SimpleLotteryMachine (2.0)

Kcits970

December 30, 2021

Sections

Section 1. Description

The purpose of the program 'SimpleLotteryMachine' is to simulate a basic Powerball game. It provides a menu-driven interface to purchase, sell, and view tickets.

Base Problem

The idea for this program is based on an exam problem from a programming class[1] I took this year. The original problem is basically a lottery ticket checker, and it follows a few sets of definitions and requirements.

> **Lottery Type:** In a classic Powerball game, the user chooses 5 unique integers within [1, 69]. If we generalize this rule, the user can choose L unique integers within the range of [1, N^2][2]. We define the value N as the 'Lottery Type'.
> **Lottery Size:** The value L.
> **Lottery Seed:** A number used to generate lottery tickets. If the same seed is used, then the generated tickets will always contain the same value.
> **Requirements:** Fix the lottery size to 6, and let the user decide the lottery type. Generate a random ticket using a random seed, and save it to a text file. Then generate a new ticket using 202112 as the seed, and compare the two tickets. If the two tickets match, the program should output "won". If not, then the program should output "lost".

Modified Problem

In attempts to approach this problem differently, I modified some of the definitions and requirements.

> **Lottery Range:** The original definition of 'Lottery Type'.
> **Ticket Type:** There are 2 types of tickets: 'auto' and 'manual'. Auto tickets are automatically generated by the machine, and manual tickets are manually written by the user.
> **Requirement 1:** Fix the lottery range and size to 45 and 6.
> **Requirement 2:** At program launch, generate the ticket that contains the winning numbers.
> **Requirement 3:** Provide a menu-driven interface to purchase, sell, and view tickets. Let the user choose the ticket type when the user selects the purchase menu.
> **Optional Requirement:** Add any other preferred features.

---

[1] Dankook University, SW Convergence Coding 2, Prof. Samuel Woo (2021, Fall Semester)
[2] N must always be an integer greater than L.

Assigning Extra Properties to Tickets

Because of **Requirement 3,** which states that the program must provide an interface to buy/sell tickets, I assigned extra properties to tickets. Rank and worth.

Ticket Rank: Ticket ranks are numerical values that determine how close a ticket is to the winning ticket. Ranks are calculated with the formula below.

$$Rank = Ticket\ Size - Number\ of\ Matches + 1$$

In this case, the size of a ticket is fixed to 6, so the formula can be reduced.

$$Rank = 7 - Number\ of\ Matches$$

Because the number of matches ranges from 0 to 6, ticket ranks range from 1 to 7. If the rank of a ticket is 1, that means all of the numbers match. If the rank is 2, that means the ticket is off by 1 number. If the rank is 3, the ticket is off by 2 numbers, and so on.

Ticket Worth: Ticket worths are numerical values that display the worth of a ticket. Ticket worths depend on the probability of getting the certain rank. Because the percentage of getting rank 5 or lower is about 97.6%, worths are not assigned for those tickets.

We start by calculating the probability for tickets that rank higher than 5.

$$P(Rank\ 4) = \frac{\binom{6}{3} \times \binom{39}{3}}{\binom{45}{6}} = 0.02244 \dots [3]$$

$$P(Rank\ 3) = \frac{\binom{6}{4} \times \binom{39}{2}}{\binom{45}{6}} = 0.00136 \dots$$

$$P(Rank\ 2) = \frac{\binom{6}{5} \times \binom{39}{1}}{\binom{45}{6}} = 0.00002 \dots$$

---

[3] $\binom{n}{k}$ is the binomial coefficient.

$$P(Rank\ 1) = \frac{\binom{6}{6} \times \binom{39}{0}}{\binom{45}{6}} = 0.00000\dots$$

We then assign \$5 for rank 4 tickets, and multiply the ratio of probability for higher ranked tickets.

$$Worth(Rank\ 4) = \$5$$

$$Worth(Rank\ 3) = Worth(Rank\ 4) \times \frac{P(Rank\ 4)}{P(Rank\ 3)} = \$82\text{[4]}$$

$$Worth(Rank\ 2) = Worth(Rank\ 3) \times \frac{P(Rank\ 3)}{P(Rank\ 2)} = \$3906$$

$$Worth(Rank\ 1) = Worth(Rank\ 2) \times \frac{P(Rank\ 2)}{P(Rank\ 1)} = \$913900$$

The program is written in C language, and compiled using Microsoft Visual C++ 14.2. Attempts to compile the given source code with a different compiler may result in unexpected or undefined behavior.

---

[4] The calculations are rounded to ensure that the values are integers.

Section 2. Features

The features of 'SimpleLotteryProgram' can be divided into 4 main parts, which are ticket purchase, ticket details, ticket sale, and user status.

Ticket Purchase

The main options are prompted at program launch.

```
>>LOTTERY MACHINE: VERSION 2

    <MAIN OPTIONS>
PURCHASE TICKET ---------------- A
BULK PURCHASE TICKETS ---------- B
SELL TICKET -------------------- C
SELL ALL TICKETS --------------- D
VIEW TICKET DETAILS ------------ E
VIEW ALL TICKETS --------------- F
VIEW USER STATUS --------------- G
EXIT --------------------------- Q
>>
```

Each ticket costs $5. To purchase a ticket, select the PURCHASE TICKET option. From there, the user can choose a ticket type.

```
    <TICKET PURCHASE PROMPT>
AUTO TICKET -------------------- A
MANUAL TICKET ------------------ B
SELECT A TYPE:
```

If AUTO TICKET is selected, a new ticket will be automatically generated and added to the list. If MANUAL TICKET is selected, a blank ticket sheet will be prompted, asking the user to select 6 unique numbers. The machine will then validate the input, and add it to the list. Below are examples of valid and invalid inputs when purchasing a manual ticket.

Valid Input: Numbers separated by whitespace

```
    <TICKET PURCHASE PROMPT>
AUTO TICKET -------------------- A
MANUAL TICKET ------------------ B
SELECT A TYPE: B

    <TICKET SHEET>
01 02 03 04 05 06 07 08
09 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45
```

```
SELECT 6 UNIQUE NUMBERS: 21 22 41 30 1 4
PURCHASED A TICKET FOR $5
USER NOW HAS $95
>>
```

Valid Input: Numbers separated by non-numeric characters

```
    <TICKET SHEET>
01 02 03 04 05 06 07 08
09 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45
SELECT 6 UNIQUE NUMBERS: 1b3#4%44& 34(23
PURCHASED A TICKET FOR $5
USER NOW HAS $90
>>
```

Invalid Input: Not enough numbers

```
    <TICKET SHEET>
01 02 03 04 05 06 07 08
09 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45
SELECT 6 UNIQUE NUMBERS: 1 2 3 4 5
FALSE TICKET: THE INPUT DOES NOT CONTAIN ENOUGH NUMBERS
FAILED TO PURCHASE TICKET
USER NOW HAS $90
>>
```

Invalid Input: Duplicate numbers

```
    <TICKET SHEET>
01 02 03 04 05 06 07 08
09 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45
SELECT 6 UNIQUE NUMBERS: 1 1 1 2 3 4
FALSE TICKET: THE INPUT CONTAINS DUPLICATE NUMBERS
FAILED TO PURCHASE TICKET
USER NOW HAS $90
>>
```

Invalid Input: Numbers out of range

```
    <TICKET SHEET>
01 02 03 04 05 06 07 08
09 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45
SELECT 6 UNIQUE NUMBERS: 1 2 3 4 5 1000
```

```
FALSE TICKET: INPUTS ARE NOT WITHIN RANGE
FAILED TO PURCHASE TICKET
USER NOW HAS $90
>>
```

The user can bulk purchase tickets by selecting the BULK
PURCHASE TICKETS option. From there, the machine will ask for
the desired number of tickets[5], as well as the desired number
of auto tickets. Then it will automatically calculate the
needed number of tickets for each type, and add them to the
list.

```
    <TICKET BULK PURCHASE PROMPT>
ENTER THE PREFERRED NUMBER OF TICKETS: 10
ENTER THE PREFERRED NUMBER OF AUTO TICKETS: 9
GENERATING 9 AUTO TICKET(S), 1 MANUAL TICKET(S)

    <TICKET SHEET>
01 02 03 04 05 06 07 08
09 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45
SELECT 6 UNIQUE NUMBERS: 1 2 3 4 5 6
PURCHASED 10 TICKET(S) FOR $50
USER NOW HAS $40
>>
```

Ticket Details

Every time a ticket is successfully purchased, it is added
to a list. To prevent the user from adding infinitely many
tickets, the size of this list is limited to 100. If a new
purchase is made while the list is full, then the last ticket
is deleted[6]. To view the ticket list, simply select the VIEW
ALL TICKETS option.

```
    <LIST OF ALL TICKETS>
TICKET 001: 01 02 03 04 05 06 (RANK 5, WORTH $0)
TICKET 002: 01 08 09 17 23 34 (RANK 6, WORTH $0)
TICKET 003: 07 19 22 29 40 41 (RANK 7, WORTH $0)
TICKET 004: 01 14 16 20 25 26 (RANK 7, WORTH $0)
TICKET 005: 09 13 25 31 41 44 (RANK 7, WORTH $0)
TICKET 006: 02 07 11 23 24 33 (RANK 5, WORTH $0)
TICKET 007: 05 16 20 21 24 26 (RANK 6, WORTH $0)
```

---

[5] The user can bulk purchase a maximum of 100 tickets at once. If greater value is
inputted, then that value will be force-changed to 100.
[6] Deleting a ticket differs from selling a ticket. Ticket deletions are forced actions,
so the ticket's worth is not returned to the user when it's deleted.

```
TICKET 008: 10 17 20 23 32 36 (RANK 6, WORTH $0)
TICKET 009: 07 11 19 21 36 40 (RANK 7, WORTH $0)
TICKET 010: 07 08 16 18 29 35 (RANK 6, WORTH $0)
TICKET 011: 01 03 04 23 34 44 (RANK 6, WORTH $0)
TICKET 012: 01 04 21 22 30 41 (RANK 7, WORTH $0)
>>
```

The VIEW ALL TICKETS option displays basic information of all tickets: their numbers, rank, and worth. To view further details of a single ticket, select the VIEW TICKET DETAILS option. From there, the user can select a specific ticket and view more details.

```
    <TICKET SELECTION PROMPT>
TICKET 001: 01 02 03 04 05 06 (RANK 5, WORTH $0)
TICKET 002: 01 08 09 17 23 34 (RANK 6, WORTH $0)
TICKET 003: 07 19 22 29 40 41 (RANK 7, WORTH $0)
TICKET 004: 01 14 16 20 25 26 (RANK 7, WORTH $0)
TICKET 005: 09 13 25 31 41 44 (RANK 7, WORTH $0)
TICKET 006: 02 07 11 23 24 33 (RANK 5, WORTH $0)
TICKET 007: 05 16 20 21 24 26 (RANK 6, WORTH $0)
TICKET 008: 10 17 20 23 32 36 (RANK 6, WORTH $0)
TICKET 009: 07 11 19 21 36 40 (RANK 7, WORTH $0)
TICKET 010: 07 08 16 18 29 35 (RANK 6, WORTH $0)
TICKET 011: 01 03 04 23 34 44 (RANK 6, WORTH $0)
TICKET 012: 01 04 21 22 30 41 (RANK 7, WORTH $0)
SELECT A TICKET: 6

    <TICKET DETAILS>
02 07 11 23 24 33 (RANK 5, WORTH $0)
GENERATED AT: Thu Dec 30 19:33:10 2021
MATCHING NUMBERS: 23 33
>>
```

The VIEW TICKET DETAILS option displays the ticket's numbers, rank, worth, generation time, as well as the detailed comparison to the winning ticket. This information is very helpful for finding the winning numbers within a couple of trials.

Ticket Sale

Aside from purchasing tickets, the user can also sell them. To sell a ticket, select the SELL TICKET option. The machine will output a ticket selection prompt. From there, the user can select a ticket to sell.

```
    <TICKET SELECTION PROMPT>
TICKET 001: 05 06 23 27 33 35 (RANK 1, WORTH $913900)
TICKET 002: 01 05 06 23 27 33 (RANK 2, WORTH $3906)
TICKET 003: 01 02 05 06 23 27 (RANK 3, WORTH $82)
TICKET 004: 01 02 03 05 06 23 (RANK 4, WORTH $5)
TICKET 005: 37 38 39 42 43 45 (RANK 7, WORTH $0)
```

```
TICKET 006: 12 15 27 28 32 37 (RANK 6, WORTH $0)
TICKET 007: 01 02 03 04 05 06 (RANK 5, WORTH $0)
TICKET 008: 01 08 09 17 23 34 (RANK 6, WORTH $0)
TICKET 009: 07 19 22 29 40 41 (RANK 7, WORTH $0)
TICKET 010: 01 14 16 20 25 26 (RANK 7, WORTH $0)
TICKET 011: 09 13 25 31 41 44 (RANK 7, WORTH $0)
TICKET 012: 02 07 11 23 24 33 (RANK 5, WORTH $0)
TICKET 013: 05 16 20 21 24 26 (RANK 6, WORTH $0)
TICKET 014: 10 17 20 23 32 36 (RANK 6, WORTH $0)
TICKET 015: 07 11 19 21 36 40 (RANK 7, WORTH $0)
TICKET 016: 07 08 16 18 29 35 (RANK 6, WORTH $0)
TICKET 017: 01 03 04 23 34 44 (RANK 6, WORTH $0)
TICKET 018: 01 04 21 22 30 41 (RANK 7, WORTH $0)
SELECT A TICKET: 4

    <TICKET SALE PROMPT>
TICKET SOLD FOR $5
USER NOW HAS $15
>>
```

If the user wishes to sell all of the owned tickets, select the SELL ALL TICKETS option. All of the tickets will be sold sequentially, and the worth will be returned to the user.

```
    <TICKET SALE PROMPT>
TICKET SOLD FOR $913900
TICKET SOLD FOR $3906
TICKET SOLD FOR $82
TICKET SOLD FOR $0
TICKET SOLD FOR $0
TICKET SOLD FOR $0
TICKET SOLD FOR $0
TICKET SOLD FOR $0
TICKET SOLD FOR $0
TICKET SOLD FOR $0
TICKET SOLD FOR $0
TICKET SOLD FOR $0
TICKET SOLD FOR $0
TICKET SOLD FOR $0
TICKET SOLD FOR $0
TICKET SOLD FOR $0
TICKET SOLD FOR $0
17 TICKET(S) SOLD FOR $917888
USER NOW HAS $917903
>>
```

User Status

The current status of the user can be viewed by selecting the VIEW USER STATUS option. The status displays basic information of the user as shown below.
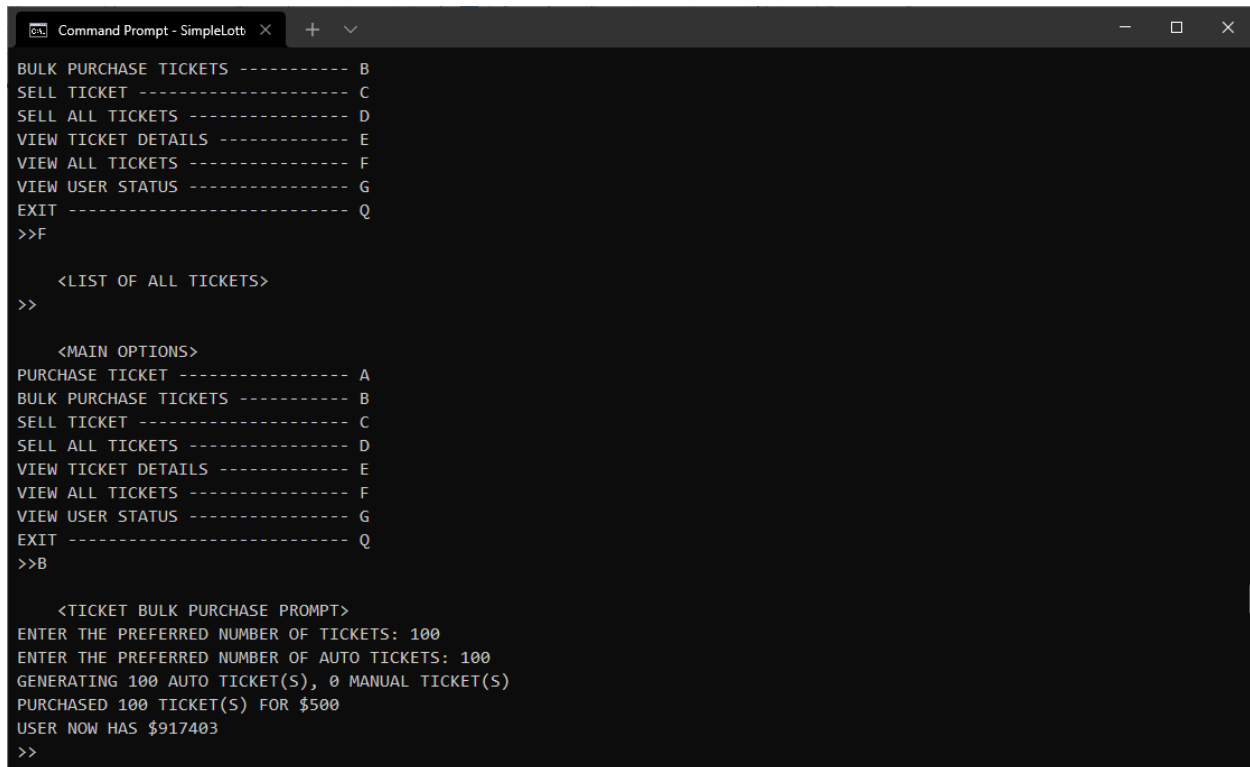
```
    <USER STATUS>
```

```
CASH: $917403
NUMBER OF TICKETS BOUGHT: 118
NUMBER OF TICKETS SOLD: 18
NUMBER OF OWNED TICKETS: 100
>>
```

For the user status, the following equation holds true, unless the tickets were force deleted by the machine.

$$Num\ of\ Tickets\ Bought = Sold\ Tickets + Owned\ Tickets$$

Section 3. Program Execution

Execution in Windows Terminal:

```
BULK PURCHASE TICKETS ---------- B
SELL TICKET -------------------- C
SELL ALL TICKETS --------------- D
VIEW TICKET DETAILS ------------ E
VIEW ALL TICKETS --------------- F
VIEW USER STATUS --------------- G
EXIT --------------------------- Q
>>F

    <LIST OF ALL TICKETS>
>>

    <MAIN OPTIONS>
PURCHASE TICKET ---------------- A
BULK PURCHASE TICKETS ---------- B
SELL TICKET -------------------- C
SELL ALL TICKETS --------------- D
VIEW TICKET DETAILS ------------ E
VIEW ALL TICKETS --------------- F
VIEW USER STATUS --------------- G
EXIT --------------------------- Q
>>B

    <TICKET BULK PURCHASE PROMPT>
ENTER THE PREFERRED NUMBER OF TICKETS: 100
ENTER THE PREFERRED NUMBER OF AUTO TICKETS: 100
GENERATING 100 AUTO TICKET(S), 0 MANUAL TICKET(S)
PURCHASED 100 TICKET(S) FOR $500
USER NOW HAS $917403
>>
```