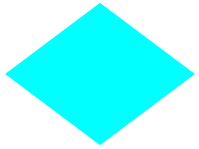
guide

end/start/continue

functions and process



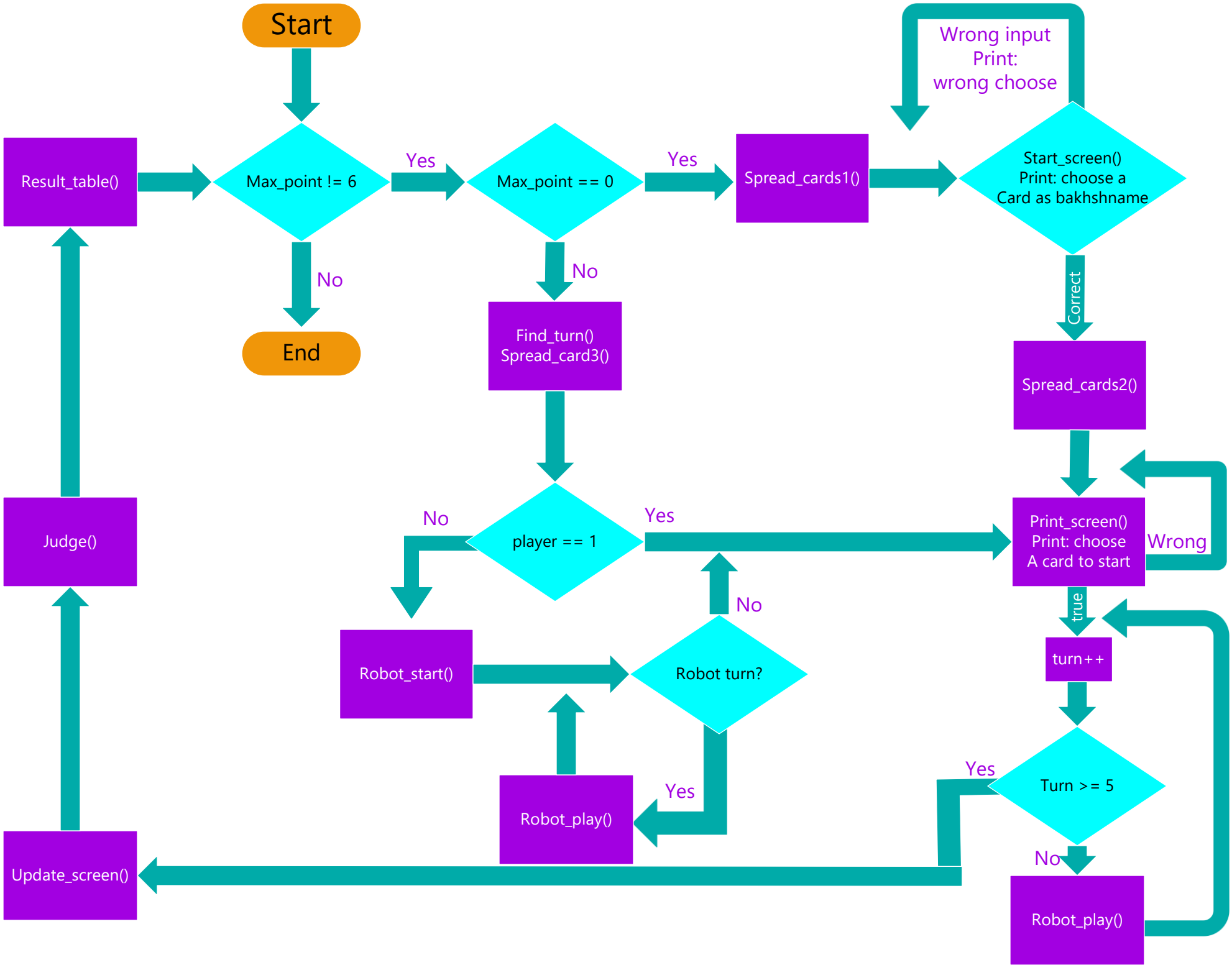
conditions

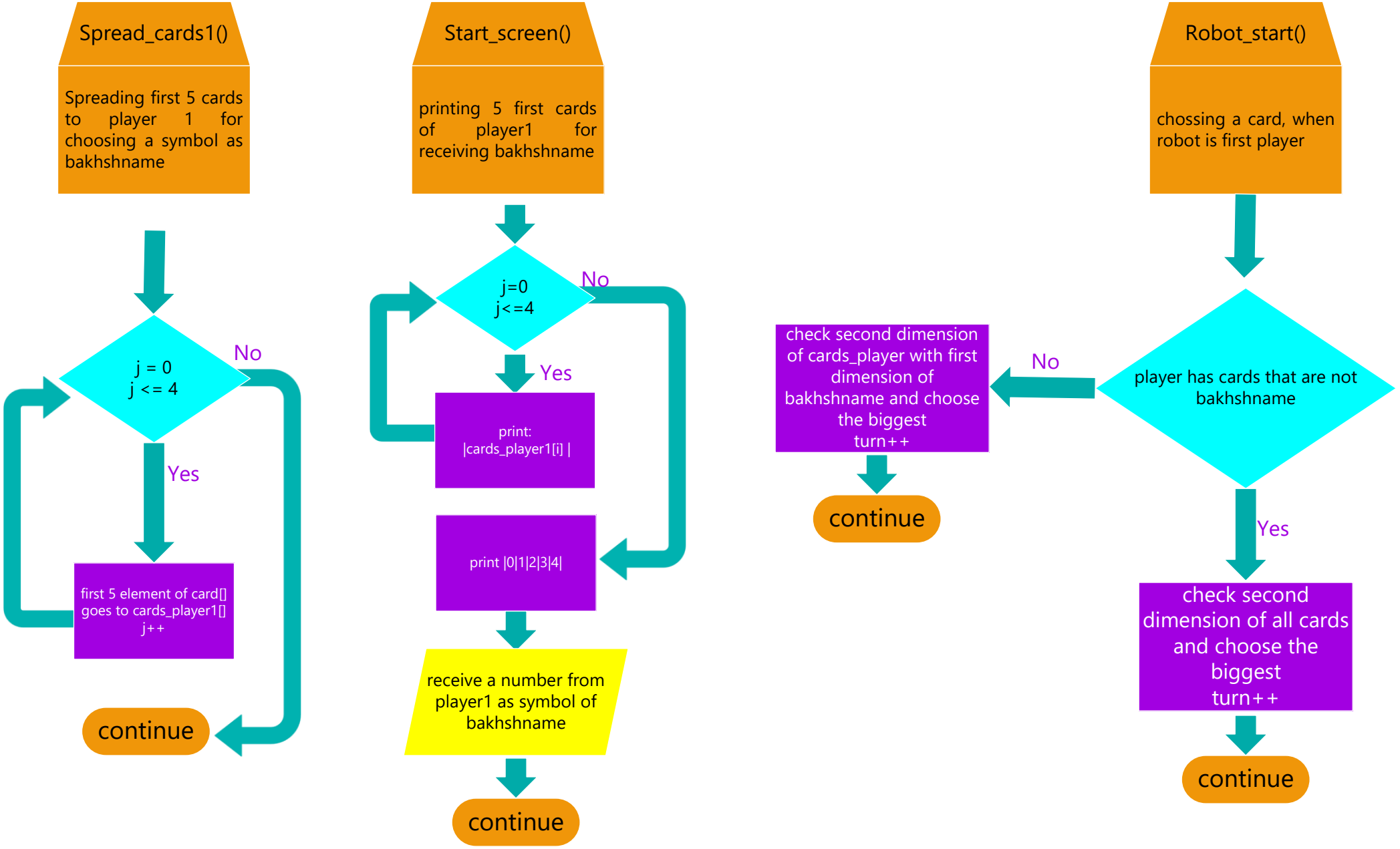
receiving data

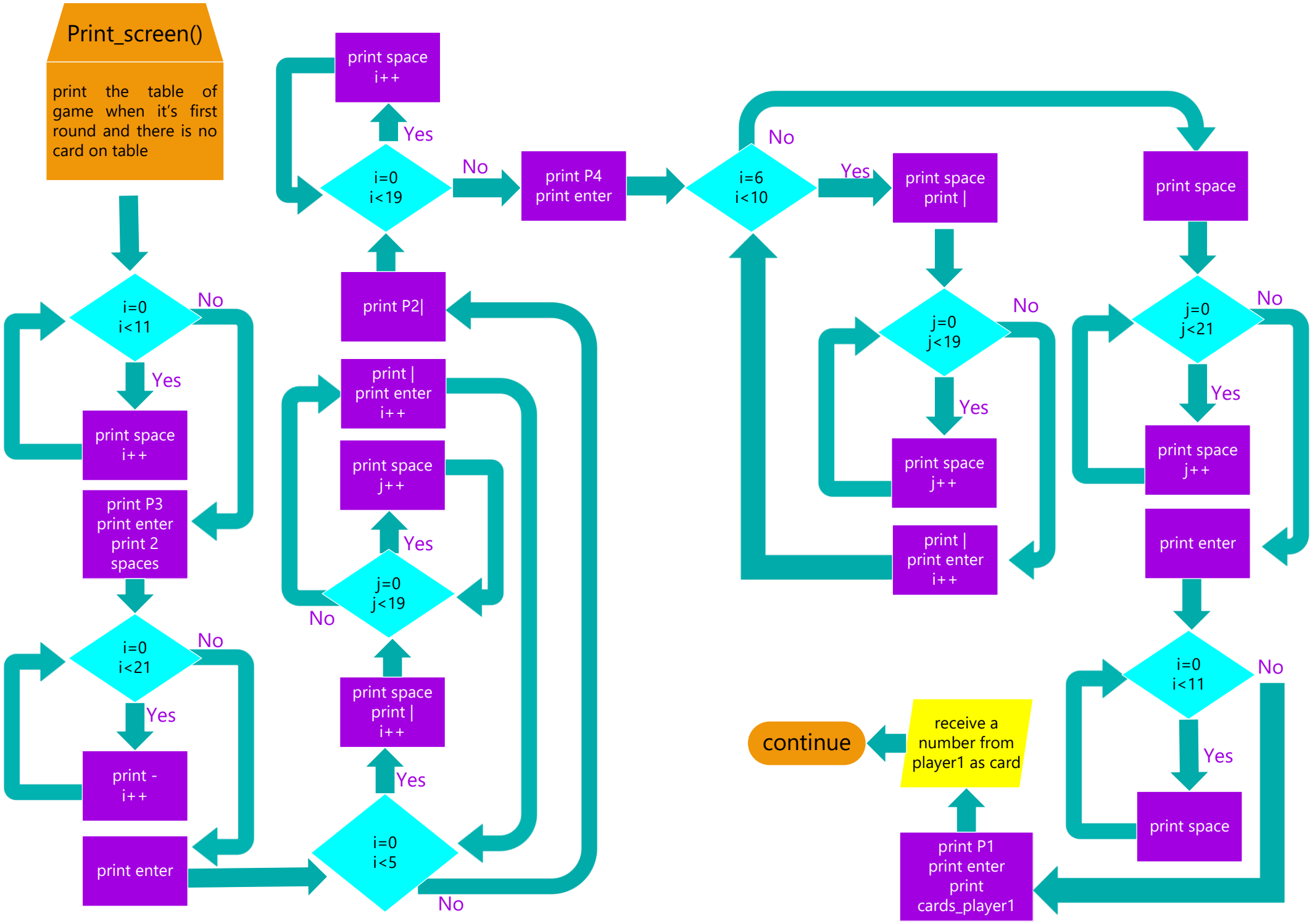
connections

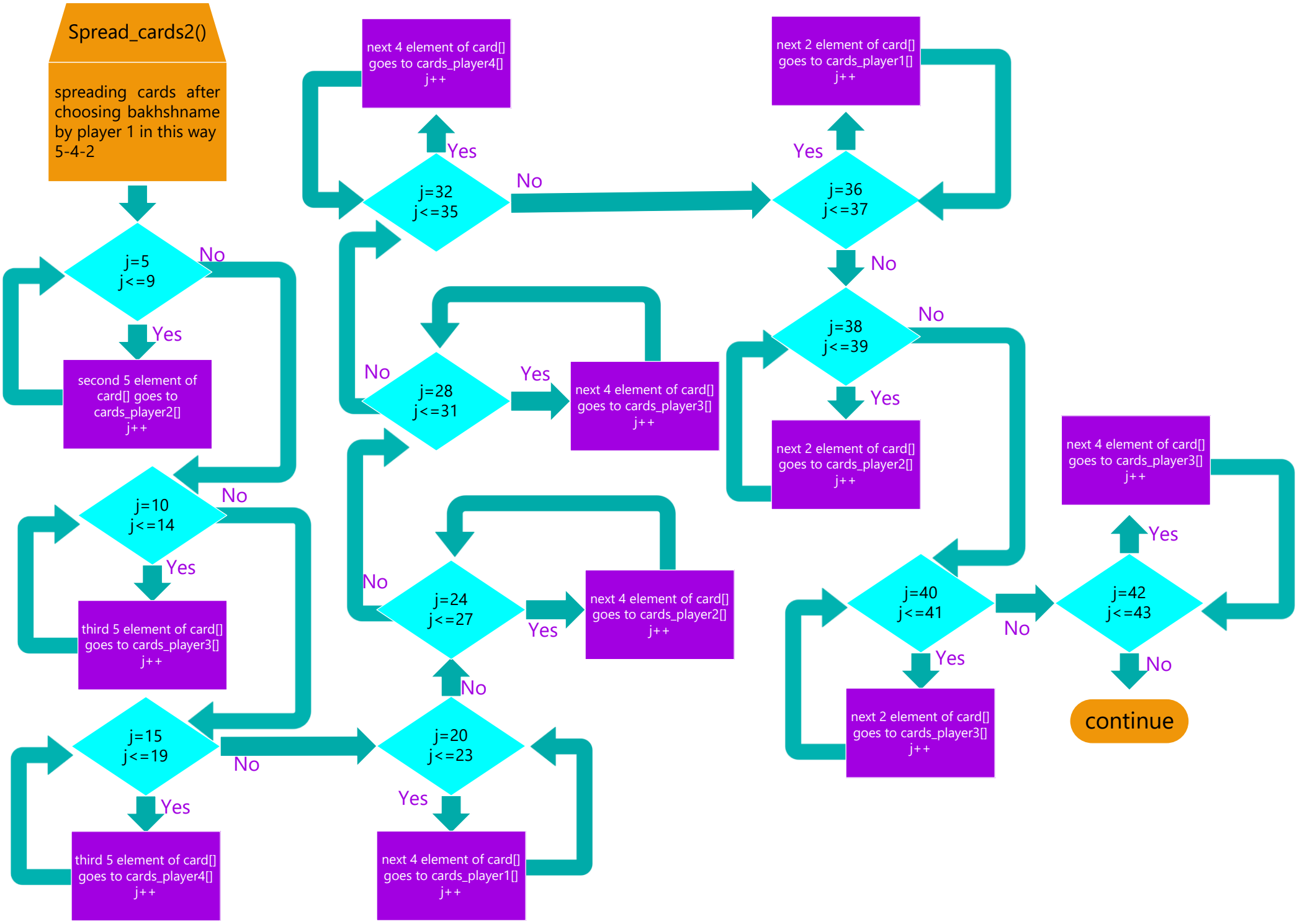
name & description
of functions

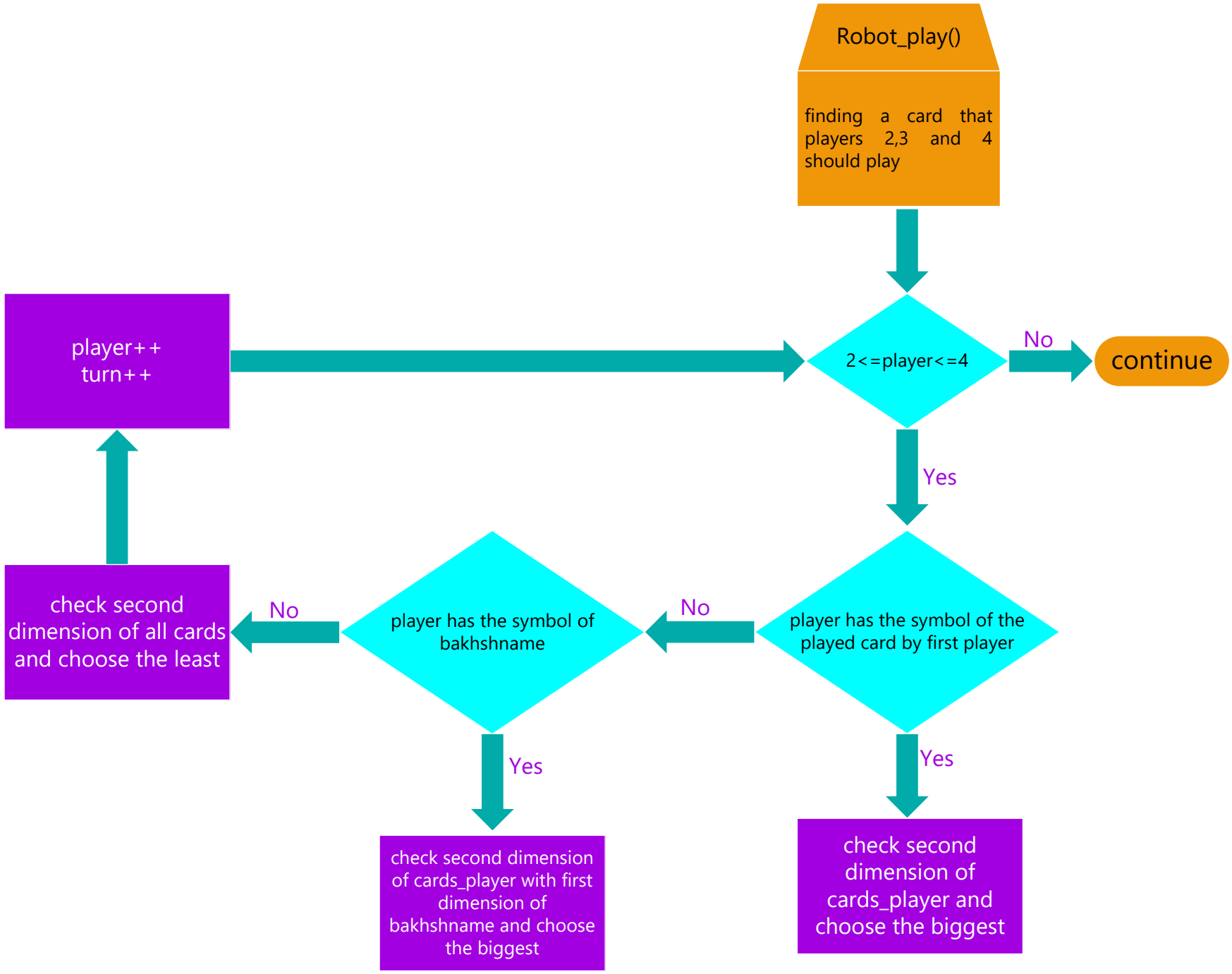
Parisa Yahyapour
STID: 810101551
Flowchart CA2

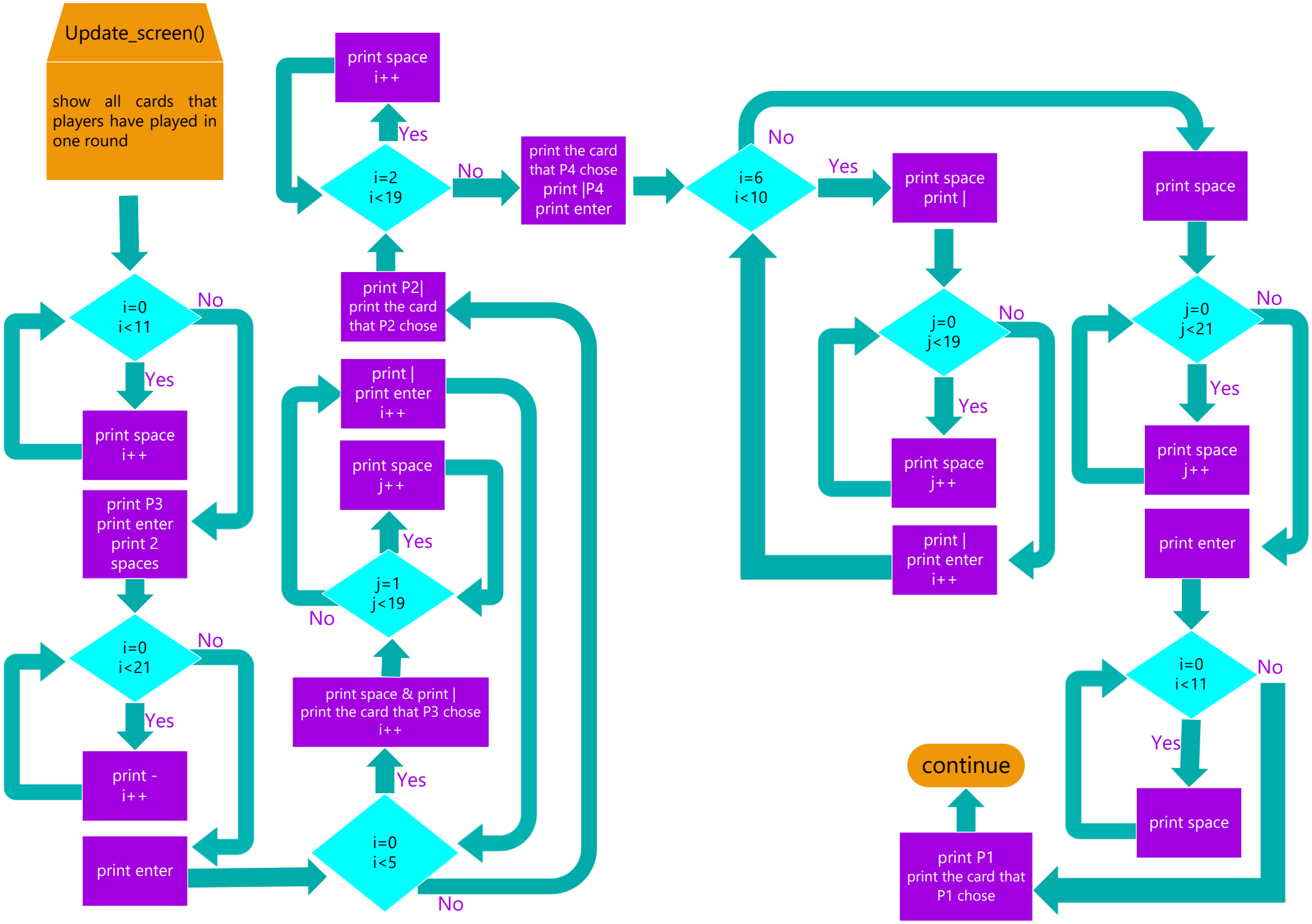

Start
End
Result_table()
Max_point != 6
Yes
No
Max_point == 0
Yes
No
Spread_cards1()
Wrong input
Print:
wrong choose
Start_screen()
Print: choose a
Card as bakhshname
Correct
Find_turn()
Spread_card3()
Spread_cards2()
Judge()
No
player == 1
Yes
Print_screen()
Print: choose
A card to start
Wrong
true
No
Robot_start()
Robot turn?
turn++
Yes
Robot_play()
Yes
Turn >= 5
No
Update_screen()
Robot_play()

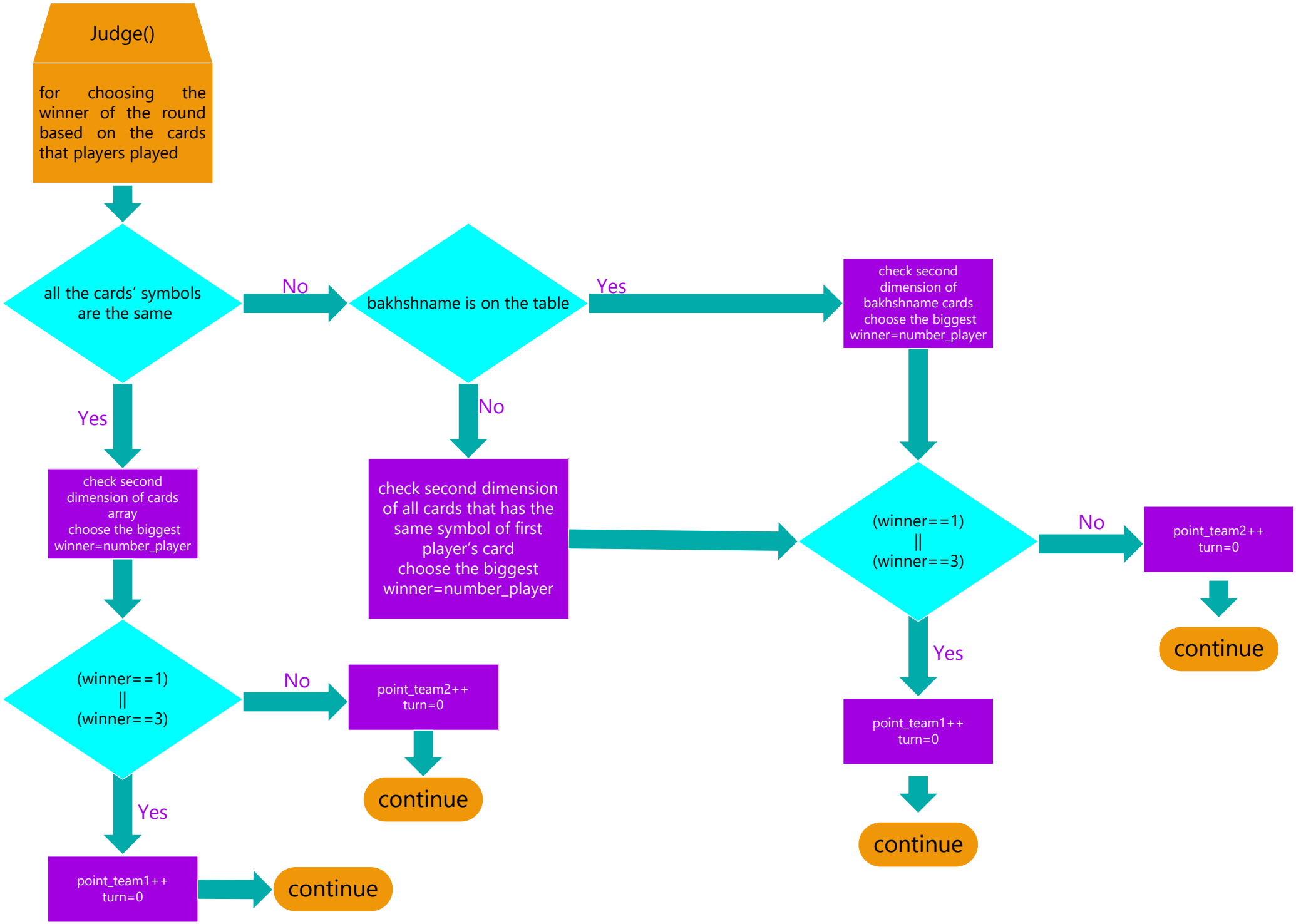

Spread_cards1()
Spreading first 5 cards to player 1 for choosing a symbol as bakhshname
j = 0
j <= 4
No
Yes
first 5 element of card[] goes to cards_player1[]
j++
continue
Start_screen()
printing 5 first cards of player1 for receiving bakhshname
j=0
j<=4
No
Yes
print:
|cards_player1[i] |
print |0|1|2|3|4|
receive a number from player1 as symbol of bakhshname
continue
Robot_start()
chossing a card, when robot is first player
player has cards that are not bakhshname
No
Yes
check second dimension of cards_player with first dimension of bakhshname and choose the biggest
turn++
continue
check second dimension of all cards and choose the biggest
turn++
continue

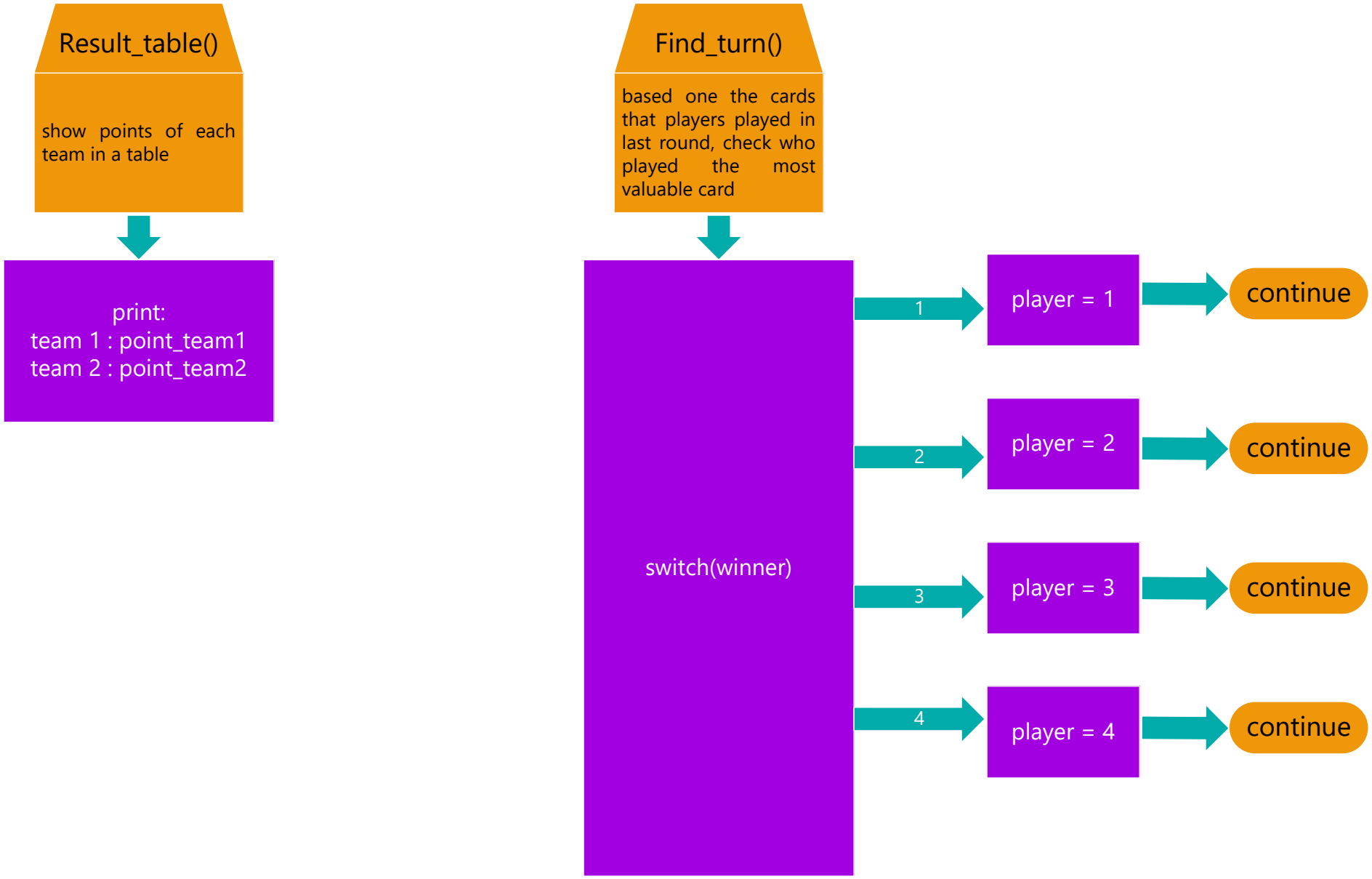

Print_screen()
print the table of game when it's first round and there is no card on table
i=0 i<11
No
Yes
print space i++
print P3 print enter print 2 spaces
i=0 i<21
No
Yes
print - i++
print enter
i=0 i<5
No
Yes
print space print | i++
j=0 j<19
No
Yes
print space j++
print | print enter i++
print P2|
i=0 i<19
No
Yes
print space i++
print P4 print enter
i=6 i<10
No
Yes
print space print |
j=0 j<19
No
Yes
print space j++
print | print enter i++
print space
j=0 j<21
No
Yes
print space j++
print enter
i=0 i<11
No
Yes
print space
print P1 print enter print cards_player1
receive a number from player1 as card
continue


Spread_cards2()
spreading cards after choosing bakhshname by player 1 in this way 5-4-2
j=5 j<=9
No
Yes
second 5 element of card[] goes to cards_player2[] j++
j=10 j<=14
No
Yes
third 5 element of card[] goes to cards_player3[] j++
j=15 j<=19
No
Yes
third 5 element of card[] goes to cards_player4[] j++
j=20 j<=23
No
Yes
next 4 element of card[] goes to cards_player1[] j++
j=24 j<=27
No
Yes
next 4 element of card[] goes to cards_player2[] j++
j=28 j<=31
No
Yes
next 4 element of card[] goes to cards_player3[] j++
j=32 j<=35
No
Yes
next 4 element of card[] goes to cards_player4[] j++
j=36 j<=37
No
Yes
next 2 element of card[] goes to cards_player1[] j++
j=38 j<=39
No
Yes
next 2 element of card[] goes to cards_player2[] j++
j=40 j<=41
No
Yes
next 2 element of card[] goes to cards_player3[] j++
j=42 j<=43
No
Yes
next 4 element of card[] goes to cards_player3[] j++
continue


Robot_play()
finding a card that players 2,3 and 4 should play
2<=player<=4
No
continue
Yes
player has the symbol of the played card by first player
No
player has the symbol of bakhshname
No
check second dimension of all cards and choose the least
player++
turn++
Yes
check second dimension of cards_player and choose the biggest
Yes
check second dimension of cards_player with first dimension of bakhshname and choose the biggest


Update_screen()
show all cards that players have played in one round
i=0
i<11
No
Yes
print space
i++
print P3
print enter
print 2
spaces
i=0
i<21
No
Yes
print -
i++
print enter
i=0
i<5
Yes
No
print space & print |
print the card that P3 chose
i++
j=1
j<19
Yes
No
print space
j++
print |
print enter
i++
print P2|
print the card
that P2 chose
i=2
i<19
Yes
No
print space
i++
print the card
that P4 chose
print |P4
print enter
i=6
i<10
No
Yes
print space
print |
j=0
j<19
No
Yes
print space
j++
print |
print enter
i++
print space
j=0
j<21
No
Yes
print space
j++
print enter
i=0
i<11
No
Yes
print space
print P1
print the card that
P1 chose
continue


Judge()
for choosing the winner of the round based on the cards that players played
all the cards' symbols are the same
No
bakhshname is on the table
Yes
check second dimension of bakhshname cards choose the biggest winner=number_player
Yes
check second dimension of cards array choose the biggest winner=number_player
No
check second dimension of all cards that has the same symbol of first player's card choose the biggest winner=number_player
(winner==1) || (winner==3)
No
point_team2++ turn=0
continue
(winner==1) || (winner==3)
No
point_team2++ turn=0
continue
Yes
point_team1++ turn=0
continue
Yes
point_team1++ turn=0
continue


Result_table()
show points of each team in a table
print:
team 1 : point_team1
team 2 : point_team2
Find_turn()
based one the cards that players played in last round, check who played the most valuable card
switch(winner)
1
player = 1
continue
2
player = 2
continue
3
player = 3
continue
4
player = 4
continue