

CSE 102 Assignment VI

In this program, I implemented a simple number-guessing game called “Vault Codebreaker.” The game has two main modes: Admin Mode and Player Mode.

The first part of the code is Admin Mode. In this mode, the user (admin) enters the game settings. These include the length of the secret code, the digit range (minimum and maximum), whether duplicate digits are allowed, the maximum number of guesses the player can make, and the point values for correct guesses. All these settings are saved into a file named vault_config.txt.

The second part is Player Mode. In this mode, the game reads the configuration from the vault_config.txt file. Then it generates a secret code using the generateCode() function. This function allocates memory dynamically and generates random digits according to the rules. The secret code is also written to vault_code.txt for record keeping.

After the code is generated, the game starts a guessing loop. In each round, the player is asked to input a numeric guess. The getGuess() function is used for this. It only uses dynamic memory allocation and accepts the user’s guess as a single integer. The input is then separated into digits using a helper function called numToDigits().

The guess is compared to the secret code using the compareCodes() function. This function returns an array indicating whether each digit is correct and in the right position (C), correct but in the wrong position (M), or wrong (W). This comparison is logged into game_log.txt after every attempt. The program follows the instructor’s instruction: if a number appears multiple times in the guess but only once in the code, it should count as 1 correct and the rest as misplaced.

The feedback and score are updated after each attempt. Points are added or subtracted according to the values defined in the config. If the player guesses all digits correctly, the game ends with a win message. If the player uses all attempts, the game ends with a loss message.

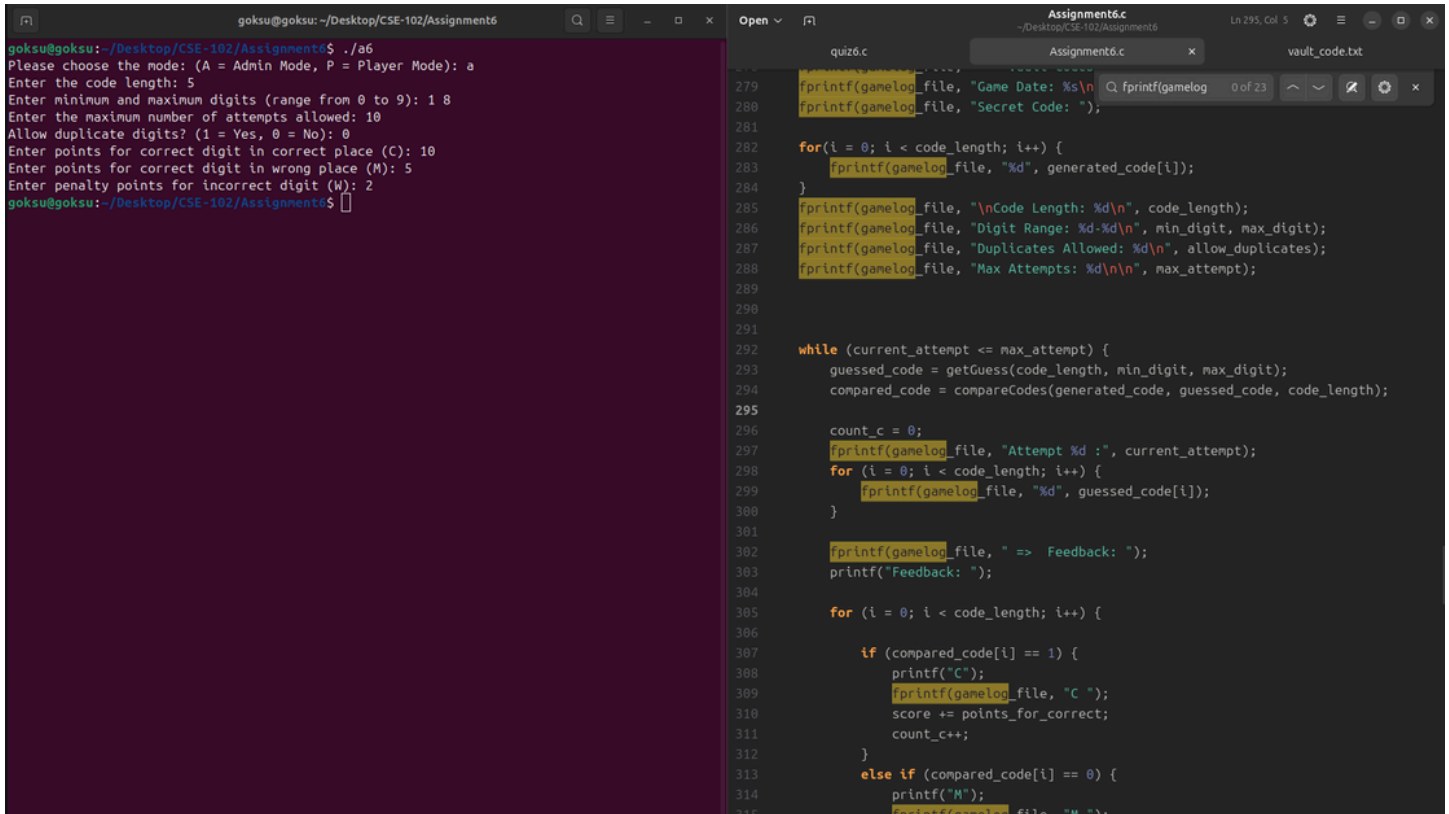
At the end of the game, the final score is printed, and a rank title is assigned based on the score (e.g., “Code Master,” “Cipher Hunter,” etc.). All guesses, feedback, and final results are also saved in the game_log.txt file with a timestamp.

All dynamic memory allocations are freed in the main function or where appropriate. File opening and closing operations are safely checked with error handling.

In my opinion, the hardest part of the code was managing dynamic memory without using any string functions. Making the getGuess() function work correctly using only integers was challenging, but I was able to solve it with a digit-splitting logic.

This program fully satisfies the assignment requirements and uses clean code structure and memory-safe techniques.

● Set Config in Admin Mode

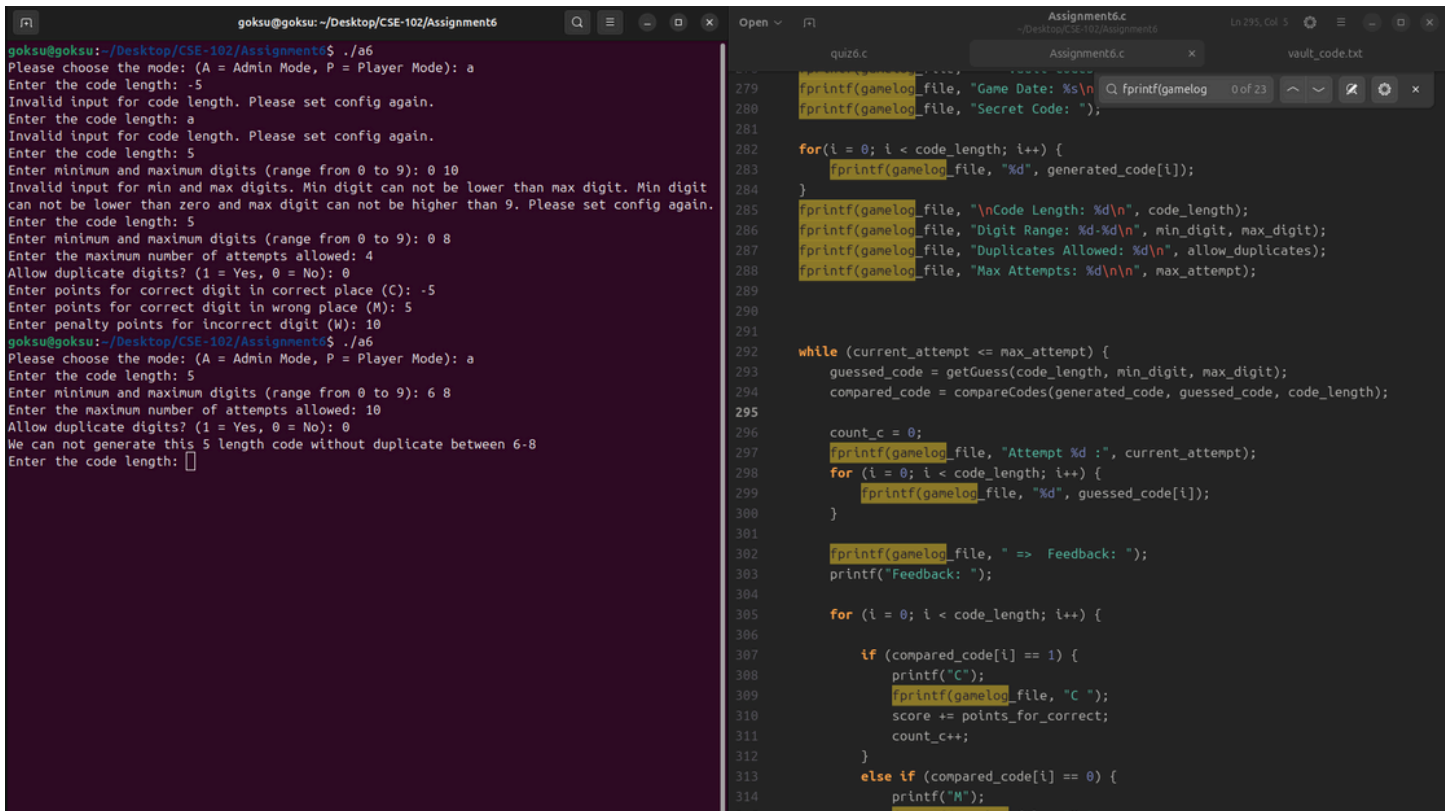


The image shows a terminal window on the left and a code editor on the right. The terminal window displays the execution of the program in Admin Mode, where the user sets various configuration parameters. The code editor shows the corresponding C code that implements these configurations.

```
goksu@goksu: ~/Desktop/CSE-102/Assignment6
goksu@goksu:~/Desktop/CSE-102/Assignment6$ ./a6
Please choose the mode: (A = Admin Mode, P = Player Mode): a
Enter the code length: 5
Enter minimum and maximum digits (range from 0 to 9): 1 8
Enter the maximum number of attempts allowed: 10
Allow duplicate digits? (1 = Yes, 0 = No): 0
Enter points for correct digit in correct place (C): 10
Enter points for correct digit in wrong place (M): 5
Enter penalty points for incorrect digit (W): 2
goksu@goksu:~/Desktop/CSE-102/Assignment6$
```

```
Assignment6.c
279 fprintf(gamelog_file, "Game Date: %s\n", date);
280 fprintf(gamelog_file, "Secret Code: ");
281
282 for(i = 0; i < code_length; i++) {
283     fprintf(gamelog_file, "%d", generated_code[i]);
284 }
285
286 fprintf(gamelog_file, "\nCode Length: %d\n", code_length);
287 fprintf(gamelog_file, "Digit Range: %d-%d\n", min_digit, max_digit);
288 fprintf(gamelog_file, "Duplicates Allowed: %d\n", allow_duplicates);
289 fprintf(gamelog_file, "Max Attempts: %d\n", max_attempt);
290
291
292 while (current_attempt <= max_attempt) {
293     guessed_code = getGuess(code_length, min_digit, max_digit);
294     compared_code = compareCodes(generated_code, guessed_code, code_length);
295
296     count_c = 0;
297     fprintf(gamelog_file, "Attempt %d :", current_attempt);
298     for (i = 0; i < code_length; i++) {
299         fprintf(gamelog_file, "%d", guessed_code[i]);
300     }
301
302     fprintf(gamelog_file, " => Feedback: ");
303     printf("Feedback: ");
304
305     for (i = 0; i < code_length; i++) {
306
307         if (compared_code[i] == 1) {
308             printf("C");
309             fprintf(gamelog_file, "C ");
310             score += points_for_correct;
311             count_c++;
312         }
313         else if (compared_code[i] == 0) {
314             printf("M");
315             fprintf(gamelog_file, "M ");
316         }
317     }
318 }
```

● Some Error outputs in Admin Mode.



The image shows a terminal window on the left and a code editor on the right. The terminal window displays the execution of the program in Admin Mode, where the user sets various configuration parameters. The code editor shows the corresponding C code that implements these configurations.

```
goksu@goksu: ~/Desktop/CSE-102/Assignment6
goksu@goksu:~/Desktop/CSE-102/Assignment6$ ./a6
Please choose the mode: (A = Admin Mode, P = Player Mode): a
Enter the code length: -5
Invalid input for code length. Please set config again.
Enter the code length: a
Invalid input for code length. Please set config again.
Enter the code length: 5
Enter minimum and maximum digits (range from 0 to 9): 0 10
Invalid input for min and max digits. Min digit can not be lower than zero and max digit can not be higher than 9. Please set config again.
Enter the code length: 5
Enter minimum and maximum digits (range from 0 to 9): 0 8
Enter the maximum number of attempts allowed: 4
Allow duplicate digits? (1 = Yes, 0 = No): 0
Enter points for correct digit in correct place (C): -5
Enter points for correct digit in wrong place (M): 5
Enter penalty points for incorrect digit (W): 10
goksu@goksu:~/Desktop/CSE-102/Assignment6$ ./a6
Please choose the mode: (A = Admin Mode, P = Player Mode): a
Enter the code length: 5
Enter minimum and maximum digits (range from 0 to 9): 6 8
Enter the maximum number of attempts allowed: 10
Allow duplicate digits? (1 = Yes, 0 = No): 0
We can not generate this 5 length code without duplicate between 6-8
Enter the code length: 
```

```
Assignment6.c
279 fprintf(gamelog_file, "Game Date: %s\n", date);
280 fprintf(gamelog_file, "Secret Code: ");
281
282 for(i = 0; i < code_length; i++) {
283     fprintf(gamelog_file, "%d", generated_code[i]);
284 }
285
286 fprintf(gamelog_file, "\nCode Length: %d\n", code_length);
287 fprintf(gamelog_file, "Digit Range: %d-%d\n", min_digit, max_digit);
288 fprintf(gamelog_file, "Duplicates Allowed: %d\n", allow_duplicates);
289 fprintf(gamelog_file, "Max Attempts: %d\n", max_attempt);
290
291
292 while (current_attempt <= max_attempt) {
293     guessed_code = getGuess(code_length, min_digit, max_digit);
294     compared_code = compareCodes(generated_code, guessed_code, code_length);
295
296     count_c = 0;
297     fprintf(gamelog_file, "Attempt %d :", current_attempt);
298     for (i = 0; i < code_length; i++) {
299         fprintf(gamelog_file, "%d", guessed_code[i]);
300     }
301
302     fprintf(gamelog_file, " => Feedback: ");
303     printf("Feedback: ");
304
305     for (i = 0; i < code_length; i++) {
306
307         if (compared_code[i] == 1) {
308             printf("C");
309             fprintf(gamelog_file, "C ");
310             score += points_for_correct;
311             count_c++;
312         }
313         else if (compared_code[i] == 0) {
314             printf("M");
315             fprintf(gamelog_file, "M ");
316         }
317     }
318 }
```

● Win the game

```
goksu@goksu: ~/Desktop/CSE-102/Assignment6
goksu@goksu:~/Desktop/CSE-102/Assignment6$ ./a6
Please choose the mode: (A = Admin Mode, P = Player Mode): p
Enter your guess (4-digit number): 4523
Feedback: MWCM
Enter your guess (4-digit number): 6721
Feedback: WWCW
Enter your guess (4-digit number): 1824
Feedback: CWCM
Enter your guess (4-digit number): 1427
Feedback: CWCW
Enter your guess (4-digit number): 1420
Feedback: CCCW
Enter your guess (4-digit number): 1429
Feedback: CCCC

Congratulations! You've guessed the secret code!
The secret code was: 1429
Your final score is: 141
Rank: Code Master
goksu@goksu:~/Desktop/CSE-102/Assignment6$
```

```
Assignment6.c
279 fprintf(gamelog_file, "Game Date: %s\n", game_date);
280 fprintf(gamelog_file, "Secret Code: ");
281
282 for(i = 0; i < code_length; i++) {
283     fprintf(gamelog_file, "%d", generated_code[i]);
284 }
285 fprintf(gamelog_file, "\nCode Length: %d\n", code_length);
286 fprintf(gamelog_file, "Digit Range: %d-%d\n", min_digit, max_digit);
287 fprintf(gamelog_file, "Duplicates Allowed: %d\n", allow_duplicates);
288 fprintf(gamelog_file, "Max Attempts: %d\n", max_attempt);
289
290
291
292 while (current_attempt <= max_attempt) {
293     guessed_code = getGuess(code_length, min_digit, max_digit);
294     compared_code = compareCodes(generated_code, guessed_code, code_length);
295
296     count_c = 0;
297     fprintf(gamelog_file, "Attempt %d :", current_attempt);
298     for (i = 0; i < code_length; i++) {
299         fprintf(gamelog_file, "%d", guessed_code[i]);
300     }
301
302     fprintf(gamelog_file, " => Feedback: ");
303     printf("Feedback: ");
304
305     for (i = 0; i < code_length; i++) {
306
307         if (compared_code[i] == 1) {
308             printf("C");
309             fprintf(gamelog_file, "C ");
310             score += points_for_correct;
311             count_c++;
312         }
313         else if (compared_code[i] == 0) {
314             printf("M");
315             fprintf(gamelog_file, "M ");
316         }
317     }
318 }
```

• Lose the game

```
goksu@goksu: ~/Desktop/CSE-102/Assignment6
goksu@goksu:~/Desktop/CSE-102/Assignment6$ ./a6
Please choose the mode: (A = Admin Mode, P = Player Mode): p
Enter your guess (5-digit number): 35487
Feedback: MCMWM
Enter your guess (5-digit number): 34567
Feedback: MMMWM
Enter your guess (5-digit number): 24215
Feedback: WMMWM
Enter your guess (5-digit number): 35412
Feedback: MCMWM
Enter your guess (5-digit number): 45461
Feedback: MCMWC
Game Over. You've used all your attempts.
The secret code was: 75341
Your final score is: 103
Rank: Code Master
goksu@goksu:~/Desktop/CSE-102/Assignment6$
```

```
Assignment6.c
279 fprintf(gamelog_file, "Game Date: %s\n", game_date);
280 fprintf(gamelog_file, "Secret Code: ");
281
282 for(i = 0; i < code_length; i++) {
283     fprintf(gamelog_file, "%d", generated_code[i]);
284 }
285 fprintf(gamelog_file, "\nCode Length: %d\n", code_length);
286 fprintf(gamelog_file, "Digit Range: %d-%d\n", min_digit, max_digit);
287 fprintf(gamelog_file, "Duplicates Allowed: %d\n", allow_duplicates);
288 fprintf(gamelog_file, "Max Attempts: %d\n", max_attempt);
289
290
291
292 while (current_attempt <= max_attempt) {
293     guessed_code = getGuess(code_length, min_digit, max_digit);
294     compared_code = compareCodes(generated_code, guessed_code, code_length);
295
296     count_c = 0;
297     fprintf(gamelog_file, "Attempt %d :", current_attempt);
298     for (i = 0; i < code_length; i++) {
299         fprintf(gamelog_file, "%d", guessed_code[i]);
300     }
301
302     fprintf(gamelog_file, " => Feedback: ");
303     printf("Feedback: ");
304
305     for (i = 0; i < code_length; i++) {
306
307         if (compared_code[i] == 1) {
308             printf("C");
309             fprintf(gamelog_file, "C ");
310             score += points_for_correct;
311             count_c++;
312         }
313         else if (compared_code[i] == 0) {
314             printf("M");
315             fprintf(gamelog_file, "M ");
316         }
317     }
318 }
```

• vault_config.txt

The terminal window on the left shows the execution of a program named `a6`. The user is prompted to choose a mode (A for Admin, P for Player) and enters 'p'. They then enter five 5-digit guesses: 35487, 34567, 24215, 35412, and 45461. The program provides feedback for each guess (e.g., 'MCMWM', 'MMWMM') and eventually reveals the secret code as 75341. The final score is 103, and the user is ranked 'Code Master'.

The file editor on the right shows the contents of `vault_config.txt`, which defines game parameters:

```
1 CODE_LENGTH=5
2 DIGIT_MIN=1
3 DIGIT_MAX=8
4 MAX_ATTEMPTS=5
5 ALLOW_DUPLICATES=0
6 POINTS_CORRECT=10
7 POINTS_MISPLACED=5
8 PENALTY_WRONG=2
```

- **vault_code.txt.**

The terminal window on the left is identical to the one in the previous screenshot, showing the same game execution and results.

The file editor on the right now shows the contents of `vault_code.txt`, which contains the secret code:

```
1 75341
```

- **game_log.txt**

```
goksu@goksu: ~/Desktop/CSE-102/Assignment6
goksu@goksu:~/Desktop/CSE-102/Assignment6$ ./a6
Please choose the mode: (A = Admin Mode, P = Player Mode): p
Enter your guess (5-digit number): 35487
Feedback: MCMWM
Enter your guess (5-digit number): 34567
Feedback: MMMWM
Enter your guess (5-digit number): 24215
Feedback: WMMWM
Enter your guess (5-digit number): 35412
Feedback: MCMWM
Enter your guess (5-digit number): 45461
Feedback: MCMWC
Game Over. You've used all your attempts.
The secret code was: 75341
Your final score is: 103
Rank: Code Master
goksu@goksu:~/Desktop/CSE-102/Assignment6$
```

```
game_log.txt
~/Desktop/CSE-102/Assignment6
Ln 1, Col 1
Open
quiz6.c Assignment6.c vault_code.txt game_log.txt x vault_config.txt

1 --- Vault Codebreaker Game Log ---
2 Game Date: 2025-04-18 13:20:26
3 Secret Code: 75341
4 Code Length: 5
5 Digit Range: 1-8
6 Duplicates Allowed: 0
7 Max Attempts: 5
8
9 Attempt 1 :35487 => Feedback: M C M W M | Score: 23
10 Attempt 2 :34567 => Feedback: M M M W M | Score: 41
11 Attempt 3 :24215 => Feedback: W M W M M | Score: 52
12 Attempt 4 :35412 => Feedback: M C M M W | Score: 75
13 Attempt 5 :45461 => Feedback: M C M W C | Score: 103
14
15 Final Score: 103
16 Rank: Code Master
17 -----
```