1. Problem Analysis

Golems in the City of Ankh Morpork are out of control and require a special spell to control them again. The spell must consist of four possible elements in the spell: Fire(F), Water(W), Air(A), and Earth(E). The spell caster has 10 chances to guess the correct spell and duplicates of the same element are allowed. For every correct element in the correct spot the golems eyes will flow Red(R) and for every correct element in the incorrect spot the golems eyes will flow Blue(B).

1. Problem Definition Table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Known Facts** | **User Requirements** | **Necessary Processing** | **Alternative Solutions** |
| -10 chances to guess correct spell  -Spell will be generated randomly when program starts.  -Spell must consist of F W A or E  -4, 6, 8 elements in spell depending on difficulty level.  -Output “R” for correct element in correct spot  -Output “B” for correct element in incorrect spot.  -Output guesses left and current attempt  -Output motivation message when has incorrect combination  -Output congratulations message when guess correct spell. | -Input chosen difficulty level  -Input guess for spell. # of elements depend on difficulty level. | Declare variables:  -int for # of guesses  -string for difficulty  -string array for possible elements  -string array for random generated elements  -int for number of elements depending on difficulty level/times to loop each for loop.  -bool to check if user’s chosen difficulty is valid  -string to read user’s guess  -bool to check if the user inputted the correct spell  -do while or while loop to get user to choose a difficulty level  -for loop to generate the random spell combination in the array  -create a temp string array with the same generated letters for the “B” outputs  -outer do while loop that loops until the user enters the correct answer or exceeds 10 guesses  -inner do while loop that loops until the user’s answer is valid  -if statement that checks to see if the correct number of elements is entered  -for loop that checks every element in the array and if element is invalid let user know which element is invalid  -if user’s guess is correct exit loop  -else for loop check to see which elements are right and in the right spot and output “R” replace them with a “&” in the inputted array and “#” in temp array  -another for loop nested inside of another for loop to check for elements that are right but in the wrong spot and output “B” and replace that element in the temp array with the Boolean false.  -for loop that resets the temp array to the original generated array for the next round of guessing.  -if user guess is wrong output motivational message  -if user guess right output congratulations message  -if user guess exceeds 10 output sad message. | Declare variables: -int for # of guesses  -string for difficulty  -string array for possible elements  -string array for random generated elements  -int for number of elements depending on difficulty level/times to loop each for loop.  -bool to check if user’s chosen difficulty is valid  -string to read user’s guess  -bool to check if the user inputted the correct spell  -do while or while loop to get user to choose a difficulty level  -for loop to generate the random spell combination in the array  -if statements to check if user inputted correct spell and “R” for elements correct in correct spot and “B” for elements correct but in wrong spot.  -outer do while loop that loops until the user enters the correct answer or exceeds 10 guesses  -inner do while loop that loops until the user’s answer is valid  -if user’s guess is correct exit loop  -if user guess is wrong output motivational message  -if user guess right output congratulations message  -if user guess exceeds 10 output sad message. |

1. IPO Chart:

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| -Difficulty level  -Spell guess | -check to see if difficulty is valid  -generate a random spell  -check to see if spell has correct letters in correct places  -check to see if spell has correct letters in different places | -Messages to input valid inputs  -“R” for correct elements in correct spot  -“B” for correct elements in incorrect spots  -Congratulations message if correct spell guess  -Sad message if exceeds 10 guesses |

1. Psuedo-code

Declaration:

-int for # of guesses

-string for difficulty

-string array for possible elements

-string array for random generated elements

-int for number of elements depending on difficulty level/times to loop each for loop.

-bool to check if user’s chosen difficulty is valid

-string to read user’s guess

-bool to check if the user inputted the correct spell

Ask user to input difficulty until difficulty bool = true

If difficulty == valid

Proceed to ask user to guess spell

Difficulty bool = true

Else if difficulty == invalid

Keep asking user to input valid guess

Difficulty bool = false

Do{

Ask user to input a guess until valid bool = true

If guess > or < # of elements

Ask user to input valid number of elements

Valid bool = false

Else if guess == invalid elements

Ask user to input valid elements

Valid bool = false

If guess == generated spell

Exit loop and output congratulations message

Correct booelean =true

Else if guess != generated spell but correct elements in correct spot

For loop checks and:

Output “R” for every correct element in correct spot

Replace correct elements in inputted array with “&”

Replace correct elements in temp array with “#”

Correct Boolean = false

Else if guess != generated spell but correct elements in incorrect spot

For loop within a for loop checks and:

Output “B” for every correct element in incorrect spot

Replaces every matched element in temp array with false Boolean

Correct Boolean = false

For loop resets temp array to original generated array

Output # of guess left if guess != generated spell

Add 1 to guess attempts

}loop until correct boolean == true or guess exceeds 10

If guess == generated spell

Output congratulations message

Else if guess attempts exceed 10

Output sad message

Testing:

~summary of guessing test cases (since there are too many guessing test cases):

I tested that when the user enters the wrong guess but there is an element in the right spot it will output “R” and when the guess is also wrong but there is an element that is right in the wrong spot it will output “B”. I inputted the same guess again to ensure the “R” and “B” reset after each guess.

(All user input is converted to capitals for checking efficiency)

*Choosing Difficulty Level:*

**Test Case 1(not entering 1 of the 3 Mastery levels):**

Input:

Cheesecake

Output:

The Mastery you have entered seems to be invalid.

What Mastery do you possess young padawan?:

**Test Case 2(Entering “beginner”):**

Input:

Beginner

Output:

Guess num: 1

What spell shall you cast?(4 Elements in your spell):

**Test Case 3(Entering “average”):**

Input:

average

Output:

Guess num: 1

What spell shall you cast?(6 Elements in your spell):

**Test Case 4(Entering “master”):**

Input:

master

Output:

Guess num: 1

What spell shall you cast?(8 Elements in your spell):

*Casting the spell:*

**Test Case 1(if the user enters more characters or less than necessary):**

Beginner difficulty:

More than necessary

Input:

aaaaa

Output:

Please enter 4 elements

Less than necessary

Input:

aaa

Output:

Please enter 4 elements

Average difficulty:

More than necessary

Input:

aaaaaaa

Output:

Please enter 4 elements

Less than necessary

Input:

aaaaa

Master difficulty:

More than necessary

Input:

aaaaaaaaa

Output:

Please enter 4 elements

Less than necessary

Input:

Aaaaaaa

**Test Case 2(Beginner all right):**

Sample generated spell: FAAE

Input:  
FAAE

Output:

The City Of Ankh Morpork thanks you and your BEGINNER mastery. We will dispose of Adora to prevent this from happening again

**Test Case 3(Beginner 1 Right)**

Sample generated spell: FAAE

Input:  
FWWW

Output:

R

**Test Case 3(Beginner 2 Right)**

Sample generated spell: FAAE

Input:  
FAWW

Output:

RR

**Test Case 4(Beginner 3 Right)**

Sample generated spell: FAAE

Input:  
FAAW

Output:

RRR

**Test Case 5(Beginner 0 Right)**

Sample generated spell: FAAE

Input:  
WWWW

Output:

**Test Case 6(Beginner All Right But Wrong Order)**

Sample generated spell: FAAE

Input:  
AFEA

Output:

BBBB

**Test Case 7(Beginner 1 Right But Wrong Order)**

Sample generated spell: FAAE

Input:  
WWWF

Output:

B

**Test Case 8(Beginner 2 Right But Wrong Order)**

Sample generated spell: FAAE

Input:  
WWEF

Output:

BB

**Test Case 9(Beginner 3 Right But Wrong Order)**

Sample generated spell: FAAE

Input:  
EFFA

Output:

BBB

**Test Case 10(Beginner 1 Right But Rest Right And Wrong Order)**

Sample generated spell: FWAE

Input:  
FAEW

Output:

RBBB

**Test Case 11(Beginner 2 Right But Rest Right And Wrong Order)**

Sample generated spell: FWAE

Input:  
FWEA

Output:

RRBB

**Test Case 12(None right)**

Sample generated spell: FWWW

Input:  
AAAA

Output:

(nothing)

Average Difficulty

**Test Case 1(Average 1 Right)**

Sample generated spell: WEEFFF

Input:  
WAAAAA

Output:

R

**Test Case 2(Average 2 Right)**

Sample generated spell: WEEFFF

Input:  
WEAAAA

Output:

RR

**Test Case 3(Average 3 Right)**

Sample generated spell: WEEFFF

Input:  
WEEAAA

Output:

RRR

**Test Case 4(Average 4 Right)**

Sample generated spell: WEEFFF

Input:  
WEEFAA

Output:

RRRR

**Test Case 5(Average 5 Right)**

Sample generated spell: WEEFFF

Input:  
WEEFFA

Output:

RRRRR

**Test Case 6(Average All Right)**

Sample generated spell: WEEFFF

Input:  
WEEFFF

Output:

RRRRRR

**Test Case 7(Average 0 Right)**

Sample generated spell: WEEFFF

Input:  
AAAAAA

Output:

**Test Case 8(Average 1 Right But In Wrong Order)**

Sample generated spell: WEEFFF

Input:  
AAAAAW

Output:

B

**Test Case 9(Average 2 Right But In Wrong Order)**

Sample generated spell: WEEFFF

Input:  
AAAAEW

Output:

BB

**Test Case 10(Average 3 Right But In Wrong Order)**

Sample generated spell: WEEFFF

Input:  
AAAEEW

Output:

BBB

**Test Case 11(Average 4 Right But In Wrong Order)**

Sample generated spell: WEEFFF

Input:  
AAFEEW

Output:

BBBB

**Test Case 12(Average 5 Right But In Wrong Order)**

Sample generated spell: WEEFFF

Input:  
AFFEEW

Output:

BBBBB

**Test Case 13(Average All Right But In Wrong Order)**

Sample generated spell: WEEFFF

Input:  
FFFEEW

Output:

BBBBBB

**Test Case 14(Average 1 Right But Rest In Wrong Order)**

Sample generated spell: WEEFFA

Input:  
WFFEAE

Output:

RBBBBB

**Test Case 15(Average 2 Right But Rest In Wrong Order)**

Sample generated spell: WEEFFA

Input:  
WEFEAF

Output:

RRBBBB

**Test Case 16(Average 4 Right But Rest In Wrong Order)**

Sample generated spell: FAWEWW

Input:  
FAWWEW

Output:

RRRRBB

**Test Case 17(None right)**

Sample generated spell: FWWWWW

Input:  
AAAAAA

Output:

(nothing)

Master Difficulty:

**Test Case 1(All Right)**

Sample generated spell: AAAEEFEE

Input:  
AAAEEFEE

Output:

RRRRRRRR

**Test Case 2 (7 Right)**

Sample generated spell: AAAEEFEE

Input:  
AAAEEFEW

Output:

RRRRRRR

**Test Case 3 (6 Right)**

Sample generated spell: AAAEEFEE

Input:  
AAAEEFWW

Output:

RRRRRR

**Test Case 4 (5 Right)**

Sample generated spell: AAAEEFEE

Input:  
AAAEEWWW

Output:

RRRRR

**Test Case 5 (4 Right)**

Sample generated spell: AAAEEFEE

Input:  
AAAEWWWW

Output:

RRRR

**Test Case 6 (3 Right)**

Sample generated spell: AAAEEFEE

Input:  
AAAWWWWW

Output:

RRR

**Test Case 7 (2 Right)**

Sample generated spell: AAAEEFEE

Input:  
AAWWWWWW

Output:

RR

**Test Case 8 (1 Right)**

Sample generated spell: AAAEEFEE

Input:  
AWWWWWWW

Output:

R

**Test Case 8 (All Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
EEEEFFFF

Output:

BBBBBBBB

**Test Case 9 (1 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
EWWWWWWW

Output:

B

**Test Case 10 (2 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
EEWWWWWW

Output:

BB

**Test Case 11 (3 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
EEEWWWWW

Output:

BBB

**Test Case 12 (4 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
EEEEWWWW

Output:

BBBB

**Test Case 13 (5 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
EEEEFWWW

Output:

BBBBB

**Test Case 14 (6 Right But Wrong Order)**

Sample generated spell: FFFFEEEA

Input:  
EEEEAWWE

Output:

BBBBBB

**Test Case 15 (7 Right But Wrong Order)**

Sample generated spell: FFFFEEEA

Input:  
EEEEWAWE

Output:

BBBBBBB

**Test Case 16 (1 Right And 1 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FEWWWWWW

Output:

RB

**Test Case 16 (2 Right And 1 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FEFWWWWW

Output:

RRB

**Test Case 17 (3 Right And 1 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FEFFWWWW

Output:

RRRB

**Test Case 18 (4 Right And 1 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FEFFEWWW

Output:

RRRRB

**Test Case 19 (5 Right And 1 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FEFFEEWW

Output:

RRRRRB

**Test Case 20 (6 Right And 1 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FEFFEEEW

Output:

RRRRRRB

**Test Case 21 (1 Right And 2 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FEEWWWWW

Output:

RBB

**Test Case 22 (1 Right And 3 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FEEEWWWW

Output:

RBBB

**Test Case 23 (1 Right And 4 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FEEEFWWW

Output:

RBBBB

**Test Case 24 (1 Right And 5 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FEEEFFWW

Output:

RBBBBB

**Test Case 25 (1 Right And 6 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FEEEFFFW

Output:

RBBBBBB

**Test Case 26 (2 Right And 6 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FEEEFFFE

Output:

RRBBBBBB

**Test Case 27 (2 Right And 5 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FFEEFFFW

Output:

RRBBBBB

**Test Case 28 (2 Right And 4 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FFEEFFWW

Output:

RRBBBB

**Test Case 29 (2 Right And 4 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FFEEFFWW

Output:

RRBBBB

**Test Case 30 (2 Right And 3 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FFEEFWW

Output:

RRBBB

**Test Case 31(4 Right And 4 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FFEEFFEE

Output:

RRRRBBBB

**Test Case 32(4 Right And 2Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FFEEEEWW

Output:

RRRRBB

**Test Case 32(5 Right And 2 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FFEWEEEF

Output:

RRRRRBB

**Test Case 33 (4 Right And 3 Right But Wrong Order)**

Sample generated spell: FFFFEAAF

Input:  
FFFFAWEA

Output:

RRRRBBB

**Test Case 34 (6 Right And 2 Right But Wrong Order)**

Sample generated spell: FFFFEEEE

Input:  
FFEFEFEE

Output:

RRRRRRBB

**Test Case 35 (None right)**

Sample generated spell: FWWWWWWW

Input:  
AAAAAAAA

Output:

(nothing)