRANK1 2.0 A 0 False 438 2191 4ChoiceRANK2 2.0 B 0 False 438 2192 4ChoiceRANK3 2.0 C 0 False 438 2193 4ChoiceRANK4 2.0 D 0 False 439 439 2195 4ChoiceRANK1 2.0 D 0 False 439 2196 4ChoiceRANK2 2.0 C 0 False 439 2197 4ChoiceRANK3 2.0 A 0 False 439 2198 4ChoiceRANK4 2.0 B 0 False 440 440 2200 4ChoiceRANK1 2.0 A 0 False 440 2201 4ChoiceRANK2 2.0 B 0 False 440 2202 4ChoiceRANK3 2.0 C 0 False 440 2203 4ChoiceRANK4 2.0 D 0 False 441 441 2205 4ChoiceRANK1 2.0 D 0 False 441 2206 4ChoiceRANK2 2.0 B 0 False 441 2207 4ChoiceRANK3 2.0 C 0 False 441 2208 4ChoiceRANK4 2.0 A 0 False 442 2210 4ChoiceRANK1 3.0 D 0 False 442 2211 4ChoiceRANK2 3.0 A 0 False 442 2212 4ChoiceRANK3 3.0 C 0 False 442 2213 4ChoiceRANK4 3.0 B 0 False 443 443 2215 4ChoiceRANK1 2.0 B 0 False 443 2216 4ChoiceRANK2 2.0 A 0 False 443 2217 4ChoiceRANK3 2.0 D 0 False 443 2218 4ChoiceRANK4 2.0 C 0 False 444 444 2220 4ChoiceRANK1

2.0 B 0 False

```
444 2221 4ChoiceRANK2
2.0 D 0 False
444 2222 4ChoiceRANK3
2.0 C 0 False
444 2223 4ChoiceRANK4
2.0 A 0 False
445 2225 4ChoiceRANK1
2.0 C 0
445 2226 4ChoiceRANK2
2.0 A 0
445 2227 4ChoiceRANK3
2.0 B 0
445 2228 4ChoiceRANK4
error
446
446 2230 4ChoiceRANK1
2.0 A 0 False
446 2231 4ChoiceRANK2
2.0 B 0 False
446 2232 4ChoiceRANK3
2.0 D 0 False
446 2233 4ChoiceRANK4
2.0 C 0 False
447 2235 4ChoiceRANK1
3.0 A 0 False
447 2236 4ChoiceRANK2
3.0 B 0 False
447 2237 4ChoiceRANK3
3.0 D 0 False
447 2238 4ChoiceRANK4
3.0 C 0 False
448
448 2240 4ChoiceRANK1
2.0 C 0 False
448 2241 4ChoiceRANK2
2.0 D 0 False
448 2242 4ChoiceRANK3
2.0 A 0 False
448 2243 4ChoiceRANK4
2.0 B 0 False
449
449 2245 4ChoiceRANK1
4.0 C 0 False
449 2246 4ChoiceRANK2
4.0 B 0 False
449 2247 4ChoiceRANK3
4.0 D 0 False
449 2248 4ChoiceRANK4
4.0 A 0 False
450
450 2250 4ChoiceRANK1
2.0 C 0
450 2251 4ChoiceRANK2
2.0 B 0
450 2252 4ChoiceRANK3
2.0 A 0
450 2253 4ChoiceRANK4
error
451
451 2255 4ChoiceRANK1
3.0 D 0 False
451 2256 4ChoiceRANK2
3.0 C 0 False
451 2257 4ChoiceRANK3
3.0 B 0 False
451 2258 4ChoiceRANK4
3.0 A 0 False
452 2260 4ChoiceRANK1
3.0 D 0 False
452 2261 4ChoiceRANK2
3.0 B 0 False
452 2262 4ChoiceRANK3
3.0 C 0 False
452 2263 4ChoiceRANK4
3.0 A 0 False
453
453 2265 4ChoiceRANK1
4.0 C 0 False
453 2266 4ChoiceRANK2
4.0 D 0 False
453 2267 4ChoiceRANK3
4.0 B 0 False
453 2268 4ChoiceRANK4
4.0 A 0 False
454
454 2270 4ChoiceRANK1
2.0 B 0 False
454 2271 4ChoiceRANK2
2.0 A 0 False
454 2272 4ChoiceRANK3
```

```
2.0 D 0 False
454 2273 4ChoiceRANK4
2.0 C 0 False
455 2275 4ChoiceRANK1
3.0 C 0
455 2276 4ChoiceRANK2
3.0 D 0
455 2277 4ChoiceRANK3
3.0 B 0
455 2278 4ChoiceRANK4
3.0 A 0
456
456 2280 4ChoiceRANK1
2.0 B 0
456 2281 4ChoiceRANK2
2.0 A 0
456 2282 4ChoiceRANK3
2.0 C 0
456 2283 4ChoiceRANK4
error
457
457 2285 4ChoiceRANK1
2.0 C 0
457 2286 4ChoiceRANK2
2.0 B 0
457 2287 4ChoiceRANK3
2.0 A 0
457 2288 4ChoiceRANK4
error
458
458 2290 4ChoiceRANK1
4.0 C 0 False
458 2291 4ChoiceRANK2
4.0 D 0 False
458 2292 4ChoiceRANK3
4.0 A 0 False
458 2293 4ChoiceRANK4
4.0 B 0 False
459
459 2295 4ChoiceRANK1
1.0 B 0 False
459 2296 4ChoiceRANK2
1.0 A 0 False
459 2297 4ChoiceRANK3
1.0 D 0 False
459 2298 4ChoiceRANK4
1.0 C 0 False
460
460 2300 4ChoiceRANK1
2.0 A 0 False
460 2301 4ChoiceRANK2
2.0 B 0 False
460 2302 4ChoiceRANK3
2.0 D 0 False
460 2303 4ChoiceRANK4
2.0 C 0 False
461 2305 4ChoiceRANK1
3.0 B 0 False
461 2306 4ChoiceRANK2
3.0 A 0 False
461 2307 4ChoiceRANK3
3.0 D 0 False
461 2308 4ChoiceRANK4
3.0 C 0 False
462
462 2310 4ChoiceRANK1
2.0 B 0 False
462 2311 4ChoiceRANK2
2.0 D 0 False
462 2312 4ChoiceRANK3
2.0 A 0 False
462 2313 4ChoiceRANK4
2.0 C 0 False
463
463 2315 4ChoiceRANK1
2.0 B 0 False
463 2316 4ChoiceRANK2
2.0 A 0 False
463 2317 4ChoiceRANK3
2.0 D 0 False
463 2318 4ChoiceRANK4
2.0 C 0 False
464 2320 4ChoiceRANK1
3.0 D 0
464 2321 4ChoiceRANK2
3.0 B 0
464 2322 4ChoiceRANK3
3.0 A 0
464 2323 4ChoiceRANK4
```

3.0 C 0

```
465
465 2325 4ChoiceRANK1
3.0 C 0
465 2326 4ChoiceRANK2
3.0 A 0
465 2327 4ChoiceRANK3
3.0 D 0
465 2328 4ChoiceRANK4
3.0 B 0
466 2330 4ChoiceRANK1
2.0 D 0 False
466 2331 4ChoiceRANK2
2.0 A 0 False
466 2332 4ChoiceRANK3
2.0 B 0 False
466 2333 4ChoiceRANK4
2.0 C 0 False
467
467 2335 4ChoiceRANK1
3.0 A 0 False
467 2336 4ChoiceRANK2
3.0 D 0 False
467 2337 4ChoiceRANK3
3.0 B 0 False
467 2338 4ChoiceRANK4
3.0 C 0 False
468
468 2340 4ChoiceRANK1
1.0 B 0 False
468 2341 4ChoiceRANK2
1.0 D 0 False
468 2342 4ChoiceRANK3
1.0 A 0 False
468 2343 4ChoiceRANK4
1.0 C 0 False
469
469 2345 4ChoiceRANK1
2.0 D 0 False
469 2346 4ChoiceRANK2
2.0 B 0 False
469 2347 4ChoiceRANK3
2.0 A 0 False
469 2348 4ChoiceRANK4
2.0 C 0 False
470
470 2350 4ChoiceRANK1
3.0 B 0 False
470 2351 4ChoiceRANK2
3.0 A 0 False
470 2352 4ChoiceRANK3
3.0 D 0 False
470 2353 4ChoiceRANK4
3.0 C 0 False
471
471 2355 4ChoiceRANK1
3.0 D 0 False
471 2356 4ChoiceRANK2
3.0 C 0 False
471 2357 4ChoiceRANK3
3.0 B 0 False
471 2358 4ChoiceRANK4
3.0 A 0 False
472
472 2360 4ChoiceRANK1
1.0 B 0
472 2361 4ChoiceRANK2
472 2362 4ChoiceRANK3
472 2363 4ChoiceRANK4
error
473
473 2365 4ChoiceRANK1
2.0 A 0
473 2366 4ChoiceRANK2
2.0 C 0
473 2367 4ChoiceRANK3
2.0 B 0
473 2368 4ChoiceRANK4
error
474
474 2370 4ChoiceRANK1
4.0 C 0 False
474 2371 4ChoiceRANK2
4.0 D 0 False
474 2372 4ChoiceRANK3
4.0 B 0 False
474 2373 4ChoiceRANK4
4.0 A 0 False
475
475 2375 4ChoiceRANK1
2.0 D 0 False
```

```
475 2376 4ChoiceRANK2
2.0 A 0 False
475 2377 4ChoiceRANK3
2.0 C 0 False
475 2378 4ChoiceRANK4
2.0 B 0 False
476 2380 4ChoiceRANK1
1.0 C 0 False
476 2381 4ChoiceRANK2
1.0 D 0 False
476 2382 4ChoiceRANK3
1.0 A 0 False
476 2383 4ChoiceRANK4
1.0 B 0 False
477
477 2385 4ChoiceRANK1
4.0 D 0 False
477 2386 4ChoiceRANK2
4.0 B 0 False
477 2387 4ChoiceRANK3
4.0 C 0 False
477 2388 4ChoiceRANK4
4.0 A 0 False
478 2390 4ChoiceRANK1
1.0 B 0 False
478 2391 4ChoiceRANK2
1.0 A 0 False
478 2392 4ChoiceRANK3
1.0 D 0 False
478 2393 4ChoiceRANK4
1.0 C 0 False
479
479 2395 4ChoiceRANK1
3.0 B 0 False
479 2396 4ChoiceRANK2
3.0 A 0 False
479 2397 4ChoiceRANK3
3.0 D 0 False
479 2398 4ChoiceRANK4
3.0 C 0 False
480
480 2400 4ChoiceRANK1
1.0 B 0 False
480 2401 4ChoiceRANK2
1.0 D 0 False
480 2402 4ChoiceRANK3
1.0 A 0 False
480 2403 4ChoiceRANK4
1.0 C 0 False
481
481 2405 4ChoiceRANK1
3.0 A 0 False
481 2406 4ChoiceRANK2
3.0 B 0 False
481 2407 4ChoiceRANK3
3.0 D 0 False
481 2408 4ChoiceRANK4
3.0 C 0 False
482
482 2410 4ChoiceRANK1
1.0 A 0 False
482 2411 4ChoiceRANK2
1.0 B 0 False
482 2412 4ChoiceRANK3
1.0 D 0 False
482 2413 4ChoiceRANK4
1.0 C 0 False
483 2415 4ChoiceRANK1
2.0 A 0
483 2416 4ChoiceRANK2
2.0 B 0
483 2417 4ChoiceRANK3
2.0 C 0
483 2418 4ChoiceRANK4
error
484
484 2420 4ChoiceRANK1
3.0 B 0 False
484 2421 4ChoiceRANK2
3.0 A 0 False
484 2422 4ChoiceRANK3
3.0 D 0 False
484 2423 4ChoiceRANK4
3.0 C 0 False
485
485 2425 4ChoiceRANK1
3.0 D 0
485 2426 4ChoiceRANK2
3.0 B 0
```

485 2427 4ChoiceRANK3

```
3.0 A 0
485 2428 4ChoiceRANK4
3.0 C 0
486
486 2430 4ChoiceRANK1
4.0 A 0 False
486 2431 4ChoiceRANK2
4.0 B 0 False
486 2432 4ChoiceRANK3
4.0 C 0 False
486 2433 4ChoiceRANK4
4.0 D 0 False
487
487 2435 4ChoiceRANK1
2.0 D 0 False
487 2436 4ChoiceRANK2
2.0 A 0 False
487 2437 4ChoiceRANK3
2.0 B 0 False
487 2438 4ChoiceRANK4
2.0 C 0 False
488
488 2440 4ChoiceRANK1
2.0 A 0 False
488 2441 4ChoiceRANK2
2.0 C 0 False
488 2442 4ChoiceRANK3
2.0 B 0 False
488 2443 4ChoiceRANK4
2.0 D 0 False
489
489 2445 4ChoiceRANK1
3.0 D 0
489 2446 4ChoiceRANK2
3.0 B 0
489 2447 4ChoiceRANK3
3.0 A 0
489 2448 4ChoiceRANK4
3.0 C 0
490
490 2450 4ChoiceRANK1
1.0 A 0 False
490 2451 4ChoiceRANK2
1.0 B 0 False
490 2452 4ChoiceRANK3
1.0 D 0 False
490 2453 4ChoiceRANK4
1.0 C 0 False
491
491 2455 4ChoiceRANK1
2.0 A 0 False
491 2456 4ChoiceRANK2
2.0 C 0 False
491 2457 4ChoiceRANK3
2.0 D 0 False
491 2458 4ChoiceRANK4
2.0 B 0 False
492 2460 4ChoiceRANK1
3.0 A 0 False
492 2461 4ChoiceRANK2
3.0 B 0 False
492 2462 4ChoiceRANK3
3.0 D 0 False
492 2463 4ChoiceRANK4
3.0 C 0 False
493
493 2465 4ChoiceRANK1
3.0 D 0 False
493 2466 4ChoiceRANK2
3.0 C 0 False
493 2467 4ChoiceRANK3
3.0 B 0 False
493 2468 4ChoiceRANK4
3.0 A 0 False
494
494 2470 4ChoiceRANK1
2.0 C 0 False
494 2471 4ChoiceRANK2
2.0 D 0 False
494 2472 4ChoiceRANK3
2.0 B 0 False
494 2473 4ChoiceRANK4
2.0 A 0 False
495
495 2475 4ChoiceRANK1
2.0 A 0
495 2476 4ChoiceRANK2
2.0 C 0
495 2477 4ChoiceRANK3
2.0 B 0
495 2478 4ChoiceRANK4
```

error

```
496
496 2480 4ChoiceRANK1
3.0 B 0
496 2481 4ChoiceRANK2
3.0 A 0
496 2482 4ChoiceRANK3
3.0 C 0
496 2483 4ChoiceRANK4
3.0 D 0
497
497 2485 4ChoiceRANK1
3.0 C 0
497 2486 4ChoiceRANK2
3.0 D 0
497 2487 4ChoiceRANK3
3.0 B 0
497 2488 4ChoiceRANK4
3.0 A 0
498
498 2490 4ChoiceRANK1
2.0 D 0 False
498 2491 4ChoiceRANK2
2.0 C 0 False
498 2492 4ChoiceRANK3
2.0 B 0 False
498 2493 4ChoiceRANK4
2.0 A 0 False
499
499 2495 4ChoiceRANK1
2.0 A 0 False
499 2496 4ChoiceRANK2
2.0 B 0 False
499 2497 4ChoiceRANK3
2.0 C 0 False
499 2498 4ChoiceRANK4
2.0 D 0 False
500
500 2500 4ChoiceRANK1
3.0 D 0 False
500 2501 4ChoiceRANK2
3.0 B 0 False
500 2502 4ChoiceRANK3
3.0 C 0 False
500 2503 4ChoiceRANK4
3.0 A 0 False
501
501 2505 4ChoiceRANK1
2.0 A 0
501 2506 4ChoiceRANK2
2.0 C 0
501 2507 4ChoiceRANK3
2.0 B 0
501 2508 4ChoiceRANK4
502
502 2510 4ChoiceRANK1
2.0 D 0 False
502 2511 4ChoiceRANK2
2.0 C 0 False
502 2512 4ChoiceRANK3
2.0 B 0 False
502 2513 4ChoiceRANK4
2.0 A 0 False
503
503 2515 4ChoiceRANK1
2.0 D 0 False
503 2516 4ChoiceRANK2
2.0 B 0 False
503 2517 4ChoiceRANK3
2.0 C 0 False
503 2518 4ChoiceRANK4
2.0 A 0 False
504
504 2520 4ChoiceRANK1
2.0 A 0 False
504 2521 4ChoiceRANK2
2.0 B 0 False
504 2522 4ChoiceRANK3
2.0 C 0 False
504 2523 4ChoiceRANK4
2.0 D 0 False
505
505 2525 4ChoiceRANK1
3.0 B 0 False
505 2526 4ChoiceRANK2
3.0 C 0 False
505 2527 4ChoiceRANK3
3.0 A 0 False
505 2528 4ChoiceRANK4
3.0 D 0 False
506
506 2530 4ChoiceRANK1
```

3.0 D 0 False

506 2531 4ChoiceRANK2
3.0 A 0 False
506 2532 4ChoiceRANK3
3.0 B 0 False
506 2533 4ChoiceRANK4
3.0 C 0 False
0 out of 2028

```
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df new.at[i,'var.'+cols] = df1 to copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
```

df\_new.at[i,'var.'+cols] = df1\_to\_copy.at[0,cols]

```
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW 9\AppData\Local\Temp\ipykernel 27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df new.at[i,'var.'+cols] = df1 to copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
```

C:\Users\THW\_9\AppData\Local\Temp\ipykernel\_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i

```
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance.  Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
```

C:\Users\THW\_9\AppData\Local\Temp\ipykernel\_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi

```
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW 9\AppData\Local\Temp\ipykernel 27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
```

ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`

```
df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
 df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
```

```
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW 9\AppData\Local\Temp\ipykernel 27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
  df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\1591171231.py:118: PerformanceWarning: DataFrame is highly fragmented. This i
s usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once usi
ng pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
df_new.at[i,'var.'+cols] = df1_to_copy.at[0,cols]
```

```
1 out of 2028
2 out of 2028
3 out of 2028
4 out of 2028
5 out of 2028
6 out of 2028
7 out of 2028
8 out of 2028
9 out of 2028
10 out of 2028
11 out of 2028
12 out of 2028
13 out of 2028
14 out of 2028
15 out of 2028
16 out of 2028
17 out of 2028
18 out of 2028
19 out of 2028
20 out of 2028
21 out of 2028
22 out of 2028
23 out of 2028
24 out of 2028
25 out of 2028
26 out of 2028
27 out of 2028
28 out of 2028
29 out of 2028
30 out of 2028
31 out of 2028
32 out of 2028
33 out of 2028
34 out of 2028
35 out of 2028
36 out of 2028
37 out of 2028
38 out of 2028
39 out of 2028
40 out of 2028
41 out of 2028
42 out of 2028
43 out of 2028
44 out of 2028
45 out of 2028
46 out of 2028
47 out of 2028
48 out of 2028
49 out of 2028
50 out of 2028
51 out of 2028
52 out of 2028
53 out of 2028
54 out of 2028
55 out of 2028
56 out of 2028
57 out of 2028
58 out of 2028
59 out of 2028
60 out of 2028
61 out of 2028
62 out of 2028
63 out of 2028
64 out of 2028
65 out of 2028
66 out of 2028
67 out of 2028
68 out of 2028
69 out of 2028
70 out of 2028
71 out of 2028
72 out of 2028
73 out of 2028
74 out of 2028
75 out of 2028
76 out of 2028
77 out of 2028
78 out of 2028
79 out of 2028
80 out of 2028
81 out of 2028
82 out of 2028
83 out of 2028
84 out of 2028
85 out of 2028
86 out of 2028
87 out of 2028
88 out of 2028
89 out of 2028
90 out of 2028
91 out of 2028
```

92 out of 2028 93 out of 2028

```
94 out of 2028
95 out of 2028
96 out of 2028
97 out of 2028
98 out of 2028
99 out of 2028
100 out of 2028
101 out of 2028
102 out of 2028
103 out of 2028
104 out of 2028
105 out of 2028
106 out of 2028
107 out of 2028
108 out of 2028
109 out of 2028
110 out of 2028
111 out of 2028
112 out of 2028
113 out of 2028
114 out of 2028
115 out of 2028
116 out of 2028
117 out of 2028
118 out of 2028
119 out of 2028
120 out of 2028
121 out of 2028
122 out of 2028
123 out of 2028
124 out of 2028
125 out of 2028
126 out of 2028
127 out of 2028
128 out of 2028
129 out of 2028
130 out of 2028
131 out of 2028
132 out of 2028
133 out of 2028
134 out of 2028
135 out of 2028
136 out of 2028
137 out of 2028
138 out of 2028
139 out of 2028
140 out of 2028
141 out of 2028
142 out of 2028
143 out of 2028
144 out of 2028
145 out of 2028
146 out of 2028
147 out of 2028
148 out of 2028
149 out of 2028
150 out of 2028
151 out of 2028
152 out of 2028
153 out of 2028
154 out of 2028
155 out of 2028
156 out of 2028
157 out of 2028
158 out of 2028
159 out of 2028
160 out of 2028
161 out of 2028
162 out of 2028
163 out of 2028
164 out of 2028
165 out of 2028
166 out of 2028
167 out of 2028
168 out of 2028
169 out of 2028
170 out of 2028
171 out of 2028
172 out of 2028
173 out of 2028
174 out of 2028
175 out of 2028
176 out of 2028
177 out of 2028
178 out of 2028
179 out of 2028
180 out of 2028
181 out of 2028
182 out of 2028
183 out of 2028
184 out of 2028
185 out of 2028
```

```
187 out of 2028
188 out of 2028
189 out of 2028
190 out of 2028
191 out of 2028
192 out of 2028
193 out of 2028
194 out of 2028
195 out of 2028
196 out of 2028
197 out of 2028
198 out of 2028
199 out of 2028
200 out of 2028
201 out of 2028
202 out of 2028
203 out of 2028
204 out of 2028
205 out of 2028
206 out of 2028
207 out of 2028
208 out of 2028
209 out of 2028
210 out of 2028
211 out of 2028
212 out of 2028
213 out of 2028
214 out of 2028
215 out of 2028
216 out of 2028
217 out of 2028
218 out of 2028
219 out of 2028
220 out of 2028
221 out of 2028
222 out of 2028
223 out of 2028
224 out of 2028
225 out of 2028
226 out of 2028
227 out of 2028
228 out of 2028
229 out of 2028
230 out of 2028
231 out of 2028
232 out of 2028
233 out of 2028
234 out of 2028
235 out of 2028
236 out of 2028
237 out of 2028
238 out of 2028
239 out of 2028
240 out of 2028
241 out of 2028
242 out of 2028
243 out of 2028
244 out of 2028
245 out of 2028
246 out of 2028
247 out of 2028
248 out of 2028
249 out of 2028
250 out of 2028
251 out of 2028
252 out of 2028
253 out of 2028
254 out of 2028
255 out of 2028
256 out of 2028
257 out of 2028
258 out of 2028
259 out of 2028
260 out of 2028
261 out of 2028
262 out of 2028
263 out of 2028
264 out of 2028
265 out of 2028
266 out of 2028
267 out of 2028
268 out of 2028
269 out of 2028
270 out of 2028
271 out of 2028
272 out of 2028
273 out of 2028
274 out of 2028
275 out of 2028
276 out of 2028
277 out of 2028
278 out of 2028
```

```
280 out of 2028
281 out of 2028
282 out of 2028
283 out of 2028
284 out of 2028
285 out of 2028
286 out of 2028
287 out of 2028
288 out of 2028
289 out of 2028
290 out of 2028
291 out of 2028
292 out of 2028
293 out of 2028
294 out of 2028
295 out of 2028
296 out of 2028
297 out of 2028
298 out of 2028
299 out of 2028
300 out of 2028
301 out of 2028
302 out of 2028
303 out of 2028
304 out of 2028
305 out of 2028
306 out of 2028
307 out of 2028
308 out of 2028
309 out of 2028
310 out of 2028
311 out of 2028
312 out of 2028
313 out of 2028
314 out of 2028
315 out of 2028
316 out of 2028
317 out of 2028
318 out of 2028
319 out of 2028
320 out of 2028
321 out of 2028
322 out of 2028
323 out of 2028
324 out of 2028
325 out of 2028
326 out of 2028
327 out of 2028
328 out of 2028
329 out of 2028
330 out of 2028
331 out of 2028
332 out of 2028
333 out of 2028
334 out of 2028
335 out of 2028
336 out of 2028
337 out of 2028
338 out of 2028
339 out of 2028
340 out of 2028
341 out of 2028
342 out of 2028
343 out of 2028
344 out of 2028
345 out of 2028
346 out of 2028
347 out of 2028
348 out of 2028
349 out of 2028
350 out of 2028
351 out of 2028
352 out of 2028
353 out of 2028
354 out of 2028
355 out of 2028
356 out of 2028
357 out of 2028
358 out of 2028
359 out of 2028
360 out of 2028
361 out of 2028
362 out of 2028
363 out of 2028
364 out of 2028
365 out of 2028
366 out of 2028
367 out of 2028
368 out of 2028
369 out of 2028
370 out of 2028
371 out of 2028
```

```
373 out of 2028
374 out of 2028
375 out of 2028
376 out of 2028
377 out of 2028
378 out of 2028
379 out of 2028
380 out of 2028
381 out of 2028
382 out of 2028
383 out of 2028
384 out of 2028
385 out of 2028
386 out of 2028
387 out of 2028
388 out of 2028
389 out of 2028
390 out of 2028
391 out of 2028
392 out of 2028
393 out of 2028
394 out of 2028
395 out of 2028
396 out of 2028
397 out of 2028
398 out of 2028
399 out of 2028
400 out of 2028
401 out of 2028
402 out of 2028
403 out of 2028
404 out of 2028
405 out of 2028
406 out of 2028
407 out of 2028
408 out of 2028
409 out of 2028
410 out of 2028
411 out of 2028
412 out of 2028
413 out of 2028
414 out of 2028
415 out of 2028
416 out of 2028
417 out of 2028
418 out of 2028
419 out of 2028
420 out of 2028
421 out of 2028
422 out of 2028
423 out of 2028
424 out of 2028
425 out of 2028
426 out of 2028
427 out of 2028
428 out of 2028
429 out of 2028
430 out of 2028
431 out of 2028
432 out of 2028
433 out of 2028
434 out of 2028
435 out of 2028
436 out of 2028
437 out of 2028
438 out of 2028
439 out of 2028
440 out of 2028
441 out of 2028
442 out of 2028
443 out of 2028
444 out of 2028
445 out of 2028
446 out of 2028
447 out of 2028
448 out of 2028
449 out of 2028
450 out of 2028
451 out of 2028
452 out of 2028
453 out of 2028
454 out of 2028
455 out of 2028
456 out of 2028
457 out of 2028
458 out of 2028
459 out of 2028
460 out of 2028
461 out of 2028
462 out of 2028
463 out of 2028
464 out of 2028
```

```
466 out of 2028
467 out of 2028
468 out of 2028
469 out of 2028
470 out of 2028
471 out of 2028
472 out of 2028
473 out of 2028
474 out of 2028
475 out of 2028
476 out of 2028
477 out of 2028
478 out of 2028
479 out of 2028
480 out of 2028
481 out of 2028
482 out of 2028
483 out of 2028
484 out of 2028
485 out of 2028
486 out of 2028
487 out of 2028
488 out of 2028
489 out of 2028
490 out of 2028
491 out of 2028
492 out of 2028
493 out of 2028
494 out of 2028
495 out of 2028
496 out of 2028
497 out of 2028
498 out of 2028
499 out of 2028
500 out of 2028
501 out of 2028
502 out of 2028
503 out of 2028
504 out of 2028
505 out of 2028
506 out of 2028
507 out of 2028
508 out of 2028
509 out of 2028
510 out of 2028
511 out of 2028
512 out of 2028
513 out of 2028
514 out of 2028
515 out of 2028
516 out of 2028
517 out of 2028
518 out of 2028
519 out of 2028
520 out of 2028
521 out of 2028
522 out of 2028
523 out of 2028
524 out of 2028
525 out of 2028
526 out of 2028
527 out of 2028
528 out of 2028
529 out of 2028
530 out of 2028
531 out of 2028
532 out of 2028
533 out of 2028
534 out of 2028
535 out of 2028
536 out of 2028
537 out of 2028
538 out of 2028
539 out of 2028
540 out of 2028
541 out of 2028
542 out of 2028
543 out of 2028
544 out of 2028
545 out of 2028
546 out of 2028
547 out of 2028
548 out of 2028
549 out of 2028
550 out of 2028
551 out of 2028
552 out of 2028
553 out of 2028
554 out of 2028
555 out of 2028
556 out of 2028
557 out of 2028
```

```
559 out of 2028
560 out of 2028
561 out of 2028
562 out of 2028
563 out of 2028
564 out of 2028
565 out of 2028
566 out of 2028
567 out of 2028
568 out of 2028
569 out of 2028
570 out of 2028
571 out of 2028
572 out of 2028
573 out of 2028
574 out of 2028
575 out of 2028
576 out of 2028
577 out of 2028
578 out of 2028
579 out of 2028
580 out of 2028
581 out of 2028
582 out of 2028
583 out of 2028
584 out of 2028
585 out of 2028
586 out of 2028
587 out of 2028
588 out of 2028
589 out of 2028
590 out of 2028
591 out of 2028
592 out of 2028
593 out of 2028
594 out of 2028
595 out of 2028
596 out of 2028
597 out of 2028
598 out of 2028
599 out of 2028
600 out of 2028
601 out of 2028
602 out of 2028
603 out of 2028
604 out of 2028
605 out of 2028
606 out of 2028
607 out of 2028
608 out of 2028
609 out of 2028
610 out of 2028
611 out of 2028
612 out of 2028
613 out of 2028
614 out of 2028
615 out of 2028
616 out of 2028
617 out of 2028
618 out of 2028
619 out of 2028
620 out of 2028
621 out of 2028
622 out of 2028
623 out of 2028
624 out of 2028
625 out of 2028
626 out of 2028
627 out of 2028
628 out of 2028
629 out of 2028
630 out of 2028
631 out of 2028
632 out of 2028
633 out of 2028
634 out of 2028
635 out of 2028
636 out of 2028
637 out of 2028
638 out of 2028
639 out of 2028
640 out of 2028
641 out of 2028
642 out of 2028
643 out of 2028
644 out of 2028
645 out of 2028
646 out of 2028
647 out of 2028
648 out of 2028
649 out of 2028
650 out of 2028
```

```
652 out of 2028
653 out of 2028
654 out of 2028
655 out of 2028
656 out of 2028
657 out of 2028
658 out of 2028
659 out of 2028
660 out of 2028
661 out of 2028
662 out of 2028
663 out of 2028
664 out of 2028
665 out of 2028
666 out of 2028
667 out of 2028
668 out of 2028
669 out of 2028
670 out of 2028
671 out of 2028
672 out of 2028
673 out of 2028
674 out of 2028
675 out of 2028
676 out of 2028
677 out of 2028
678 out of 2028
679 out of 2028
680 out of 2028
681 out of 2028
682 out of 2028
683 out of 2028
684 out of 2028
685 out of 2028
686 out of 2028
687 out of 2028
688 out of 2028
689 out of 2028
690 out of 2028
691 out of 2028
692 out of 2028
693 out of 2028
694 out of 2028
695 out of 2028
696 out of 2028
697 out of 2028
698 out of 2028
699 out of 2028
700 out of 2028
701 out of 2028
702 out of 2028
703 out of 2028
704 out of 2028
705 out of 2028
706 out of 2028
707 out of 2028
708 out of 2028
709 out of 2028
710 out of 2028
711 out of 2028
712 out of 2028
713 out of 2028
714 out of 2028
715 out of 2028
716 out of 2028
717 out of 2028
718 out of 2028
719 out of 2028
720 out of 2028
721 out of 2028
722 out of 2028
723 out of 2028
724 out of 2028
725 out of 2028
726 out of 2028
727 out of 2028
728 out of 2028
729 out of 2028
730 out of 2028
731 out of 2028
732 out of 2028
733 out of 2028
734 out of 2028
735 out of 2028
736 out of 2028
737 out of 2028
738 out of 2028
739 out of 2028
740 out of 2028
741 out of 2028
742 out of 2028
743 out of 2028
```

```
745 out of 2028
746 out of 2028
747 out of 2028
748 out of 2028
749 out of 2028
750 out of 2028
751 out of 2028
752 out of 2028
753 out of 2028
754 out of 2028
755 out of 2028
756 out of 2028
757 out of 2028
758 out of 2028
759 out of 2028
760 out of 2028
761 out of 2028
762 out of 2028
763 out of 2028
764 out of 2028
765 out of 2028
766 out of 2028
767 out of 2028
768 out of 2028
769 out of 2028
770 out of 2028
771 out of 2028
772 out of 2028
773 out of 2028
774 out of 2028
775 out of 2028
776 out of 2028
777 out of 2028
778 out of 2028
779 out of 2028
780 out of 2028
781 out of 2028
782 out of 2028
783 out of 2028
784 out of 2028
785 out of 2028
786 out of 2028
787 out of 2028
788 out of 2028
789 out of 2028
790 out of 2028
791 out of 2028
792 out of 2028
793 out of 2028
794 out of 2028
795 out of 2028
796 out of 2028
797 out of 2028
798 out of 2028
799 out of 2028
800 out of 2028
801 out of 2028
802 out of 2028
803 out of 2028
804 out of 2028
805 out of 2028
806 out of 2028
807 out of 2028
808 out of 2028
809 out of 2028
810 out of 2028
811 out of 2028
812 out of 2028
813 out of 2028
814 out of 2028
815 out of 2028
816 out of 2028
817 out of 2028
818 out of 2028
819 out of 2028
820 out of 2028
821 out of 2028
822 out of 2028
823 out of 2028
824 out of 2028
825 out of 2028
826 out of 2028
827 out of 2028
828 out of 2028
829 out of 2028
830 out of 2028
831 out of 2028
832 out of 2028
833 out of 2028
834 out of 2028
835 out of 2028
836 out of 2028
837 out of 2028
```

```
838 out of 2028
839 out of 2028
840 out of 2028
841 out of 2028
842 out of 2028
843 out of 2028
844 out of 2028
845 out of 2028
846 out of 2028
847 out of 2028
848 out of 2028
849 out of 2028
850 out of 2028
851 out of 2028
852 out of 2028
853 out of 2028
854 out of 2028
855 out of 2028
856 out of 2028
857 out of 2028
858 out of 2028
859 out of 2028
860 out of 2028
861 out of 2028
862 out of 2028
863 out of 2028
864 out of 2028
865 out of 2028
866 out of 2028
867 out of 2028
868 out of 2028
869 out of 2028
870 out of 2028
871 out of 2028
872 out of 2028
873 out of 2028
874 out of 2028
875 out of 2028
876 out of 2028
877 out of 2028
878 out of 2028
879 out of 2028
880 out of 2028
881 out of 2028
882 out of 2028
883 out of 2028
884 out of 2028
885 out of 2028
886 out of 2028
887 out of 2028
888 out of 2028
889 out of 2028
890 out of 2028
891 out of 2028
892 out of 2028
893 out of 2028
894 out of 2028
895 out of 2028
896 out of 2028
897 out of 2028
898 out of 2028
899 out of 2028
900 out of 2028
901 out of 2028
902 out of 2028
903 out of 2028
904 out of 2028
905 out of 2028
906 out of 2028
907 out of 2028
908 out of 2028
909 out of 2028
910 out of 2028
911 out of 2028
912 out of 2028
913 out of 2028
914 out of 2028
915 out of 2028
916 out of 2028
917 out of 2028
918 out of 2028
919 out of 2028
920 out of 2028
921 out of 2028
922 out of 2028
923 out of 2028
924 out of 2028
925 out of 2028
926 out of 2028
927 out of 2028
928 out of 2028
929 out of 2028
```

```
931 out of 2028
932 out of 2028
933 out of 2028
934 out of 2028
935 out of 2028
936 out of 2028
937 out of 2028
938 out of 2028
939 out of 2028
940 out of 2028
941 out of 2028
942 out of 2028
943 out of 2028
944 out of 2028
945 out of 2028
946 out of 2028
947 out of 2028
948 out of 2028
949 out of 2028
950 out of 2028
951 out of 2028
952 out of 2028
953 out of 2028
954 out of 2028
955 out of 2028
956 out of 2028
957 out of 2028
958 out of 2028
959 out of 2028
960 out of 2028
961 out of 2028
962 out of 2028
963 out of 2028
964 out of 2028
965 out of 2028
966 out of 2028
967 out of 2028
968 out of 2028
969 out of 2028
970 out of 2028
971 out of 2028
972 out of 2028
973 out of 2028
974 out of 2028
975 out of 2028
976 out of 2028
977 out of 2028
978 out of 2028
979 out of 2028
980 out of 2028
981 out of 2028
982 out of 2028
983 out of 2028
984 out of 2028
985 out of 2028
986 out of 2028
987 out of 2028
988 out of 2028
989 out of 2028
990 out of 2028
991 out of 2028
992 out of 2028
993 out of 2028
994 out of 2028
995 out of 2028
996 out of 2028
997 out of 2028
998 out of 2028
999 out of 2028
1000 out of 2028
1001 out of 2028
1002 out of 2028
1003 out of 2028
1004 out of 2028
1005 out of 2028
1006 out of 2028
1007 out of 2028
1008 out of 2028
1009 out of 2028
1010 out of 2028
1011 out of 2028
1012 out of 2028
1013 out of 2028
1014 out of 2028
1015 out of 2028
1016 out of 2028
1017 out of 2028
1018 out of 2028
1019 out of 2028
1020 out of 2028
1021 out of 2028
1022 out of 2028
1023 out of 2028
```

```
1024 out of 2028
1025 out of 2028
1026 out of 2028
1027 out of 2028
1028 out of 2028
1029 out of 2028
1030 out of 2028
1031 out of 2028
1032 out of 2028
1033 out of 2028
1034 out of 2028
1035 out of 2028
1036 out of 2028
1037 out of 2028
1038 out of 2028
1039 out of 2028
1040 out of 2028
1041 out of 2028
1042 out of 2028
1043 out of 2028
1044 out of 2028
1045 out of 2028
1046 out of 2028
1047 out of 2028
1048 out of 2028
1049 out of 2028
1050 out of 2028
1051 out of 2028
1052 out of 2028
1053 out of 2028
1054 out of 2028
1055 out of 2028
1056 out of 2028
1057 out of 2028
1058 out of 2028
1059 out of 2028
1060 out of 2028
1061 out of 2028
1062 out of 2028
1063 out of 2028
1064 out of 2028
1065 out of 2028
1066 out of 2028
1067 out of 2028
1068 out of 2028
1069 out of 2028
1070 out of 2028
1071 out of 2028
1072 out of 2028
1073 out of 2028
1074 out of 2028
1075 out of 2028
1076 out of 2028
1077 out of 2028
1078 out of 2028
1079 out of 2028
1080 out of 2028
1081 out of 2028
1082 out of 2028
1083 out of 2028
1084 out of 2028
1085 out of 2028
1086 out of 2028
1087 out of 2028
1088 out of 2028
1089 out of 2028
1090 out of 2028
1091 out of 2028
1092 out of 2028
1093 out of 2028
1094 out of 2028
1095 out of 2028
1096 out of 2028
1097 out of 2028
1098 out of 2028
1099 out of 2028
1100 out of 2028
1101 out of 2028
1102 out of 2028
1103 out of 2028
1104 out of 2028
1105 out of 2028
1106 out of 2028
1107 out of 2028
1108 out of 2028
1109 out of 2028
1110 out of 2028
1111 out of 2028
1112 out of 2028
1113 out of 2028
1114 out of 2028
1115 out of 2028
1116 out of 2028
```

```
1117 out of 2028
1118 out of 2028
1119 out of 2028
1120 out of 2028
1121 out of 2028
1122 out of 2028
1123 out of 2028
1124 out of 2028
1125 out of 2028
1126 out of 2028
1127 out of 2028
1128 out of 2028
1129 out of 2028
1130 out of 2028
1131 out of 2028
1132 out of 2028
1133 out of 2028
1134 out of 2028
1135 out of 2028
1136 out of 2028
1137 out of 2028
1138 out of 2028
1139 out of 2028
1140 out of 2028
1141 out of 2028
1142 out of 2028
1143 out of 2028
1144 out of 2028
1145 out of 2028
1146 out of 2028
1147 out of 2028
1148 out of 2028
1149 out of 2028
1150 out of 2028
1151 out of 2028
1152 out of 2028
1153 out of 2028
1154 out of 2028
1155 out of 2028
1156 out of 2028
1157 out of 2028
1158 out of 2028
1159 out of 2028
1160 out of 2028
1161 out of 2028
1162 out of 2028
1163 out of 2028
1164 out of 2028
1165 out of 2028
1166 out of 2028
1167 out of 2028
1168 out of 2028
1169 out of 2028
1170 out of 2028
1171 out of 2028
1172 out of 2028
1173 out of 2028
1174 out of 2028
1175 out of 2028
1176 out of 2028
1177 out of 2028
1178 out of 2028
1179 out of 2028
1180 out of 2028
1181 out of 2028
1182 out of 2028
1183 out of 2028
1184 out of 2028
1185 out of 2028
1186 out of 2028
1187 out of 2028
1188 out of 2028
1189 out of 2028
1190 out of 2028
1191 out of 2028
1192 out of 2028
1193 out of 2028
1194 out of 2028
1195 out of 2028
1196 out of 2028
1197 out of 2028
1198 out of 2028
1199 out of 2028
1200 out of 2028
1201 out of 2028
1202 out of 2028
1203 out of 2028
1204 out of 2028
1205 out of 2028
1206 out of 2028
1207 out of 2028
1208 out of 2028
```

```
1210 out of 2028
1211 out of 2028
1212 out of 2028
1213 out of 2028
1214 out of 2028
1215 out of 2028
1216 out of 2028
1217 out of 2028
1218 out of 2028
1219 out of 2028
1220 out of 2028
1221 out of 2028
1222 out of 2028
1223 out of 2028
1224 out of 2028
1225 out of 2028
1226 out of 2028
1227 out of 2028
1228 out of 2028
1229 out of 2028
1230 out of 2028
1231 out of 2028
1232 out of 2028
1233 out of 2028
1234 out of 2028
1235 out of 2028
1236 out of 2028
1237 out of 2028
1238 out of 2028
1239 out of 2028
1240 out of 2028
1241 out of 2028
1242 out of 2028
1243 out of 2028
1244 out of 2028
1245 out of 2028
1246 out of 2028
1247 out of 2028
1248 out of 2028
1249 out of 2028
1250 out of 2028
1251 out of 2028
1252 out of 2028
1253 out of 2028
1254 out of 2028
1255 out of 2028
1256 out of 2028
1257 out of 2028
1258 out of 2028
1259 out of 2028
1260 out of 2028
1261 out of 2028
1262 out of 2028
1263 out of 2028
1264 out of 2028
1265 out of 2028
1266 out of 2028
1267 out of 2028
1268 out of 2028
1269 out of 2028
1270 out of 2028
1271 out of 2028
1272 out of 2028
1273 out of 2028
1274 out of 2028
1275 out of 2028
1276 out of 2028
1277 out of 2028
1278 out of 2028
1279 out of 2028
1280 out of 2028
1281 out of 2028
1282 out of 2028
1283 out of 2028
1284 out of 2028
1285 out of 2028
1286 out of 2028
1287 out of 2028
1288 out of 2028
1289 out of 2028
1290 out of 2028
1291 out of 2028
1292 out of 2028
1293 out of 2028
1294 out of 2028
1295 out of 2028
1296 out of 2028
1297 out of 2028
1298 out of 2028
1299 out of 2028
1300 out of 2028
1301 out of 2028
1302 out of 2028
```

```
1303 out of 2028
1304 out of 2028
1305 out of 2028
1306 out of 2028
1307 out of 2028
1308 out of 2028
1309 out of 2028
1310 out of 2028
1311 out of 2028
1312 out of 2028
1313 out of 2028
1314 out of 2028
1315 out of 2028
1316 out of 2028
1317 out of 2028
1318 out of 2028
1319 out of 2028
1320 out of 2028
1321 out of 2028
1322 out of 2028
1323 out of 2028
1324 out of 2028
1325 out of 2028
1326 out of 2028
1327 out of 2028
1328 out of 2028
1329 out of 2028
1330 out of 2028
1331 out of 2028
1332 out of 2028
1333 out of 2028
1334 out of 2028
1335 out of 2028
1336 out of 2028
1337 out of 2028
1338 out of 2028
1339 out of 2028
1340 out of 2028
1341 out of 2028
1342 out of 2028
1343 out of 2028
1344 out of 2028
1345 out of 2028
1346 out of 2028
1347 out of 2028
1348 out of 2028
1349 out of 2028
1350 out of 2028
1351 out of 2028
1352 out of 2028
1353 out of 2028
1354 out of 2028
1355 out of 2028
1356 out of 2028
1357 out of 2028
1358 out of 2028
1359 out of 2028
1360 out of 2028
1361 out of 2028
1362 out of 2028
1363 out of 2028
1364 out of 2028
1365 out of 2028
1366 out of 2028
1367 out of 2028
1368 out of 2028
1369 out of 2028
1370 out of 2028
1371 out of 2028
1372 out of 2028
1373 out of 2028
1374 out of 2028
1375 out of 2028
1376 out of 2028
1377 out of 2028
1378 out of 2028
1379 out of 2028
1380 out of 2028
1381 out of 2028
1382 out of 2028
1383 out of 2028
1384 out of 2028
1385 out of 2028
1386 out of 2028
1387 out of 2028
1388 out of 2028
1389 out of 2028
1390 out of 2028
1391 out of 2028
1392 out of 2028
1393 out of 2028
1394 out of 2028
```

```
1396 out of 2028
1397 out of 2028
1398 out of 2028
1399 out of 2028
1400 out of 2028
1401 out of 2028
1402 out of 2028
1403 out of 2028
1404 out of 2028
1405 out of 2028
1406 out of 2028
1407 out of 2028
1408 out of 2028
1409 out of 2028
1410 out of 2028
1411 out of 2028
1412 out of 2028
1413 out of 2028
1414 out of 2028
1415 out of 2028
1416 out of 2028
1417 out of 2028
1418 out of 2028
1419 out of 2028
1420 out of 2028
1421 out of 2028
1422 out of 2028
1423 out of 2028
1424 out of 2028
1425 out of 2028
1426 out of 2028
1427 out of 2028
1428 out of 2028
1429 out of 2028
1430 out of 2028
1431 out of 2028
1432 out of 2028
1433 out of 2028
1434 out of 2028
1435 out of 2028
1436 out of 2028
1437 out of 2028
1438 out of 2028
1439 out of 2028
1440 out of 2028
1441 out of 2028
1442 out of 2028
1443 out of 2028
1444 out of 2028
1445 out of 2028
1446 out of 2028
1447 out of 2028
1448 out of 2028
1449 out of 2028
1450 out of 2028
1451 out of 2028
1452 out of 2028
1453 out of 2028
1454 out of 2028
1455 out of 2028
1456 out of 2028
1457 out of 2028
1458 out of 2028
1459 out of 2028
1460 out of 2028
1461 out of 2028
1462 out of 2028
1463 out of 2028
1464 out of 2028
1465 out of 2028
1466 out of 2028
1467 out of 2028
1468 out of 2028
1469 out of 2028
1470 out of 2028
1471 out of 2028
1472 out of 2028
1473 out of 2028
1474 out of 2028
1475 out of 2028
1476 out of 2028
1477 out of 2028
1478 out of 2028
1479 out of 2028
1480 out of 2028
1481 out of 2028
1482 out of 2028
1483 out of 2028
1484 out of 2028
1485 out of 2028
1486 out of 2028
1487 out of 2028
```

```
1489 out of 2028
1490 out of 2028
1491 out of 2028
1492 out of 2028
1493 out of 2028
1494 out of 2028
1495 out of 2028
1496 out of 2028
1497 out of 2028
1498 out of 2028
1499 out of 2028
1500 out of 2028
1501 out of 2028
1502 out of 2028
1503 out of 2028
1504 out of 2028
1505 out of 2028
1506 out of 2028
1507 out of 2028
1508 out of 2028
1509 out of 2028
1510 out of 2028
1511 out of 2028
1512 out of 2028
1513 out of 2028
1514 out of 2028
1515 out of 2028
1516 out of 2028
1517 out of 2028
1518 out of 2028
1519 out of 2028
1520 out of 2028
1521 out of 2028
1522 out of 2028
1523 out of 2028
1524 out of 2028
1525 out of 2028
1526 out of 2028
1527 out of 2028
1528 out of 2028
1529 out of 2028
1530 out of 2028
1531 out of 2028
1532 out of 2028
1533 out of 2028
1534 out of 2028
1535 out of 2028
1536 out of 2028
1537 out of 2028
1538 out of 2028
1539 out of 2028
1540 out of 2028
1541 out of 2028
1542 out of 2028
1543 out of 2028
1544 out of 2028
1545 out of 2028
1546 out of 2028
1547 out of 2028
1548 out of 2028
1549 out of 2028
1550 out of 2028
1551 out of 2028
1552 out of 2028
1553 out of 2028
1554 out of 2028
1555 out of 2028
1556 out of 2028
1557 out of 2028
1558 out of 2028
1559 out of 2028
1560 out of 2028
1561 out of 2028
1562 out of 2028
1563 out of 2028
1564 out of 2028
1565 out of 2028
1566 out of 2028
1567 out of 2028
1568 out of 2028
1569 out of 2028
1570 out of 2028
1571 out of 2028
1572 out of 2028
1573 out of 2028
1574 out of 2028
1575 out of 2028
1576 out of 2028
1577 out of 2028
1578 out of 2028
1579 out of 2028
1580 out of 2028
```

```
1582 out of 2028
1583 out of 2028
1584 out of 2028
1585 out of 2028
1586 out of 2028
1587 out of 2028
1588 out of 2028
1589 out of 2028
1590 out of 2028
1591 out of 2028
1592 out of 2028
1593 out of 2028
1594 out of 2028
1595 out of 2028
1596 out of 2028
1597 out of 2028
1598 out of 2028
1599 out of 2028
1600 out of 2028
1601 out of 2028
1602 out of 2028
1603 out of 2028
1604 out of 2028
1605 out of 2028
1606 out of 2028
1607 out of 2028
1608 out of 2028
1609 out of 2028
1610 out of 2028
1611 out of 2028
1612 out of 2028
1613 out of 2028
1614 out of 2028
1615 out of 2028
1616 out of 2028
1617 out of 2028
1618 out of 2028
1619 out of 2028
1620 out of 2028
1621 out of 2028
1622 out of 2028
1623 out of 2028
1624 out of 2028
1625 out of 2028
1626 out of 2028
1627 out of 2028
1628 out of 2028
1629 out of 2028
1630 out of 2028
1631 out of 2028
1632 out of 2028
1633 out of 2028
1634 out of 2028
1635 out of 2028
1636 out of 2028
1637 out of 2028
1638 out of 2028
1639 out of 2028
1640 out of 2028
1641 out of 2028
1642 out of 2028
1643 out of 2028
1644 out of 2028
1645 out of 2028
1646 out of 2028
1647 out of 2028
1648 out of 2028
1649 out of 2028
1650 out of 2028
1651 out of 2028
1652 out of 2028
1653 out of 2028
1654 out of 2028
1655 out of 2028
1656 out of 2028
1657 out of 2028
1658 out of 2028
1659 out of 2028
1660 out of 2028
1661 out of 2028
1662 out of 2028
1663 out of 2028
1664 out of 2028
1665 out of 2028
1666 out of 2028
1667 out of 2028
1668 out of 2028
1669 out of 2028
1670 out of 2028
1671 out of 2028
1672 out of 2028
1673 out of 2028
```

```
1675 out of 2028
1676 out of 2028
1677 out of 2028
1678 out of 2028
1679 out of 2028
1680 out of 2028
1681 out of 2028
1682 out of 2028
1683 out of 2028
1684 out of 2028
1685 out of 2028
1686 out of 2028
1687 out of 2028
1688 out of 2028
1689 out of 2028
1690 out of 2028
1691 out of 2028
1692 out of 2028
1693 out of 2028
1694 out of 2028
1695 out of 2028
1696 out of 2028
1697 out of 2028
1698 out of 2028
1699 out of 2028
1700 out of 2028
1701 out of 2028
1702 out of 2028
1703 out of 2028
1704 out of 2028
1705 out of 2028
1706 out of 2028
1707 out of 2028
1708 out of 2028
1709 out of 2028
1710 out of 2028
1711 out of 2028
1712 out of 2028
1713 out of 2028
1714 out of 2028
1715 out of 2028
1716 out of 2028
1717 out of 2028
1718 out of 2028
1719 out of 2028
1720 out of 2028
1721 out of 2028
1722 out of 2028
1723 out of 2028
1724 out of 2028
1725 out of 2028
1726 out of 2028
1727 out of 2028
1728 out of 2028
1729 out of 2028
1730 out of 2028
1731 out of 2028
1732 out of 2028
1733 out of 2028
1734 out of 2028
1735 out of 2028
1736 out of 2028
1737 out of 2028
1738 out of 2028
1739 out of 2028
1740 out of 2028
1741 out of 2028
1742 out of 2028
1743 out of 2028
1744 out of 2028
1745 out of 2028
1746 out of 2028
1747 out of 2028
1748 out of 2028
1749 out of 2028
1750 out of 2028
1751 out of 2028
1752 out of 2028
1753 out of 2028
1754 out of 2028
1755 out of 2028
1756 out of 2028
1757 out of 2028
1758 out of 2028
1759 out of 2028
1760 out of 2028
1761 out of 2028
1762 out of 2028
1763 out of 2028
1764 out of 2028
1765 out of 2028
1766 out of 2028
```

```
1768 out of 2028
1769 out of 2028
1770 out of 2028
1771 out of 2028
1772 out of 2028
1773 out of 2028
1774 out of 2028
1775 out of 2028
1776 out of 2028
1777 out of 2028
1778 out of 2028
1779 out of 2028
1780 out of 2028
1781 out of 2028
1782 out of 2028
1783 out of 2028
1784 out of 2028
1785 out of 2028
1786 out of 2028
1787 out of 2028
1788 out of 2028
1789 out of 2028
1790 out of 2028
1791 out of 2028
1792 out of 2028
1793 out of 2028
1794 out of 2028
1795 out of 2028
1796 out of 2028
1797 out of 2028
1798 out of 2028
1799 out of 2028
1800 out of 2028
1801 out of 2028
1802 out of 2028
1803 out of 2028
1804 out of 2028
1805 out of 2028
1806 out of 2028
1807 out of 2028
1808 out of 2028
1809 out of 2028
1810 out of 2028
1811 out of 2028
1812 out of 2028
1813 out of 2028
1814 out of 2028
1815 out of 2028
1816 out of 2028
1817 out of 2028
1818 out of 2028
1819 out of 2028
1820 out of 2028
1821 out of 2028
1822 out of 2028
1823 out of 2028
1824 out of 2028
1825 out of 2028
1826 out of 2028
1827 out of 2028
1828 out of 2028
1829 out of 2028
1830 out of 2028
1831 out of 2028
1832 out of 2028
1833 out of 2028
1834 out of 2028
1835 out of 2028
1836 out of 2028
1837 out of 2028
1838 out of 2028
1839 out of 2028
1840 out of 2028
1841 out of 2028
1842 out of 2028
1843 out of 2028
1844 out of 2028
1845 out of 2028
1846 out of 2028
1847 out of 2028
1848 out of 2028
1849 out of 2028
1850 out of 2028
1851 out of 2028
1852 out of 2028
1853 out of 2028
1854 out of 2028
1855 out of 2028
1856 out of 2028
1857 out of 2028
1858 out of 2028
1859 out of 2028
```

```
1861 out of 2028
1862 out of 2028
1863 out of 2028
1864 out of 2028
1865 out of 2028
1866 out of 2028
1867 out of 2028
1868 out of 2028
1869 out of 2028
1870 out of 2028
1871 out of 2028
1872 out of 2028
1873 out of 2028
1874 out of 2028
1875 out of 2028
1876 out of 2028
1877 out of 2028
1878 out of 2028
1879 out of 2028
1880 out of 2028
1881 out of 2028
1882 out of 2028
1883 out of 2028
1884 out of 2028
1885 out of 2028
1886 out of 2028
1887 out of 2028
1888 out of 2028
1889 out of 2028
1890 out of 2028
1891 out of 2028
1892 out of 2028
1893 out of 2028
1894 out of 2028
1895 out of 2028
1896 out of 2028
1897 out of 2028
1898 out of 2028
1899 out of 2028
1900 out of 2028
1901 out of 2028
1902 out of 2028
1903 out of 2028
1904 out of 2028
1905 out of 2028
1906 out of 2028
1907 out of 2028
1908 out of 2028
1909 out of 2028
1910 out of 2028
1911 out of 2028
1912 out of 2028
1913 out of 2028
1914 out of 2028
1915 out of 2028
1916 out of 2028
1917 out of 2028
1918 out of 2028
1919 out of 2028
1920 out of 2028
1921 out of 2028
1922 out of 2028
1923 out of 2028
1924 out of 2028
1925 out of 2028
1926 out of 2028
1927 out of 2028
1928 out of 2028
1929 out of 2028
1930 out of 2028
1931 out of 2028
1932 out of 2028
1933 out of 2028
1934 out of 2028
1935 out of 2028
1936 out of 2028
1937 out of 2028
1938 out of 2028
1939 out of 2028
1940 out of 2028
1941 out of 2028
1942 out of 2028
1943 out of 2028
1944 out of 2028
1945 out of 2028
1946 out of 2028
1947 out of 2028
1948 out of 2028
1949 out of 2028
1950 out of 2028
1951 out of 2028
1952 out of 2028
```

```
1954 out of 2028
        1955 out of 2028
        1956 out of 2028
        1957 out of 2028
        1958 out of 2028
        1959 out of 2028
        1960 out of 2028
        1961 out of 2028
        1962 out of 2028
        1963 out of 2028
        1964 out of 2028
        1965 out of 2028
        1966 out of 2028
        1967 out of 2028
        1968 out of 2028
        1969 out of 2028
        1970 out of 2028
        1971 out of 2028
        1972 out of 2028
        1973 out of 2028
        1974 out of 2028
        1975 out of 2028
        1976 out of 2028
        1977 out of 2028
        1978 out of 2028
        1979 out of 2028
        1980 out of 2028
        1981 out of 2028
        1982 out of 2028
        1983 out of 2028
        1984 out of 2028
        1985 out of 2028
        1986 out of 2028
        1987 out of 2028
        1988 out of 2028
        1989 out of 2028
        1990 out of 2028
        1991 out of 2028
        1992 out of 2028
        1993 out of 2028
        1994 out of 2028
        1995 out of 2028
        1996 out of 2028
        1997 out of 2028
        1998 out of 2028
        1999 out of 2028
        2000 out of 2028
        2001 out of 2028
        2002 out of 2028
        2003 out of 2028
        2004 out of 2028
        2005 out of 2028
        2006 out of 2028
        2007 out of 2028
        2008 out of 2028
        2009 out of 2028
        2010 out of 2028
        2011 out of 2028
        2012 out of 2028
        2013 out of 2028
        2014 out of 2028
        2015 out of 2028
        2016 out of 2028
        2017 out of 2028
        2018 out of 2028
        2019 out of 2028
        2020 out of 2028
        2021 out of 2028
        2022 out of 2028
        2023 out of 2028
        2024 out of 2028
        2025 out of 2028
        2026 out of 2028
        2027 out of 2028
In [7]: pre_df = df_new
         #df_new.to_csv('P.Housing_Merged_2(ix)(forMixedLogitModel)(pre-policy).csv')
```

## **Mixed Logit Model**

```
In [6]: pre_df = df_new

# cleaning up the values
for i in range(0,len(pre_df)):
    if pre_df.at[i,'Location'] == 'Non-Mature':
        pre_df.at[i,'Location'] = 'Non_Mature'
    if pre_df.at[i,'4ChoiceRANK1'] != 1.0:
        pre_df.at[i,'4ChoiceRANK1'] = 0
```

```
#changing options ABCD to 1234
         for i in range(0,len(pre_df)):
             if pre_df.at[i,'Option'] in ['','A','B','C','D']:
    pre_df.at[i,'Option1'] = ['','A','B','C','D'].index(pre_df.at[i,'Option'])
                 pre_df.at[i,'Avail'] = 1
             else:
                 pre_df.at[i,'Option1'] = 0
                 pre_df.at[i,'Avail'] = 0
         pre_df = pd.get_dummies(pre_df, columns=['Location'])
         pre_df = pd.get_dummies(pre_df, columns=['Type'])
         C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\2610197272.py:14: PerformanceWarning: DataFrame is highly fragmented. This is
         usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using
         pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
           pre_df.at[i,'Option1'] = ['','A','B','C','D'].index(pre_df.at[i,'Option'])
         C:\Users\THW_9\AppData\Local\Temp\ipykernel_27260\2610197272.py:15: PerformanceWarning: DataFrame is highly fragmented. This is
         usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using
         pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
         pre_df.at[i,'Avail'] = 1
In [7]: #To clean up data, where avail = 0 and Options are empty
         for i in list(pre_df.loc[pre_df['Avail']==0].index):
             pre_df.at[i,'Option'] = list({'A','B','C','D'}.difference(
             set(pre_df.loc[pre_df['SN'] == pre_df.at[i,'SN']]['Option'].unique())
             ))[0]
In [8]: pre_df
```

Out[8]:

	index	SN	Rank	Option	Size	Rooms	МОР	Clawback	Pricing	4ChoiceRANK1	•••	var.Pm3	var.P%3	Option1	Avail	Location_Mature
0	0	1.0	1.0	В	1100.0	5.0	5.0	0%	\$500	1.0		-120.0	-0.315789	2.0	1.0	0
1	1	1.0	2.0	D	800.0	4.0	5.0	0%	\$500	0.0		-120.0	-0.315789	4.0	1.0	0
2	2	1.0	3.0	А	650.0	4.0	5.0	0%	\$500	0.0		-120.0	-0.315789	1.0	1.0	1
3	3	1.0	4.0	С	650.0	3.0	5.0	0%	\$500	0.0		-120.0	-0.315789	3.0	1.0	1
4	5	2.0	1.0	D	1100.0	5.0	5.0	0%	\$1,000	1.0		-50.0	-0.052632	4.0	1.0	0
•••																
2023	2528	506.0	4.0	D	800.0	4.0	5.0	0%	\$680	0.0		-30.0	-0.046154	4.0	1.0	0
2024	2530	507.0	1.0	D	800.0	4.0	5.0	0%	\$680	1.0		170.0	0.200000	4.0	1.0	0
2025	2531	507.0	2.0	А	800.0	4.0	5.0	0%	\$680	0.0		170.0	0.200000	1.0	1.0	1
2026	2532	507.0	3.0	В	1100.0	5.0	5.0	0%	\$680	0.0		170.0	0.200000	2.0	1.0	0
2027	2533	507.0	4.0	С	650.0	3.0	5.0	0%	\$680	0.0		170.0	0.200000	3.0	1.0	1

2028 rows × 246 columns

```
In [9]: varnames=['MOP',
                   'Location_Mature', 'Location_Non_Mature' ,
                   'Type_BTO', 'Type_EC', 'Type_Resale','Size', 'Rooms'
         model = MixedLogit()
         model.fit(X=pre_df[varnames], y=pre_df['4ChoiceRANK1'], varnames=varnames,
                   alts=pre_df['Option'],
                   ids=pre_df['SN'], avail=pre_df['Avail'],
                   #panels=pre_df["SN"],
                   randvars={'Size': 'n', 'Rooms': 'n', 'MOP': 'n', 'Location_Mature': 'n', 'Location_Non_Mature': 'n', 'Type_BTO': 'n',
                   n_draws=507,
                   optim_method='L-BFGS-B')
         model.summary()
```

```
ValueError
                                          Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27260\859624992.py in <module>
      4
      5 model = MixedLogit()
---> 6 model.fit(X=pre_df[varnames], y=pre_df['4ChoiceRANK1'], varnames=varnames,
      7
                  alts=pre_df['Option'],
      8
                  ids=pre_df['SN'], avail=pre_df['Avail'],
c:\Users\THW_9\anaconda3\lib\site-packages\xlogit\mixed_logit.py in fit(self, X, y, varnames, alts, ids, randvars, isvars, weigh
ts, avail, panels, base_alt, fit_intercept, init_coeff, maxiter, random_state, n_draws, halton, verbose, batch_size, halton_opt
s, tol_opts, robust, num_hess, scale_factor, optim_method, mnl_init, addit, skip_std_errs)
                    # Initialize coefficients using a multinomial logit model
    190
    191
                    mnl = MultinomialLogit()
--> 192
                    mnl.fit(X, y, varnames, alts, ids, isvars=isvars, weights=weights, addit=addit,
    193
                            avail=avail, base_alt=base_alt, fit_intercept=fit_intercept, skip_std_errs=True)
    194
                    init_coeff = np.concatenate((mnl.coeff_, np.repeat(.1, len(randvars))))
c:\Users\THW_9\anaconda3\lib\site-packages\xlogit\multinomial_logit.py in fit(self, X, y, varnames, alts, ids, isvars, weights,
avail, base_alt, fit_intercept, init_coeff, maxiter, random_state, tol_opts, verbose, robust, num_hess, scale_factor, addit, ski
p_std_errs)
    143
    144
                betas, X, y, weights, avail, Xnames, scale, addit = \
--> 145
                    self._setup_input_data(X, y, varnames, alts, ids,
    146
                                           isvars=isvars, weights=weights, avail=avail,
    147
                                           init_coeff=init_coeff,
c:\Users\THW_9\anaconda3\lib\site-packages\xlogit\multinomial_logit.py in _setup_input_data(self, X, y, varnames, alts, ids, isv
ars, weights, avail, base_alt, fit_intercept, init_coeff, random_state, verbose, predict_mode, scale_factor, addit)
    287
                    init_coeff=None, random_state=None, verbose=1, predict_mode=False,
    288
                    scale_factor=None, addit=None):
--> 289
                self._check_long_format_consistency(ids, alts)
    290
                y = self._format_choice_var(y, alts) if not predict_mode else None
                X, Xnames = self._setup_design_matrix(X)
    291
c:\Users\THW_9\anaconda3\lib\site-packages\xlogit\_choice_model.py in _check_long_format_consistency(self, ids, alts)
                expected_alts = np.tile(uq_alts, int(len(ids)/len(uq_alts)))
    174
    175
                if not np.array_equal(alts, expected_alts):
--> 176
                    raise ValueError('inconsistent alts values in long format')
    177
                   obs_by_id = np.unique(ids, return_counts=True)
                if not np.all(obs_by_id/len(uq_alts)): # Multiple of J
    178
ValueError: inconsistent alts values in long format
```

In [101... pre\_df.to\_csv('P.Housing\_Merged\_2(ix\_ii)(forMixedLogitModel)(pre-policy).csv')