String manipulation is a popular problem in computer science. One of the important problems of string manipulation is rearranging a string in the form of integer sum followed by the minimized character. In this post, we'll go through how to rearrange a string so that it will return integer sum along with the reduced character.

In the question, a string is given which consists of lowercase alphabets with some digits. We need to rearrange the string in such a way that the new string begins with the integer sum of all the digits in the old string and ends with the minimized character in the new string.

**Note:** If no digit is present in the given string add 0 to the new string.

Let’s try to understand with a few examples.

**Example 1:**

Suppose, the input string is “ab37b3a8”. We have to first get the sum of all the numeric digits in this string. So the integer sum is (3 + 7 + 3 + 8) which is equal to 21. Then, We have to sum all alphabets and minimize it to a single character. So the sum of alphabets is (a + b + b + a) = (1 + 2 + 2 + 1) = 6. Find the remainder of 6 with 26 i.e. 6. The minimized character in this example is the alphabet at 6th position i.e. ‘f’.

So, the output of this example is “21f”.

**Example 2:**

Suppose, the input string is “ayush”. We have to first get the sum of all the numeric digits in this string. In this example no numeric digit is present so the integer sum is equal to 0. Then, We have to sum all alphabets and minimize it to a single character. So the sum of alphabets is (a + y + u + s + h) = (1 + 25 + 21 + 19 + 8) = 74. Find the remainder of 74 with 26 i.e. 22. The minimized character in this example is the alphabet at 22th position i.e. ‘v’.

So, the output of this example is “0v”.

Hope it's enough to understand this problem. Now, discuss how to code this problem.

## **Approach 1: Simple Approach**

### **Algorithm**

**Step 1:** Create a function with the name separateChar which takes the input string.

**Step 2:** Using a for loop, traverse the input string character by character.

**Step 3:** If the character is a digit, then sum all the digits and store it in a variable.

**Step 4:** Similarly, if the character is an alphabet, then sum all the alphabets str[i]-‘a’+1 and store it in a variable.

**Step 5:** Convert the sum of integer and sum of alphabets to string and concatenate it in a new string.

**Step 6:** Return the final string.

### **Explanation**

In this approach, a function separateChar executes which accepts user input, and outputs the results to the console. The technique is simple and effective. it just makes one iteration around the input string and uses constant time operations throughout.

### **Program**

|  |
| --- |
| #include <iostream>  #include <string>  using namespace std;  string separateChar(string str)  {  int n = str.size(), digitSum = 0;  int alphabetSum = 0, j = 0;  for (int i = 0; i < n; i++) {  if (isdigit(str[i]))  digitSum += str[i] - '0';  else {  alphabetSum += str[i] - 'a' + 1;  alphabetSum %= 26;  }  }  string sumStr = to\_string(digitSum);  char alphabetStr = char(alphabetSum + 'a' - 1);  sumStr += alphabetStr;  return sumStr;  }  int main()  {  string str;  cout << "Enter the string: ";  cin >> str;  cout << separateChar(str);  return 0;  } |

**Output**

|  |
| --- |
| Enter the string: ab37b3a8  21f |

## **Approach 2: Recursive Approach**

### **Algorithm**

**Step 1:** Create a function separateChar with two helper functions digitSum and alphabetSum.

**Step 2:** digitSum function finds the sum of all the digits in the input string and returns the sum.

**Step 3:** alphabetSum function finds the sum of all the alphabets in the given string and returns the sum.

**Step 4:** Then we convert the sums into strings and concatenate it into a new string.

**Step 5:** Return the final string.

### **Explanation**

This approach uses the recursive approach. In this method, a function separateChar executes with its two helper functions to get the job done. The code is more modular and readable when helper functions are used.

### **Program**

|  |
| --- |
| #include <iostream>  #include <string>  using namespace std;  int digitSum(string str, int i)  {  if (i == str.size()) {  return 0;  }  if (isdigit(str[i])) {  return (str[i] - '0') + digitSum(str, i+1);  }  return digitSum(str, i+1);  }  int alphabetSum(string str, int i)  {  if (i == str.size()) {  return 0;  }  if (isalpha(str[i])) {  return (tolower(str[i]) - 'a' + 1) + alphabetSum(str, i+1);  }  return alphabetSum(str, i+1);  }  string separateChar(string str)  {  int dsum = digitSum(str, 0);  int asum = alphabetSum(str, 0) % 26;    string sumStr = to\_string(dsum) + char(asum + 'a' - 1);  return sumStr;  }  int main()  {  string str;  cout << "Enter the string: ";  cin >> str;  cout << separateChar(str);  return 0;  } |

**Output**

|  |
| --- |
| Enter the string: ab37b3a8  21f |

## **Conclusion**

In this article, we understand this problem with the best explanation and examples. We also found its solution with code implementation in c++. Basic string manipulation techniques can be used to rearrange a string in the form of integer sum followed by the minimized character.