sentence reversal

```
#include <stdio.h>
#include <string.h>
#include <stdbool.h>

void flip(char *b, int k, int l)
{
  int tmp;
  for ( ; k<l ; k++, l--) {
    tmp = b[k];
    b[k] = b[l];
    b[l] = tmp;
  }
}

void sentence_reversal(char *a)
{
  int i, j;
  j = 0;
  for (i=0; a[i]!='\0'; i++) {
    if (a[i] == ' ') {
      flip(a, j, i-1);
      j = i+1;
    }

  }
  flip(a, 0, strlen(a)-2);
 }


void dump(char *a) {

    bool first_word = true;
    bool wording = false;
    int sentence_length = strlen(a);
    for (int j=0; j<sentence_length; j++) {
      if (a[j] == ' ') {
        wording = false;
      }
      else {
        // i.e. a[i] != ' '

        if (!wording) {
          if (first_word) {
            first_word = false;
          }
          else {
            printf(" ");
          }
        }

        printf("%c", a[j]);
        wording = true;
      }
    }
    printf("\n");
}
```

```
int main(void)
{
  char a[111];
  for(int i=0; i<10; i++){

    if(fgets(a, 99, stdin) != NULL) {
      if(strlen(a) > 1){
        // update aa
        //
        // e.g.,
        //
        // if a = ' xx y zz '
        // then aa = 'xx y zz'
        //
        // if a = 'xx y  zz'
        // then aa = 'xx y zz'
        //
        // if a = 'xx  y  zz'
        // then aa = 'xx y zz'
        //

        a[strlen(a)-1] = 32;
        a[strlen(a)] = 0;
        sentence_reversal(a);
        dump(a);
      }
      else {
        // i.e. a == '\n'
        printf("\n");
      }
    }
  }

  return 0;
}
```

typo

```
#include <stdio.h>
#include <string.h>
#include <stdbool.h>

/*
void dump(char *arr) {
    printf("DEBUG start dumping\n");
    for (int i=0; i<strlen(arr); i++) printf("%d ", arr[i]);
    printf("\n");
}*/

// (1) Niflheimr accidentally types one more character.
// ex. abc abvc
int rule_one_more(char *b, char *c) {
  bool result = true;

  if (strlen(b) == strlen(c)-1) {
    // case 1: xyz xyza --> true
    // case 2: xyz axyz --> true
    // case 3: xyz xayz --> true
    // case 4: xyz abca --> false
    // case 5: xyz xyab --> false

    bool flag = true;
    for(int i=0; i<strlen(c); i++){
      if (flag) {
        if (c[i] == b[i]) continue;
        else flag = false;
      }
      else {
        if (c[i] == b[i-1]) continue;
        else {result = false; break;}
      }
    }

  }
  else result = false;

  return result;
}
// (2) Niflheimr accidentally forgets to type one character.
int rule_one_less(char *b, char *c) {
  bool result = true;

  if(strlen(b) == strlen(c)+1) {
    // case 1: xyz xy --> true
    // case 2: xyz yz --> true
    // case 3: xyz xz --> true
    // case 4: xyz xa --> false
    // case 5: xyz ab --> false

    bool flag = true;
    for(int i=0; i<strlen(b); i++){
      if(flag) {
        if (b[i] == c[i]) continue;
        else flag = false;
      }
      else {
```

```
        if(b[i] == c[i-1]) continue;
        else {result = false; break;}
      }
    }
  }
  else result = false;

  return result;
}
// (3) Niflheimr accidentally types one wrong character.
int rule_one_wrong(char *b, char *c) {
    bool result = false;

    if(strlen(b) == strlen(c)){
        int count = 0;
        for(int i=0; i<strlen(b); i++){
            if(b[i] != c[i]) {
                if (count == 0) {count = 1; result = true;}
                else {result = false; break;}
            }
      }
    }

    return result;
}
// (4) Niflheimr accidentally switches two different adjacent characters.
int rule_switch_two(char *b, char *c) {
    bool result = false;

    if (strlen(b) == strlen(c)) {

        bool found = false;
        for(int i=0; i<strlen(b); i++){
          if (b[i] == c[i]) continue;
          else {
              if (!found && i != strlen(b)-1) {
                  if (b[i] == c[i+1] && b[i+1] == c[i]) {
                      result = true; found = true; i++;
                  }
                  else break;
              }
              else {
                 result = false; break;
              }
          }
        }
    }

    return result;
}

void compare(char *b, char *c) {
    // case: 1232 232 --> typo (failed)
    int rule1 = rule_one_more(b, c);
    int rule2 = rule_one_less(b, c);
    int rule3 = rule_one_wrong(b, c);
    int rule4 = rule_switch_two(b, c);
    if ( rule1 || rule2 || rule3 || rule4) printf("Yes\n");
    else printf("No\n");
  }
```

```
void analysis(char *a) {
    //dump(a);

    char b[200001];
    int index_a = 0;
    for(int j=0; a[j] != 32; j++){
        b[j] = a[j];
        index_a = j;
    }
    b[index_a+1] = 0;
    //dump(b);

    char c[200001];
    index_a +=2;
    int k = 0;
    for(int j=index_a; j<strlen(a)-1; j++){
        c[k++] = a[j];
    }
    c[k] = 0;
    //dump(c);

    compare(b, c);
}

int main(){
  int tcnum;
  scanf("%d\n", &tcnum);
  for(int i=0; i<tcnum; i++) {
    char a[400002];
    if(fgets(a, 99, stdin) != NULL){
      analysis(a);
    }
  }
  return 0;
}
```

Reverse and Compare

```
#include <stdio.h>
#include <string.h>
char a[200000];

void swap(char *x, char *y)
{
    char temp;
    temp = *x;
    *x = *y;
    *y = temp;
}

int main(){

  scanf("%s", &a);

  // case: aatt

  int count = 1;
  for(int i=0; i<strlen(a); i++){
    for(int j=i+1; j<strlen(a); j++){
      if (a[i] == a[j]) continue;
      else count++;
    }
  }
  printf("%d\n", count);

  return 0;
}
```