Yasmine Siala

Professor Veenstra

CSE13S

28 October 2024

Assignment 2 Draft

is_lowercase_letter()

1. Purpose: the purpose of the function is to determine whether a character is lowercase
2. Parameters: char c
3. Return value: true or false
4. Pseudocode: Check if char c is between ‘a’ and ‘z’ and if ‘a’ <= c <= ‘z’ then return true, otherwise return false

string_contains_character()

1. Purpose: to check whether a given string contains a specific character
2. Parameters: const char *string (pointer for the string), char ch
3. Return value: true/false
4. Pseudocode: loop through each character in the string via for loop, if the current character is equal to ch, return true; if at the end of the loop ch is not found then return false

is_valid_secret()

1. Purpose: to determine if the given string meets the criteria to be valid
2. Parameters: const char *secret (pointer to the secret string)
3. Return value: true/false
4. Pseudocode:

```
- check the length of secret using strlen() - if the length is greater than the max length then print an error
- Use a for loop to loop through each character in secret to check that each character is a) lowercase and b) one of the valid punctuation characters
- If any character is not valid, print an error message
```

prompt_for_and_read_character()

1. Purpose: Prompt the user and return a single character input
2. Parameters: none
3. Return value: the first user-entered valid character or in the case of EOF the function exits and returns 1
4. Pseudocode:

```
- Print a prompt message
- Enter a loop to read through characters:
    - Call getchar() and store in an int
    - If the character is \n continue to the next iteration
    - If the character is EOF, terminate the program
```

run_hangman()

1. Purpose: Run a hangman game using a user-given secret word
2. Parameters: const char *secret
3. Return value: void
4. Pseudocode:

```
- Initialize guesses_left, phrase_display, eliminated_letters, and has_won
- Setup phrase_display based on secret
```

   - Print the initial gallows, phrase_dispkay, “Eliminated:”, and “Guess a letter:”
   - Prompt for a letter using ‘prompt_for_and_read_character()’
   - Check if the guessed letter is already in eliminated_letters or phrase_display, then reprint “Guess a letter:”
   - If the guessed letter is in ‘secret’: update pphrase_display to reveal all positions of the guessed letter
   - If phrase_display matches secret, set has_won to true and break the loop
   - If the guessed letter is not in secret add it to eliminated_letters, decrease guesses_left by 1, and display the updated gallows
   - If has_won is true print “You win!” and print the complete phrase
   - If guesses_left == 0, print “You lose!” and print the complete phrase

main()

1. Purpose: Set up and validate initial conditions for the hangman game
2. Parameters: argc(int), argv(char**)
3. Return value: 1 if there’s an error, 0 if execution is successful
4. Pseudocode:
   - Verify the number of command-line arguments
     - If argc != 2, print an error message and return 1
   - Retrieve the secret from argv[1]
   - Validate the secret using is_valid_secret()
     - If is_valid_secret() return false, return 1
   - Call run_hangman(secret)
   - Return 0