

4 주차

알 고 리 즈 과 제

ToBig's 9기 김수지

Algorithm 2nd

문자열 처리 중심으로

Algorithm 2nd | Introduction

과제 소개

- 이번 알고리즘 과제는 문자열 처리와 관련된 문제 2문제이며 문제는 다음 페이지에 있습니다.

유의사항

- 다음 두 문제는 **R을 제외한** 어떤 언어로든 풀이하셔도 됩니다. (C, C++, Java, Python...)
- 데이터마켓에 과제 업로드 할 때는 '**코드 복붙+결과 캡처**' 를 올려주세요. 코드를 첨부 할 필요는 없습니다!

Question 01 | 빈출 알파벳 찾기

1. 빈출 알파벳 찾기

대소문자가 섞인 문자열이 주어지면, 이 문자열에서 가장 많이 등장한 알파벳이 무엇인지 알아내는 문제입니다.
단, 알파벳의 대, 소문자는 구분하지 않습니다.

입력

문자열이 주어지되, 그 문자열의 길이는 1,000,000을 넘지 않습니다.

출력

가장 많이 사용된 알파벳을 대문자로 출력합니다.

단, 가장 많이 사용된 알파벳이 여러 개 존재하는 경우에는 ??를 출력합니다.

입력 예시	출력 예시
Mississippi	??
zZa	Z
baaa	A

Question 01 | 빈출 알파벳 찾기

Python

```
word = input( ).upper( )
count = [i for i in range(0,26)]

for i in range(0,26):
    count[i] = word.count(chr(i+65))

maxindex = max(count)

if count.count(maxindex) >= 2:
    print('?')
else:
    print(chr(count.index(maxindex)+65))
```

Question 01 | 빈출 알파벳 찾기

C

```
#include <stdio.h>
int main(void)
{
    int max = 0, length;
    char word[1000000];
    int count[26] = { 0 }; //크기 26 배열이 모두 0으로 초기화
    char answer;

    scanf("%s", word);
    length = strlen(word);

    for (int i = 0; i < length; i++) //모든 문자를 대문자로 변경
    {
        word[i] = toupper(word[i]);
    }

    for (int i = 0; i < length; i++)
    { //word의 처음부터 계산하여 A를 뺀 index에 count를 증가
        count[word[i] - 'A']++;
    }
```

```
for (int i = 0; i < 26; i++)
{
    if (max < count[i])
    {
        answer = (char)(i + 'A');
        max = count[i];
    }
    else if (max == count[i]) {
        answer = '?';
    }
}

printf("%c ", answer);
return 0;
}
```

Question 01 | 빈출 알파벳 찾기

Java

```
import java.util.*;

public class maxWord {

    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);

        String word;
        int max=0;
        char answer='?';

        int[] count = new int[26];

        word = s.nextLine();
        word = word.toUpperCase();

        for(int i=0;i<word.length();i++) {
            count[word.charAt(i)-65]++;
        }
```

```
    }
}
```

```
        for(int i=0;i<26;i++) {
            if(max<count[i]) {
                answer=(char)(i+65);
                max=count[i];
            }
            else if(max==count[i])
                answer='?';
        }
        System.out.println(answer);

        s.close();
```

Question 02 | 문자열 반복 출력

2. 문자열 반복 출력

문자열 S 를 입력 받은 후에, 각 문자를 R 번 반복해 새로운 문자열 T 를 출력합니다.

입력

첫째 줄에 테스트 케이스의 개수가 주어집니다. ($1 \leq T \leq 1000$)

각 테스트 케이스는 반복 횟수와 ($1 \leq R \leq 8$) 문자열 S 가 공백으로 구분되어 주어집니다.

S 의 길이는 적어도 1이며, 20글자를 넘지 않습니다.

출력

각 테스트 케이스에 대해 T 를 출력합니다.

입력 예시	출력 예시
2	
3 ABC	AAABBBCCC
5 /HTP	/////HHHHHTTTTTPPPPP

Question 02 | 문자열 반복 출력

Python

```
t = int(input( ))  
for t in range(t) :  
    r, s = input( ).split( )  
    print(' '.join(x*int(r) for x in s))
```


Question 02 | 문자열 반복 출력

C

```
#include <stdio.h>
int main(t, r, i, j) {
    char s[21];

    for (scanf("%d", &t); t--;) {          // 테스트 케이스 t를 입력받아서 t번 반복한다
        scanf("%d %s", &r, s);           // 반복 횟수 r과 문자열 s를 입력받는다.
        for (i = 0; s[i] != 0; i++)       // 문자열이 null이 아닐 때까지 반복
            for (j = 0; j < r; j++)
                printf("%c", s[i]);
        printf("\n");                     // 문자열 반복 출력이 끝나면 줄바꿈. for문 영향 안받음
    }
}
```

Question 02 | 문자열 반복 출력

Java

```
import java.util.Scanner;
public class groupWord {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);

        int t = s.nextInt();

        for(int i = 0; i<t; i++){
            int r = s.nextInt();
            char[] arr = s.next().toCharArray();

            StringBuilder sb = new StringBuilder();

            for(char c : arr){
                for(int j = 0; j<r; j++){
                    sb.append(c);
                }
            }
            System.out.println(sb.toString());
        }
    }
}
```