

第2回 SASプログラミングコンテスト リバーシーチャンピオンシップ



SASユーザー総会 2026

開催概要

- 今年もリバーシーチャンピオンシップを開催します!!
- ルールは昨年と一緒ですが、制限時間を延長しています
- 公開されているプログラムの流用はNGです
機械的および視覚的に類似性を確認します
類似度の高い人には出場を辞退いただきます
- 昨年同様に配布資料を利用してプログラム開発ください
(提供プログラムは使いやすいように各々改変ください)
- 不明点は井上貴博(takahiro.inoue.y1@kyowakirin.com)まで
- 配布資料
 - ▣ プログラム開発用対戦プログラム: platform_v1.txt
 - ▣ 対戦用サンプルプログラム: tester_v1.txt
 - ▣ 自己紹介スライド(テンプレート): Self-intro.pptx

申し込み方法

- 2026年8月21日までに「発表登録」より、お願いします。

発表タイトル ※公式サイトに掲載します 必須	プログラムコンテスト出場
発表要旨 (100文字以内) ※公式サイトに掲載します 必須	プログラムコンテスト出場 12文字 / 最大100字
発表概略 (500文字以内) 必須	プログラムコンテスト出場 12文字 / 最大500字
想定聴講者 必須	初級者向け
連絡事項	プログラムコンテスト出場

申し込み

プログラム・参加条件

コンテスト参加・実装ルール

- ・ コンテスト実施時は、会場でご参加ください。
- ・ プログラムは マクロで作成します。マクロ名は世話人より指定します。例: INOUETA1(姓5文字+名2文字+連番)
- ・ 出場者のマクロを SASユーザー総会 公式Webサイト に公開することに同意してください。
- ・ SAS学習者の参考となるように、コメントを適宜追記し、分かりやすい構文を心掛けてください。

盤面データ・入出力仕様

- ・ 盤面データとして、SAS データセット[_info]を読み込み、選択したセルをグローバルマクロ変数[_xy]に x,y の形式で格納してください。例: [X=7, Y=3]: 7,3
- ・ 対局開始時に黒または白の担当が決定されます。どちらの担当でも動作するプログラムを作成してください。
- ・ 右図の場合、17手目 (No=17) で Turn = -1 (黒)の指し手となります。変数 Y, X1-X8 には、16手目までの盤面情報が格納されており、黒は -1、白は 1 で表現されます。17手目で指せるセルには 99 が入力されています。選択可能セル (99) から1つを選択し、_xy に設定してください。

命名・使用制限

- ・ マクロ内で、アンダースコア (_) から始まるデータセット名・マクロ変数名は作成しないでください。
- ・ 文字変数の入力値はASCII文字のみを使用してください。コメントには日本語等の使用は可能です。
- ・ 対戦プログラムでは[_info]以外にもデータセットが作成されますが、使用できません。コンテスト当日は、ライブ対戦用に別プログラムを使用するため、[_info]を除いたデータは名称・構造ともに異なります。

対局・実行条件

- ・ 動作不良は負けとなります。対局は、対戦者同士のマクロを 交互に実行します。細かな命名規則は定めませんので、データセット名の重複に注意してください。
- ・ SAS OnDemand 実行時に 1手6秒以内 (7秒未満) を目安としてください (合計時間ではありません) 。

```
Length No Y Turn X1 - X8 8. ;
No: ○手目
Turn: 指し手の色
      (黒 -1、白 1)
Y: 上から○マス(行)目
   (Y=1は1行目、Y=2は2行目...)
X1: 左から1マス(列)目
    (黒 -1、白 1、選択可能セル 99)
X2~X7: 左から2~7マス(列)目
X8: 左から8マス(列)目
```

提供データ(_info.sas7bdat)

No	Turn	Y	X1	X2	X3	X4	X5	X6	X7	X8
17	-1	1	0	0	0	0	0	0	0	0
17	-1	2	0	0	0	0	99	99	0	0
17	-1	3	0	0	-1	-1	1	1	99	0
17	-1	4	0	0	-1	-1	1	99	99	99
17	-1	5	0	-1	-1	1	1	1	1	1
17	-1	6	0	99	1	-1	-1	-1	99	0
17	-1	7	0	0	99	99	-1	1	99	0
17	-1	8	0	0	0	0	0	99	99	0

<イメージ>

Turn	Y	X1	X2	X3	X4	X5	X6	X7	X8
●	1								
●	2					99	99		
●	3			●	●	○	○	99	
●	4			●	●	○	99	99	99
●	5		●	●	○	○	○	○	○
●	6		99	○	●	●	●	99	
●	7			99	99	●	○	99	
●	8						99	99	

プログラム開発用対戦プログラム: platform_v1.txt

対戦プログラムは、
7, 10行目のマクロ名を変更して利用してください。

```

コード ログ 結果 出力データ
1 /* プログラム開発用対戦プログラム
2   Takahiro Inoue 2025/3/24 */
3
4 /*-----
5
6 /*対戦者1*/
7 %let _m1 = tester1;
8
9 /*対戦者2*/
10 %let _m2 = tester2;
11
12 /*-----
13
  
```

サンプルマクロは、ランダムで選ぶだけのマクロです。
動作確認もしくは対戦相手として、ご利用ください。

```

1 %macro tester1;
2   %global _xy;
3
4   proc transpose data = _info out = wk010;
5     var X1 - X8;
6     by No Turn Y;
7   run;
8
9   data wk020;
10    set wk010(where = (col1 = 99));
11    weight = ranuni(0);
12
13    proc sort;
14      by weight;
15    run;
16
17    data _null_;
18      set wk020(obs=1);
19      call symput('_xy', substr(_name_, 2, 1) || ',' || trim(put(y, best. -1)));
20    run;
21 %mend;
22
23 %macro tester2;
24   %tester1;
25 %mend;
  
```

コード ログ **結果** 出力データ

最終盤面

黒(-1): tester1 / 白(1): tester2

y	X1	X2	X3	X4	X5	X6	X7	X8
1	●	●	●	●	●	●	●	●
2	●	●	●	●	●	●	●	●
3	●	○	●	●	○	○	○	●
4	●	○	○	●	○	○	○	●
5	○	○	○	○	○	○	○	●
6	○	○	○	○	○	●	○	●
7	○	○	○	○	○	○	○	●
8	○	○	○	○	○	○	○	●

黒・白
度数

黒(-1): tester1 / 白(1): tester2

FREQ プロシジャ

COL1	度数	パーセント
●	28	43.75
○	36	56.25

一手ごと
考慮時間

黒(-1): tester1 / 白(1): tester2

MEANS プロシジャ

分析変数: time								
Turn	Obs 数	N	平均	最小値	下側四分位点	中央値	上側四分位点	最大値
-1	30	30	0	0	0	0	0	0
1	30	30	0.0333333	0	0	0	0	1.0000000

指し手の
履歴

黒(-1): tester1 / 白(1): tester2

No=1 Turn=-1 xy=6,4

y	X1	X2	X3	X4	X5	X6	X7	X8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	99	0	0	0

最大値が6秒以内になるように調整ください。

SAS Programming Contest Reversi Championship

This is an algorithmic programming competition focused on Reversi.

Participants develop decision-making logic using SAS programming. Each program determines where to place a coin based on the current board state, and competes automatically against other players' algorithms.

This competition focuses on strategy and algorithmic thinking. No user interface or graphics implementation is required.

Participants from all countries are welcome.



If you wish to participate, please contact Takahiro Inoue (takahiro.inoue.y1@kyowakirin.com) by 20Aug2026. For participants joining from outside Japan, on-site participation may be difficult; therefore, the match results and videos will be shared with participants at a later date.

Please note that reuse of publicly available programs is not permitted.

Program development must be based on the distributed materials listed below.

If you have any questions, please contact Takahiro Inoue.

Distributed materials

- Match program template for development: platform_v1.txt
- Sample program for matches: tester_v1.txt
- Self-introduction slide (template): Self-intro.pptx

Reversi – Basic Rules

- Reversi is played on an 8×8 board by two players: Black and White
- The game starts with four coins in the center
- Black moves first
- A move is valid if it sandwiches one or more opponent coins in a straight line (horizontal, vertical, or diagonal)
- All sandwiched coins are flipped to the current player's color
- If a player has no valid moves, they must pass
- The game ends when neither player can move
- The player with more coins on the board wins

Programming Requirement

General Requirements

- The program must be implemented as a SAS macro.
- The macro name will be specified by the organizers.
- By participating, you agree that your macro code may be published on the official SAS User Group website.
- Please write clear and well-commented code so that it is helpful for SAS learners.
Use ASCII characters only in code and input values (comments may include other languages)

Input / Output Specification

- The current board state is provided in a SAS dataset named `_info`.
- Your program must select one valid cell and output the move by setting the global macro variable `_xy` in the format: `x,y` [e.g., (X=7, Y=3): 7,3]

Player Color

- At the start of a match, your program is assigned Black or White.
- Your macro must work correctly for both colors. The assigned color does not change during a match.

Board Representation

- Variables `Y`, `X1`–`X8` represent the board state up to the previous move. Cell values: [-1 : Black] [1 : White] [99 : Valid move positions]
- Your program must choose one cell with value 99 as the next move.

Turn Information

- The variable `No` indicates the move number. The variable `Turn` indicates the current player [e.g., Black = -1, White = 1].

Naming Restrictions

- Do not create any dataset names or macro variables that start with an underscore (`_`).

Data Usage Restrictions

- Other datasets may exist during match execution, but only `_info` may be used.
- During live matches, datasets other than `_info` may differ in name and structure.

Execution and Stability

- If your macro fails to execute correctly, the match is considered a loss.
- Macros from both players are executed alternately during a match.
- Dataset name conflicts may cause unexpected behavior.

Time Limits

- As a guideline, execution time should be under 6 seconds per move when run on SAS OnDemand.
- The limit applies per move, not to the total match duration.

Input: `_info.sas7bdat`

No	Turn	Y	X1	X2	X3	X4	X5	X6	X7	X8
17	-1	1	0	0	0	0	0	0	0	0
17	-1	2	0	0	0	0	99	99	0	0
17	-1	3	0	0	-1	-1	1	1	99	0
17	-1	4	0	0	-1	-1	1	99	99	99
17	-1	5	0	-1	-1	1	1	1	1	1
17	-1	6	0	99	1	-1	-1	-1	99	0
17	-1	7	0	0	99	99	-1	1	99	0
17	-1	8	0	0	0	0	0	99	99	0