Program 1: TwentyFour

CS 211, UIC Spring 2023

Description

24 is a math game, in which the player is presented with 4 numbers and they must determine a way to combine those numbers using mathematical operations to get 24 as the result. Each number must be used, and each number can be used only once.

As an example, if the user is presented with the card shown below:

they may come up with the following solution (which may not be the only one!):
5 x 5 = 25
4 – 3 = 1
25 – 1 = 24
See https://www.24game.com/t-about-howtoplay.aspx for more information.

You will be creating a C program to allow the user to play a simplified version of this game. The user can only use addition (+), subtraction (-), multiplication (*), and division (/). The four numbers presented in each puzzle are all single digits (numbers 1-9). The user does not need to use all four mathematical operations.

The calculation for the result is done in sequential order on the numbers, which are also used in order. For instance, if the numbers are 3, 9, 4, 1 and the user selects the three operations as +, -, * then the calculation is done as follows:
3 + 9 = 12
12 - 4 = 8
8 * 1 = 8
(This is not a solution of course; it is just meant to demonstrate the order in which the calculations are performed.)

Running the Program
The program begins by displaying a message to introduce the game.

A puzzle is randomly chosen and the four numbers are presented to the user. After reading in the operators that the user chooses, the program performs the calculation. Note that:

- If the user enters some number of operators other than three, an error message is displayed, and a new puzzle is presented. The error message is "Error! The number of operators is incorrect. Please try again."
- If the user enters an operator that is not allowed (something other than +, -, *, or /), an error message is displayed, and a new puzzle is presented. The error message is "Error! Invalid operator entered. Please try again."
- The number of operators entered is checked before the program checks for invalid operators. (See the last screenshot of sample output below for an example of what happens when the user input does not meet either condition.)

After performing the calculation, the program will print "Well done! You are a math genius." if the result is indeed 24. Otherwise, the program will print "Sorry. Your solution did not evaluate to 24." Either way, the user will then be asked if they would like to play again. When the program ends, the message "Thanks for playing!" is displayed.

<u>Debug Mode</u>
By default, debug mode will be *off*. To turn it *on*, the user will need to use a command-line argument. The parameter will be as follows:

```
     [-d debugMode] = Whether or not to turn on debug display (1
for on / 0 for off)
```

If debug mode is *on*, then the program should find and print all solutions to the game, using numbers 1-9 and each mathematical operation (+, -, *, /). There are 3188 solutions.

To get your output to match the expected output when debug mode is on, take note of the following:

- Consider combinations of numbers before combinations of operators. Taking note of the order of solutions in the output will help here.
- Always test operations in this order: +, -, *, /.
- Use the type `double` when performing the calculations to see if the result is 24. While the real game presents numbers that have integer results each step of the way, this program simplifies it to allow for fractional results. This means that, for instance, `1/2*6*8` is a valid solution.

For the full list of solutions, see the `solutions.txt` file on Blackboard. **This cannot be hard-coded - the solutions must be generated during the execution of your program. If they are hard-coded, your submission will have a score of zero.**

Easy Mode

By default, the program will be in *easy mode*. This means that the puzzle selected is randomly chosen from one of the following 10 puzzles:

- 3, 9, 4, 1
- 8, 5, 8, 1
- 6, 1, 5, 1
- 2, 8, 7, 8
- 5, 2, 9, 2
- 2, 6, 8, 4
- 5, 5, 4, 3
- 6, 6, 2, 6
- 8, 4, 2, 6
- 6, 2, 8, 1

Remember that the order of the numbers matters because of the way in which the calculation is performed.

To turn off easy mode, the user will need to use a command-line argument. The parameter will be as follows:

```
     [-e easyMode] = Whether or not easy mode should be on (1
for on / 0 for off)
```

If easy mode is *off*, then the random puzzle will be chosen from the 3188 possible solutions.

Notes

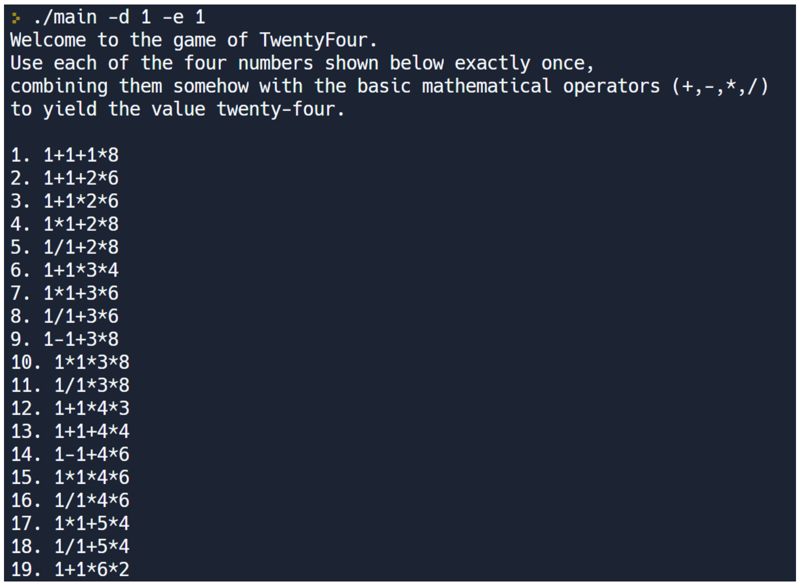
- Be sure to have srand(1) just once, at the beginning of main(), so that the output of your program will match the expected output.
- The starter code available in Replit demonstrates how to use rand() to generate random numbers, and the % (mod) operator to get random values within a range.
- For generating all the solutions, I would suggest that you first figure out how to generate all possible combinations of two single-digit numbers and mathematical operations that give a result of 24. Build on this to generate combinations of three numbers, then finally four numbers.

Sample Output

Below is sample output of playing the game a few times (default of debug mode *off* and easy mode *on*):

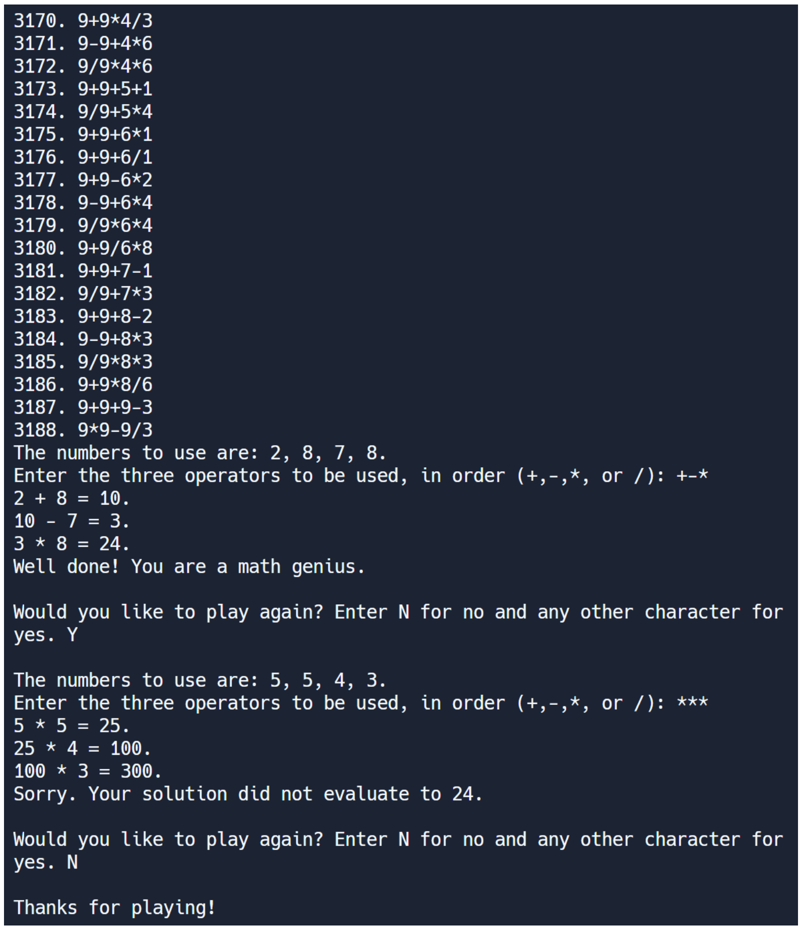
```
Welcome to the game of TwentyFour.
Use each of the four numbers shown below exactly once,
combining them somehow with the basic mathematical operators (+,-,*,/)
to yield the value twenty-four.

The numbers to use are: 2, 8, 7, 8.
Enter the three operators to be used, in order (+,-,*, or /): +-*
2 + 8 = 10.
10 - 7 = 3.
3 * 8 = 24.
Well done! You are a math genius.

Would you like to play again? Enter N for no and any other character for
yes. Y

The numbers to use are: 5, 5, 4, 3.
Enter the three operators to be used, in order (+,-,*, or /): *+-
5 * 5 = 25.
25 + 4 = 29.
29 - 3 = 26.
Sorry. Your solution did not evaluate to 24.

Would you like to play again? Enter N for no and any other character for
yes. Y

The numbers to use are: 6, 6, 2, 6.
Enter the three operators to be used, in order (+,-,*, or /): *-)
Error! Invalid operator entered. Please try again.

The numbers to use are: 2, 6, 8, 4.
Enter the three operators to be used, in order (+,-,*, or /): ++++
Error! The number of operators is incorrect. Please try again.

The numbers to use are: 2, 8, 7, 8.
Enter the three operators to be used, in order (+,-,*, or /): -+*
2 - 8 = -6.
-6 + 7 = 1.
1 * 8 = 8.
Sorry. Your solution did not evaluate to 24.

Would you like to play again? Enter N for no and any other character for
yes. N

Thanks for playing!
```

Below is sample output of playing the game when debug mode is *on* and easy mode is *on* as well (full output is not shown due to lack of space):



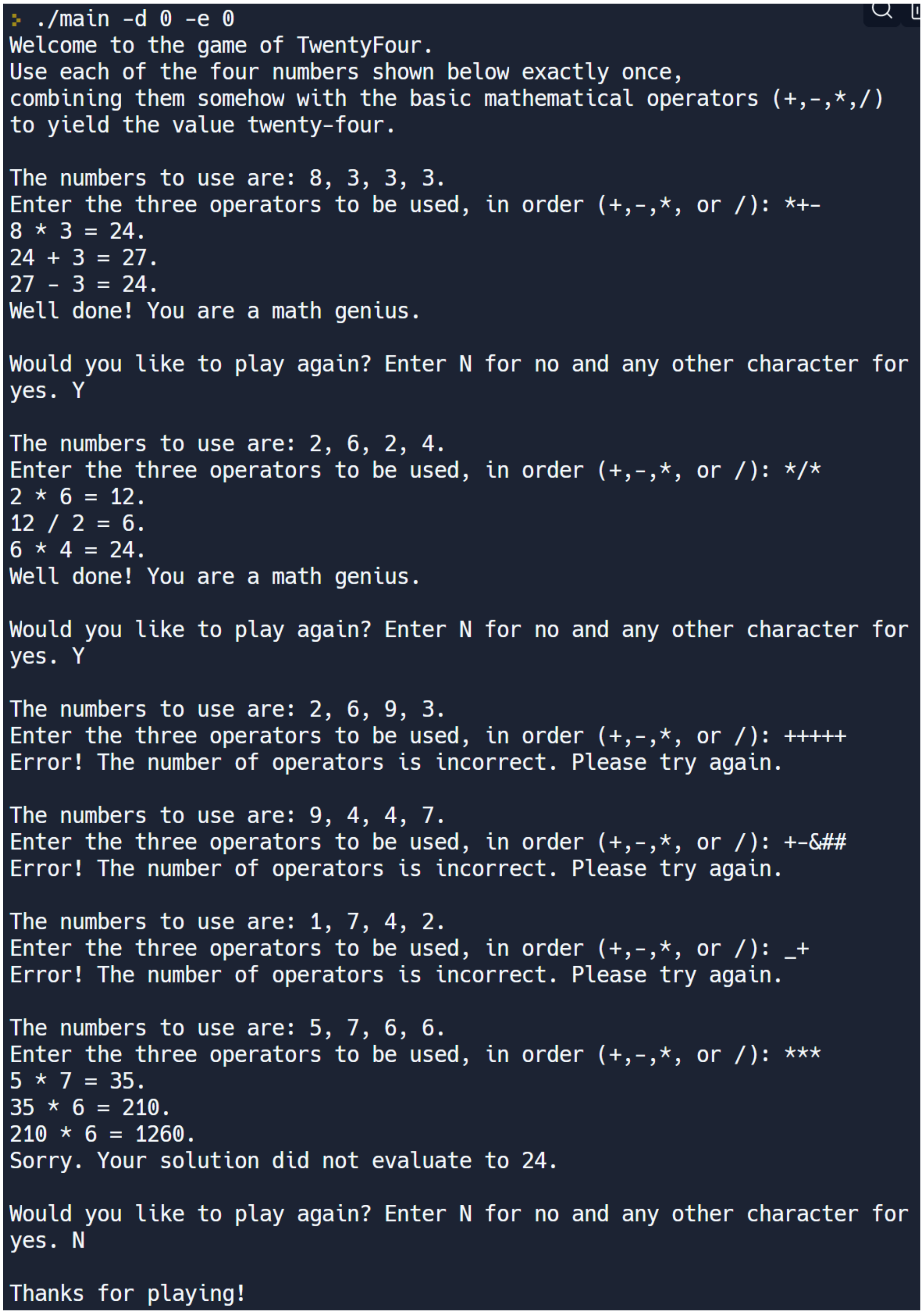
```
❯ ./main -d 1 -e 1
Welcome to the game of TwentyFour.
Use each of the four numbers shown below exactly once,
combining them somehow with the basic mathematical operators (+,-,*,/)
to yield the value twenty-four.

1. 1+1+1*8
2. 1+1+2*6
3. 1+1*2*6
4. 1*1+2*8
5. 1/1+2*8
6. 1+1*3*4
7. 1*1+3*6
8. 1/1+3*6
9. 1-1+3*8
10. 1*1*3*8
11. 1/1*3*8
12. 1+1*4*3
13. 1+1+4*4
14. 1-1+4*6
15. 1*1*4*6
16. 1/1*4*6
17. 1*1+5*4
18. 1/1+5*4
19. 1+1*6*2
```

■ ■ ■



```
3170. 9+9*4/3
3171. 9-9+4*6
3172. 9/9*4*6
3173. 9+9+5+1
3174. 9/9+5*4
3175. 9+9+6*1
3176. 9+9+6/1
3177. 9+9-6*2
3178. 9-9+6*4
3179. 9/9*6*4
3180. 9+9/6*8
3181. 9+9+7-1
3182. 9/9+7*3
3183. 9+9+8-2
3184. 9-9+8*3
3185. 9/9*8*3
3186. 9+9*8/6
3187. 9+9+9-3
3188. 9*9-9/3
The numbers to use are: 2, 8, 7, 8.
Enter the three operators to be used, in order (+,-,*, or /): +-*
2 + 8 = 10.
10 - 7 = 3.
3 * 8 = 24.
Well done! You are a math genius.

Would you like to play again? Enter N for no and any other character for
yes. Y

The numbers to use are: 5, 5, 4, 3.
Enter the three operators to be used, in order (+,-,*, or /): ***
5 * 5 = 25.
25 * 4 = 100.
100 * 3 = 300.
Sorry. Your solution did not evaluate to 24.

Would you like to play again? Enter N for no and any other character for
yes. N

Thanks for playing!
```

Below is sample output of playing the game when debug mode is *off* and easy mode is *off* too:



```
> ./main -d 0 -e 0
Welcome to the game of TwentyFour.
Use each of the four numbers shown below exactly once,
combining them somehow with the basic mathematical operators (+,-,*,/)
to yield the value twenty-four.

The numbers to use are: 8, 3, 3, 3.
Enter the three operators to be used, in order (+,-,*, or /): *+-
8 * 3 = 24.
24 + 3 = 27.
27 - 3 = 24.
Well done! You are a math genius.

Would you like to play again? Enter N for no and any other character for
yes. Y

The numbers to use are: 2, 6, 2, 4.
Enter the three operators to be used, in order (+,-,*, or /): */*
2 * 6 = 12.
12 / 2 = 6.
6 * 4 = 24.
Well done! You are a math genius.

Would you like to play again? Enter N for no and any other character for
yes. Y

The numbers to use are: 2, 6, 9, 3.
Enter the three operators to be used, in order (+,-,*, or /): +++++
Error! The number of operators is incorrect. Please try again.

The numbers to use are: 9, 4, 4, 7.
Enter the three operators to be used, in order (+,-,*, or /): +-&##
Error! The number of operators is incorrect. Please try again.

The numbers to use are: 1, 7, 4, 2.
Enter the three operators to be used, in order (+,-,*, or /): _+
Error! The number of operators is incorrect. Please try again.

The numbers to use are: 5, 7, 6, 6.
Enter the three operators to be used, in order (+,-,*, or /): ***
5 * 7 = 35.
35 * 6 = 210.
210 * 6 = 1260.
Sorry. Your solution did not evaluate to 24.

Would you like to play again? Enter N for no and any other character for
yes. N

Thanks for playing!
```