Author: Miguel Galvez
Created on: November 14th, 2021
Purpose: Flowchart Project 1 (Uno Game)

System Libraries
iostream
cstring
iomanip
fstream
string
ctime
cstdlib
standard namespace

Structures
Uno{
char color[8]
unsigned short value;
}
Deck{
unsigned short size;
Uno *all;
}

Enumerated Values
Values{
ZERO,ONE,TWO,
THREE,FOUR,FIVE,
SIX,SEVEN,EIGHT,
NINE,SKIP,REVERSE
DRAW,WILD,
DRAWWILD

Function Prototypes
create
define
destroy
display
draw
draw1
draw4
draw7
show
toss
wild
wrteRd

main

Set random number seed

Declare variables
const short SZE=108;
const short PLR=54;
bool stat[SZE];
bool skip=0;
fstream in;
fstream out;
Deck set;
Deck fSet;
Deck plr1;
Deck plr2;
Uno pile;
int input;
string in1;

OUTPUT
Welcome players.
Enter 0 to begin or anything else to end
program

INPUT
cin>>in1;

A

```
        ( A )              ( Z )

          |                  |
          v                  |
     /‾‾‾‾‾‾‾‾‾\      F       |
    < in1[0]&&size(in1)==1 >------>  ( EXIT )
     _____/                    ( return 0; )
          | T
          v
    ||  create(set,fSet,SZE)  ||
          |
          v
    ||  create(plr1,plr2,PLR)  ||
          |
          v
    ||  define(&set)  ||
          |
          v
    ||  wrteRd(set,fSet
          out,in)  ||
          |
          v
    |  int i=0  |
          |
          v  <----- i++ <-----
     /‾‾‾‾‾\    T                |
    < i<SZE >------>  | stat[i]=1; |
     \_____/                     
          | F
          v
    || draw7(plr1,plr2    || ----> || draw(stat,SZE,
       fSet,stat,SZE)  ||            pile,fSet) ||
```

(B)   (D)
  ^ T   ^
  | |
 /‾‾‾‾‾‾\   F | skip=0 |
< skip==0 >----->
 _____/
  ^ T
  |
 /‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾\   F
< plr1.size!=0&&plr2.size!=0 >-----> (Y)
 _____/
  ^
  |
  (F)

```
                    ┌─────┐
                    │  B  │
                    └──┬──┘
                       │
                       ▼
              ╱─────────────────╲
             ╱  OUTPUT           ╲
            ╱  Player 1's hand    ╲
            ╲                     ╱
             ╲_____╱
                       │
                       ▼
            ┌─┬───────────────┬─┐
            │ │ display(&plr1) │ │
            └─┴───────────────┴─┘
                       │
                       ▼
              ╱─────────────────╲
             ╱  OUTPUT           ╲
            ╱  Card on top of pile╲
            ╲                     ╱
             ╲_____╱
                       │
                       ▼
            ┌─┬───────────────┬─┐
            │ │  show(&pile)   │ │
            └─┴───────────────┴─┘
                       │
                       ▼
              ╱─────────────────╲
             ╱  OUTPUT           ╲
            ╱  prompt player 1 to  ╲
            ╲  pick card # to throw╱
             ╲  to the pile       ╱
              ╲_____╱
                       │
                       ▼
              ╱─────────────────╲
             ╱  INPUT            ╲
            ╱  cin>>input;        ╲
            ╲_____╱
```

INPUT
cin>>input;

OUTPUT
Player 1's hand

display(&plr1)

OUTPUT
Card on top of pile

show(&pile)

OUTPUT
prompt player 1 to
pick card # to throw
to the pile

INPUT
cin>>input;

B

draw1(plr1,fSet
stat,SZE)

T

input!=100&&
strcmp(plr1.all[input].color,pile.color)
&&plr1.all[input].value!=pile.value&&
plr1.all[input].value<WILD)

F

input==100

F

C

T

INPUT
cin>>input;

OUTPUT
Error message
Prompt player again

```
                          ( C )
                            |
                            v
         plr1.all[input].value
              ==SKIP&&                  T      toss(plr1,pile,input)  -->   skip=1;   -->   ( D )
              !strcmp
         (plr1.all[input].color,
              pile.color)
                            |
                            | F
                            v
         plr1.all[input].value
              ==REVERSE&&
              !strcmp                   T      toss(plr1,pile,input)  -------------------->  ( D )
         (plr1.all[input].color,
              pile.color)
                            |
                            | F
                            v
                            F
    plr1.all[input].value==DRAW&&        T      toss(plr1,pile,input)  -->  draw1(plr2,      ( D )
    !strcmp(plr1.all[input].color,                                          fSet,stat,SZE)
              pile.color)
                            |
                            | F
                            v
    plr1.all[input].value==WILD   -->   wild(plr1,pile,input)  ------------------------->    ( D )
                            |
                            | F
                            v
    plr1.all[input].value==DRAWWILD      T      wild(plr1,pile,input)  -->  draw4(plr2,fSet,  ( D )
                                                                            stat,SZE)
                            |
                            | F
                            v
    !strcmp(plr1.all[input].color,       T      toss(plr1,pile,input)  ------------------->   ( D )
              pile.color)
                            |
                            v
    plr1.all[input].value==pile.value    T      toss(plr1,pile,input)  ------------------->   ( D )
                            |
                            | F
                            v
                          ( D )
```

```mermaid
flowchart TD
    D((D)) --> C1{skip==0}
    C1 -->|T| F((F))
    C1 -->|F| O1[/OUTPUT<br>Player 2's hand/]
    O1 --> P1[[display(&plr2)]]
    P1 --> O2[/OUTPUT<br>Card on top of pile/]
    O2 --> P2[[show(&pile)]]
    P2 --> O3[/OUTPUT<br>prompt player 2 to<br>pick card # to throw<br>to the pile/]
    O3 --> I1[/INPUT<br>cin>>input;/]
    I1 --> C2{input!=100&&<br>strcmp(plr2.all[input].color,pile.color)<br>&&plr2.all[input].value!=pile.value&&<br>plr2.all[input].value<WILD}
    C2 -->|F| C3{input==100}
    C3 -->|T| P3[[draw1(plr2,fSet<br>stat,SZE)]]
    P3 --> D2((D))
    C3 -->|F| E((E))
    C2 -->|T| O4[/OUTPUT<br>Error message<br>Prompt player again/]
    O4 --> I2[/INPUT<br>cin>>input;/]
    I2 --> C2
```

D → skip==0

- T → F
- F → OUTPUT Player 2's hand

OUTPUT Player 2's hand → display(&plr2) → OUTPUT Card on top of pile → show(&pile) → OUTPUT prompt player 2 to pick card # to throw to the pile → INPUT cin>>input;

INPUT cin>>input; → input!=100&& strcmp(plr2.all[input].color,pile.color) &&plr2.all[input].value!=pile.value&& plr2.all[input].value<WILD)

- T → OUTPUT Error message Prompt player again → INPUT cin>>input; → (back to decision)
- F → input==100
  - T → draw1(plr2,fSet stat,SZE) → D
  - F → E

```mermaid
flowchart TD
    E((E))
    E --> D1{plr2.all[input].value<br/>==SKIP&&<br/>!strcmp<br/>(plr2.all[input].color,<br/>pile.color)}
    D1 -->|T| P1[[toss(plr2,pile,input)]] --> S1[skip=1;] --> F1((F))
    D1 -->|F| D2{plr2.all[input].value<br/>==REVERSE&&<br/>!strcmp<br/>(plr2.all[input].color,<br/>pile.color)}
    D2 -->|T| P2[[toss(plr2,pile,input)]] --> F2((F))
    D2 -->|F| D3{plr2.all[input].value==DRAW&&<br/>!strcmp(plr2.all[input].color,<br/>pile.color)}
    D3 -->|T| P3[[toss(plr2,pile,input)]] --> P3b[[draw1(plr1,<br/>fSet,stat,SZE)]] --> F3((F))
    D3 -->|F| D4{plr2.all[input].value==WILD}
    D4 --> P4[[wild(plr2,pile,input)]] --> F4((F))
```

E

plr2.all[input].value ==SKIP&& !strcmp (plr2.all[input].color, pile.color) — T → toss(plr2,pile,input) → skip=1; → F

F ↓

plr2.all[input].value ==REVERSE&& !strcmp (plr2.all[input].color, pile.color) — T → toss(plr2,pile,input) → F

F ↓

plr2.all[input].value==DRAW&& !strcmp(plr2.all[input].color, pile.color) — T → toss(plr2,pile,input) → draw1(plr1, fSet,stat,SZE) → F

F ↓

plr2.all[input].value==WILD → wild(plr2,pile,input) → F

F ↓

plr2.all[input].value==DRAWWILD — T → wild(plr2,pile,input) → draw4(plr1,fSet,stat,SZE) → F

F ↓

!strcmp(plr2.all[input].color, pile.color) — T → toss(plr2,pile,input) → F

↓

plr2.all[input].value==pile.value — T → toss(plr2,pile,input) → F

F ↓

F

```
        ( Y )
          |
          v
     /          \
    /            \
   <  plr1.size==0 >----F---->  / OUTPUT      /
    \            /             / Congratulate /
     \          /             / Player 2     /
          |                          |
          T                          |
          |                          |
          v                          |
  / OUTPUT       /                   |
 / Congratulate /                    |
 / player 1     /                    |
          |                          |
          v <------------------------+
   | destroy(set,fSet, |
   |   plr1,plr2)      |
          |
          v
        ( Z )
```

```
            ┌─────────────┐
            │   create()  │
            └──────┬──────┘
                   │
                   ▼
        ┌────────────────────┐
        │  set.all=new Uno   │
        │       [SZE];       │
        └─────────┬──────────┘
                  │
                  ▼
        ┌────────────────────┐
        │   set.size=SZE;    │
        └─────────┬──────────┘
                  │
                  ▼
        ┌────────────────────┐
        │  fSet.all=new Uno  │
        │       [SZE];       │
        └─────────┬──────────┘
                  │
                  ▼
        ┌────────────────────┐
        │   fSet.size=SZE;   │
        └─────────┬──────────┘
                  │
                  ▼
            ┌─────────────┐
            │    EXIT     │
            │return to main()│
            └─────────────┘
```

```
define()

K

int i=0          i++

i<x->size  --T-->  i<25  --T-->  strcpy(x->all[i].color, "Green");  -->  F
   |
   F
   |
  Exit
return to main()

i<50  --T-->  strcpy(x->all[i].color, "Blue");  -->  G
  F

i<75  --T-->  strcpy(x->all[i].color, "Red");  -->  H
  F

i<100  --T-->  strcpy(x->all[i].color, "Yellow");  -->  I
  F

strcpy(x->all[i].color, "Black");  -->  J
```

```
┌───┐
│ F │
└─┬─┘
  │
  ▼
◇ i==0 ──T──▶ [ all[i].value==ZERO ] ──────────┐
  │F                                            │
  ▼                                             │
◇ i<=2 ──T──▶ [ all[i].value==ONE ] ───────────┤
  │F                                            │
  ▼                                             │
◇ i<=4 ──T──▶ [ all[i].value==TWO ] ───────────┤
  │F                                            │
  ▼                                             │
◇ i<=6 ──T──▶ [ all[i].value==THREE ] ─────────┤
  │F                                            │
  ▼                                             │
◇ i<=8 ──T──▶ [ all[i].value==FOUR ] ──────────┤
  │F                                            │
  ▼                                             │
◇ i<=10 ──T──▶ [ all[i].value==FIVE ] ─────────┤
  │F                                            │
  ▼                                             │
◇ i<=12 ──T──▶ [ all[i].value==SIX ] ──────────┤
  │F                                            │
  ▼                                             │
◇ i<=14 ──T──▶ [ all[i].value==SEVEN ] ────────┤
  │F                                            │
  ▼                                             │
◇ i<=16 ──T──▶ [ all[i].value==EIGHT ] ────────┤
  │F                                            │
  └───────────────────────────────────────────┘

◇ i<=18 ──T──▶ [ all[i].value==NINE ] ──────────▶
  │F
  ▼
◇ i<=20 ──T──▶ [ all[i].value==SKIP ] ──────────▶
  │F
  ▼
◇ i<=22 ──T──▶ [ all[i].value==REVERSE ] ───────▶
  │F
  ▼
[ all[i].value==DRAW ] ──────────────────────────▶

                                              ┌───┐
                                              │ K │
                                              └───┘
```

```mermaid
flowchart TD
    G((G)) --> D25{i==25}
    D25 -->|T| ZERO[all i .value==ZERO]
    D25 -->|F| D27{i<=27}
    D27 -->|T| ONE[all i .value==ONE]
    D27 -->|F| D29{i<=29}
    D29 -->|T| TWO[all i .value==TWO]
    D29 -->|F| D31{i<=31}
    D31 -->|T| THREE[all i .value==THREE]
    D31 -->|F| D33{i<=33}
    D33 -->|T| FOUR[all i .value==FOUR]
    D33 -->|F| D35{i<=35}
    D35 -->|T| FIVE[all i .value==FIVE]
    D35 -->|F| D37{i<=37}
    D37 -->|T| SIX[all i .value==SIX]
    D37 -->|F| D39{i<=39}
    D39 -->|T| SEVEN[all i .value==SEVEN]
    D39 -->|F| D41{i<=41}
    D41 -->|T| EIGHT[all i .value==EIGHT]
    D41 -->|F| D43{i<=43}
    D43 -->|T| NINE[all i .value==NINE]
    D43 -->|F| D45{i<=45}
    D45 -->|T| SKIP[all i .value==SKIP]
    D45 -->|F| D47{i<=47}
    D47 -->|T| REVERSE[all i .value==REVERSE]
    D47 -->|F| DRAW[all i .value==DRAW]
    DRAW --> K((K))
```

- G
- i==25 — T → all[i].value==ZERO
- i<=27 — T → all[i].value==ONE
- i<=29 — T → all[i].value==TWO
- i<=31 — T → all[i].value==THREE
- i<=33 — T → all[i].value==FOUR
- i<=35 — T → all[i].value==FIVE
- i<=37 — T → all[i].value==SIX
- i<=39 — T → all[i].value==SEVEN
- i<=41 — T → all[i].value==EIGHT
- i<=43 — T → all[i].value==NINE
- i<=45 — T → all[i].value==SKIP
- i<=47 — T → all[i].value==REVERSE
- F → all[i].value==DRAW
- K

```
H

i==50 ──T──▶ all[i].value==ZERO ──────────────▶
  │F
  ▼
i<=52 ──T──▶ all[i].value==ONE ───────────────▶
  │F
  ▼
i<=54 ──T──▶ all[i].value==TWO ───────────────▶
  │F
  ▼
i<=56 ──T──▶ all[i].value==THREE ─────────────▶
  │F
  ▼
i<=58 ──T──▶ all[i].value==FOUR ──────────────▶
  │F
  ▼
i<=60 ──T──▶ all[i].value==FIVE ──────────────▶
  │F
  ▼
i<=62 ──T──▶ all[i].value==SIX ───────────────▶
  │F
  ▼
i<=64 ──T──▶ all[i].value==SEVEN ─────────────▶
  │F
  ▼
i<=66 ──T──▶ all[i].value==EIGHT ─────────────▶
  │F
  ▼

i<=68 ──T──▶ all[i].value==NINE ──────────────▶
  │F
  ▼
i<=70 ──T──▶ all[i].value==SKIP ──────────────▶
  │F
  ▼
i<=72 ──T──▶ all[i].value==REVERSE ───────────▶
  │F
  ▼
all[i].value==DRAW ───────────────────────────▶

K
```

```
       ( F )
         |
         v
    < i==75 > --T--> [ all[i].value==ZERO ] -------->
         |F
         v
    < i<=77 > --T--> [ all[i].value==ONE ] --------->
         |F
         v
    < i<=79 > --T--> [ all[i].value==TWO ] --------->
         |F
         v
    < i<=81 > --T--> [ all[i].value==THREE ] ------->
         |F
         v
    < i<=83 > --T--> [ all[i].value==FOUR ] -------->
         |F
         v
    < i<=85 > --T--> [ all[i].value==FIVE ] -------->
         |F
         v
    < i<=87 > --T--> [ all[i].value==SIX ] --------->
         |F
         v
    < i<=89 > --T--> [ all[i].value==SEVEN ] ------->
         |F
         v
    < i<=91 > --T--> [ all[i].value==EIGHT ] ------->
         |F


    < i<=93 > --T--> [ all[i].value==NINE ] ------->
         |F
         v
    < i<=95 > --T--> [ all[i].value==SKIP ] ------->
         |F
         v
    < i<=97 > --T--> [ all[i].value==REVERSE ] ---->
         |F
         v
    [ all[i].value==DRAW ] ------------------------->


                                        ( K )
```

```
      ┌─────┐
      │  J  │
      └──┬──┘
         │
         ▼
       ╱╲                    T      ┌──────────────────┐
      ╱  ╲ ─────────────────────────▶│ x->all[i].value=WILD │──────┐
     ╱i<=103╲                        └──────────────────┘      │
      ╲    ╱                                                     │
       ╲  ╱                                                      │
        ╲╱                                                       │
         │  F                                                    │
         ▼                                                       ▼
  ┌──────────────┐                                           ┌─────┐
  │ x->all[i].value= │──────────────────────────────────────▶│  K  │
  │   DRAWWILD      │                                          └─────┘
  └──────────────┘
```

```
        destroy()
           │
           ▼
   ┌──────────────────┐
   │ delete [] set.all; │
   └──────────────────┘
           │
           ▼
   ┌──────────────────┐
   │ delete [] fSet.all; │
   └──────────────────┘
           │
           ▼
   ┌──────────────────┐
   │ delete [] plr1.all; │
   └──────────────────┘
           │
           ▼
   ┌──────────────────┐
   │ delete [] plr2.all; │
   └──────────────────┘
           │
           ▼
     return to main()
```

```
                    ┌─────────────┐
                    │  display()  │
                    └──────┬──────┘
                           │
                           ▼
                  ╱────────────────╲
                 ╱  OUTPUT          ╲
                ╱  header/formatting ╲
                ╲  for card # ,       ╲
                 ╲  color,           ╱
                  ╲  and value      ╱
                    ╲──────┬───────╱
                           │
                           ▼
                    ┌─────────────┐
                    │   int i=0   │
                    └──────┬──────┘
                           │
                           ▼
        ╱◇╲        T    ╱──────────╲      ╱◇╲    T   ╱──────────╲
     ╱ i<x->size ╲────▶│  OUTPUT   │────▶╱x->all[i].╲──▶│ OUTPUT  │───▶
      ╲          ╱      │ card color│     ╲value==SKIP╱   │ "SKIP"  │
        ╲◇╱            ╲──────────╱        ╲◇╱          ╲──────────╱
         │                                   │ F
         │ F                                 ▼
         ▼                                  ╱◇╲    T   ╱──────────╲
   ╱──────────╲                        ╱x->all[i].╲──▶│  OUTPUT  │───▶
  │   EXIT     │                        ╲value==    ╱  │"REVERSE" │
  │ return to  │                         ╲REVERSE  ╱   ╲──────────╱
  │  main()    │                           ╲◇╱
   ╲──────────╱                             │ F
                                            ▼
                                           ╱◇╲    T   ╱──────────╲
                                      ╱x->all[i].╲──▶│  OUTPUT  │───▶
                                       ╲value==   ╱   │ "DRAW +1"│
                                        ╲DRAW    ╱    ╲──────────╱
                                          ╲◇╱
                                           │ F
                                           ▼
                                          ╱◇╲    T   ╱──────────╲
                                     ╱x->all[i].╲──▶│  OUTPUT  │───▶
                                      ╲value==   ╱   │"WILDCARD"│
                                       ╲WILD    ╱    ╲──────────╱
                                         ╲◇╱
                                          │ F
                                          ▼
                                         ╱◇╲     T   ╱──────────╲
                                    ╱x->all[i].╲──▶│  OUTPUT  │───▶
                                     ╲value==    ╱   │ "DRAW+4" │
                                      ╲DRAW WILD ╱    ╲──────────╱
                                        ╲◇╱
                                         │ F
                                         ▼
                                    ╱──────────╲
                                   │  OUTPUT    │───▶
                                   │x->all[i].value│
                                    ╲──────────╱
```

```
draw()
```

```
unsigned int
indx=rand()%SZE;
```

```
!stat[indx]
```

T → `indx=rand()%SZE;`

F

```
strcpy(pile.color,fSet.all[indx].color);
pile.value=fSet.all[indx].value;
```

```
stat[indx]=0;
```

```
return to main()
```

```
              ┌─────────────┐
              │   draw1()   │
              └─────────────┘
                     │
                     ▼
          ┌────────────────────┐
          │   unsigned int     │
          │  indx=rand()%SZE;  │
          └────────────────────┘
                     │
                     ▼
                  ◇────────◇         ┌──────────────────────┐
                 ╱          ╲   T     │                      │
                ◇ !stat[indx] ◇──────▶│  indx=rand()%SZE;    │
                 ╲          ╱         │                      │
                  ◇────────◇          └──────────────────────┘
                     │ F
                     ▼
      ┌─────────────────────────────────────┐
      │                                     │
      │  strcpy(pile.color,fSet.all[indx].color);│
      │     pile.value=fSet.all[indx].value;│
      │                                     │
      └─────────────────────────────────────┘
                     │
                     ▼
          ┌────────────────────┐
          │   stat[indx]=0;    │
          └────────────────────┘
                     │
                     ▼
          ┌────────────────────┐
          │    plr.size++;     │
          └────────────────────┘
                     │
                     ▼
              ┌────────────────┐
              │ return to main()│
              └────────────────┘
```

```
draw4()
   │
   ▼
unsigned int
indx=rand()%SZE;
   │
   ▼
plr.size+=4
   │
   ▼
int i=plr.size-4;
   │
   ▼
i<plr.size ──T──► !stat[indx] ──T──► indx=rand()%SZE;
   │                    │
   F                    F
   │                    │
   ▼                    ▼
return to main()   strcpy(pile.color,fSet.all[indx].color);
                   pile.value=fSet.all[indx].value;
                        │
                        ▼
                   stat[indx]=0;
                        │
                        ▼
                   plr.size++; ──────────► i++
```

```
draw4()
  │
  ▼
plr1.size=0;
  │
  ▼
unsigned int
indx=rand()%SZE;
  │
  ▼
int i=0;
  │
  ▼
i<7 ──T──► !stat[indx] ──T──► indx=rand()%SZE;
  │                │
  F                F
  │                │
  ▼                ▼
  L            plr1.size++;
                   │
                   ▼
      strcpy(plr1.color,fSet.all[indx].color);
      plr1.all[i].value=fSet.all[indx].value;
                   │
                   ▼
              stat[indx]=0; ──────► i++
```

draw4()

plr1.size=0;

unsigned int indx=rand()%SZE;

int i=0;

i<7

L

!stat[indx]

indx=rand()%SZE;

plr1.size++;

strcpy(plr1.color,fSet.all[indx].color);

plr1.all[i].value=fSet.all[indx].value;

stat[indx]=0;

i++

```
                    ( L )
                      │
                      ▼
            ┌───────────────────┐
            │    plr2.size=0;    │
            └───────────────────┘
                      │
                      ▼
            ┌───────────────────┐
            │   unsigned int     │
            │ indx=rand()%SZE;   │
            └───────────────────┘
                      │
                      ▼
            ┌───────────────────┐
            │     int i=0;       │
            └───────────────────┘
                      │
                      ▼                                          ┌───────────────────┐
                   ◇ i<7 ◇────T──────────────────────┐          │        i++         │
                      │                               │          └───────────────────┘
                      F                               ▼                    ▲
                      │                            ◇ !stat[indx] ◇──T──► indx=rand()%SZE;
                      ▼                               │
               ╭──────────────╮                      F
               │ return to     │                     │
               │   main()      │                     ▼
               ╰──────────────╯            ┌───────────────────┐
                                           │     plr2.size++;    │
                                           └───────────────────┘
                                                     │
                                                     ▼
                                    ┌──────────────────────────────────────┐
                                    │ strcpy(plr2.color,fSet.all[indx].color);│
                                    │                                        │
                                    │ plr2.all[i].value=fSet.all[indx].value;│
                                    └──────────────────────────────────────┘
                                                     │
                                                     ▼
                                           ┌───────────────────┐
                                           │   stat[indx]=0;     │
                                           └───────────────────┘
```

```
                    ┌─────────────┐
                   (   show()     )
                    └──────┬──────┘
                           │
                    ┌──────▼──────┐
                    │   OUTPUT    │
                    │ card color  │
                    └──────┬──────┘
                           │
                    ┌──────▼──────┐        T      ┌──────────────┐
                    │x->value==SKIP? ├───────────►│   OUTPUT     │──────────────►
                    └──────┬──────┘               │   "SKIP"     │
                           │ F                     └──────────────┘
                    ┌──────▼──────────┐    T      ┌──────────────┐
                    │x->value==REVERSE? ├─────────►│   OUTPUT     │──────────────►
                    └──────┬──────────┘           │  "REVERSE"   │
                           │ F                     └──────────────┘
                    ┌──────▼──────────┐    T      ┌──────────────┐
                    │x->value==DRAW?   ├─────────►│   OUTPUT     │──────────────►
                    └──────┬──────────┘           │  "DRAW+1"    │
                           │ F                     └──────────────┘
                    ┌──────▼──────────┐    T      ┌──────────────┐
                    │x->value==WILD?   ├─────────►│   OUTPUT     │──────────────►
                    └──────┬──────────┘           │  "WILDCARD"  │
                           │ F                     └──────────────┘
                    ┌──────▼──────────┐    T      ┌──────────────┐
                    │x->value==DRAW    ├─────────►│   OUTPUT     │──────────────►
                    │WILD?             │          │  "DRAW+4"    │
                    └──────┬──────────┘           └──────────────┘
                           │
                    ┌──────▼──────┐                              ┌──────────────┐
                    │   OUTPUT    │─────────────────────────────►│    EXIT      │
                    │  x->value   │                              │return to main()│
                    └─────────────┘                              └──────────────┘
```

```
toss()
```

```
strcpy(pile.color,plr.all[input].color);
pile.value=plr.all[input].value;
```

```
int i=input
```

```
i<plr.size
```

T

```
strcpy(plr.all[i].color,plr.all[i+1].color);
plr.all[i].value=plr.all[i+1].value;
```

```
i++
```

F

```
plr.size--
```

```
EXIT
return to main()
```

```
wild()
```

pile.value=plr.all[input].value;

int i=input

i<plr.size

T → strcpy(plr.all[i].color,plr.all[i+1].color);
plr.all[i].value=plr.all[i+1].value;

i++

F

OUTPUT
Prompt player for numbers 1-4
1-yellow
2-green
3-red
4-blue

cin>>input

input>4||input<0   T

F

input==1   T → strcpy(pile.color,"Yellow")

F

input==2   T → strcpy(pile.color,"Green")

F

input==3   T → strcpy(pile.color,"Red")

F

strcpy(pile.color,"Blue")

plr.size--;

EXIT
return to main()

```
wrteRd()
```

```
out.open("unoCards.dat",ios::out|ios::binary);
```

```
out.write(reinterpret_cast<char *>(&set),sizeof(set)*set.size);
```

```
out.close();
```

```
in.open("unoCards.dat",ios::in|ios::binary);
```

```
in.read(reinterpret_cast<char *>(&fSet),sizeof(fSet)*fSet.size);
```

```
in.close();
```