

String Craft

Objective:

The objective of this game is to create the highest scoring string possible using legal moves on a given input string of uppercase letters. But be careful with your moves. Moves cost you points.

Legal Moves:

Characters can be moved by swapping the positions of two characters in the string. For example, in the string "EXAMPLE", the last E could be swapped with the X, producing the string "EEAMPLX", which would change the score from 7 to 8. Moves that lower the score are not allowed.

points - cost = score

E X A M P L E

1 1 1 1 1 1 1 7 - 0 = 7

E E A M P L X

1 3 1 1 1 1 1 9 - 1 = 8

Point Cost:

Each move costs 1 point.

Game Over:

The game is over when no more legal moves are possible.

Scoring:

- Each letter has a base value of 1
- Each consecutive matching letter is worth 2 more than the point value of the letter before it.
- Each letter that is part of a run is worth 1 more than the previous letter in the run.
 - A run is a set of letters that are each exactly 1 letter apart in the alphabet and are in alphabetical order, such as DEFGHIJ.
 - For the purpose of determining a run, the letter A is considered to follow Z. So another possible run would be XYZABC.

Problem

Write a program that does the following:

- 1) Prompt the user for a string length
- 2) Generates a random string of that length made up of only uppercase letters.
- 3) Prints out the string followed by the score for that string.
- 4) If there are valid moves, ask the user to enter two numbers separated by a space to indicate the indexes of two characters to swap, otherwise print the final score and say game over.
- 5) If the move is legal, the program should print out the resulting string and the new score, otherwise it should print a message saying the move was illegal and prompt the user again.
- 6) Keeps repeating steps 4 and 5 until there are no more moves at which point it prints the final score and announces the game is over.

Challenge

Write a program that, given an input string, returns the maximum possible score achievable by playing this game.

Example Game:

Pts - Cost = Score

A B D K F K H G Z A D N N
 1 2 1 1 1 1 1 1 1 2 1 1 3 17 - 0 = 17

A B A K F K H G Z D D N N
 1 2 1 1 1 1 1 1 1 1 3 1 3 18 - 1 = 17

B A A K F K H G Z D D N N
 1 1 3 1 1 1 1 1 1 1 3 1 3 19 - 2 = 17

K A A B F K H G Z D D N N
 1 1 3 4 1 1 1 1 1 1 3 1 3 22 - 3 = 19

Z A A B F K H G K D D N N
 1 2 4 5 1 1 1 1 1 1 3 1 3 25 - 4 = 21

Z A A B F G H K K D D N N
 1 2 4 5 1 2 3 1 3 1 3 1 3 30 - 5 = 25

no more moves possible

Sample Run:

Welcome to String Craft

Enter the length of the string you would like to craft: 13

ABDKFKHKGZADNN is worth 17 points. Your score is 17.

Enter the indexes of the characters you would like to swap: 10 11

That move is illegal because it would lower your score to 14

ABDKFKHKGZADNN is worth 17 points. Your score is 17.

Enter the indexes of the characters you would like to swap: 2 9

ABAKFKHKGZDDNN is worth 18 points. Your score is 17.

Enter the indexes of the characters you would like to swap: 0 1

BAAKFKHKGZDDNN is worth 19 points. Your score is 17.

Enter the indexes of the characters you would like to swap: 0 3

KAABFKHKGZDDNN is worth 22 points. Your score is 19.

Enter the indexes of the characters you would like to swap: 0 8

ZAABFKHKGKDDNN is worth 25 points. Your score is 21.

Enter the indexes of the characters you would like to swap: 5 7

ZAABFGHKKDDNN is worth 30 points. Your score is 25.

Game Over! Your final score is 25 points.

After reading the rules for String Craft, determine the String worth before/after the move and whether the move is legal.

- Ex.: Swap index 2 and 3: ABEC → ABCE String value: 5 → 7 **Legal** / Illegal
1. Swap index 1 and 2: AEBD → ABED String value: _____ → _____ Legal / Illegal
2. Swap index 0 and 1: EBEE → BEEE String value: _____ → _____ Legal / Illegal
3. Swap index 1 and 3: AAAB → ABAA String value: _____ → _____ Legal / Illegal
4. Swap index 2 and 3: CADZ → CAZD String value: _____ → _____ Legal / Illegal
5. Swap index 1 and 3: BZAC → BCAZ String value: _____ → _____ Legal / Illegal

Next, see if you can get a high score for the word ZSVBESYAMZ.

Word: Z S V B E S Y A M Z
score: _ _ _ _ _ = _____pts

Move 1: _ _ _ _ _
score: _ _ _ _ _ = _____ - 1 = _____pts

Move 2: _ _ _ _ _
score: _ _ _ _ _ = _____ - 2 = _____pts

Move 3: _ _ _ _ _
score: _ _ _ _ _ = _____ - 3 = _____pts

Move 4: _ _ _ _ _
score: _ _ _ _ _ = _____ - 4 = _____pts

Move 5: _ _ _ _ _
score: _ _ _ _ _ = _____ - 5 = _____pts

String Craft Top Down Approach

Using top-down design (i.e. use helper methods as if they already exist) write the main game logic for String Craft in 30 or fewer lines. All helper methods should return values instead of relying on class attributes.

- 1) Prompt the user for a string length
- 2) Generate a random string of that length made up of only uppercase letters.
- 3) Print out the string followed by the score for that string.
- 4) If there are valid moves, ask the user to enter two numbers separated by a space to indicate the indexes of two characters to swap, otherwise print the final score and say game over.
- 5) If the move is legal, print out the resulting string and the new score, otherwise print a message saying the move was illegal and prompt the user again.
- 6) Keep repeating steps 4 and 5 until there are no more moves at which point you should print the final score and announce the game is over.

```
int strLen = promptForStringLength();
String randString = generateRandomString(strLen);
sopl(randString + " is worth " + wordValue(randString) + " points");
```

Choose two non-trivial helper methods and do the following for each method:

- Specify the method header (i.e. return type, method name, parameter list)
- Plan your strategy to accomplish the method using pseudocode
- Write the actual code
- Generate and run a battery of test cases that proves your method works independently under all conditions.

Method 1

1. Method header:

2. Strategy / pseudocode

3. Code

4. Write down a list of test cases including expected output. Make sure they cover all scenarios.

Method 2

1. Method header:

2. Strategy / pseudocode

3. Code

4. Write down a list of test cases including expected output. Make sure they cover all scenarios.