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

// Function to check if two strings are compatible
int are_compatible(const char* A, const char* B) {
    int lenA = strlen(A);
    int lenB = strlen(B);

    // If the lengths are not equal, they are not compatible
    if (lenA != lenB) {
        return 0;
    }

    // Calculate the difference between the first characters
    int diff = (B[0] - A[0] + 26) % 26;

    // Check if the difference is consistent across all characters
    for (int i = 0; i < lenA; i++) {
        if ((B[i] - A[i] + 26) % 26 != diff) {
            return 0;
        }
    }

    return 1;
}

int main() {
    char A[1000001], B[1000001];

    // Read the input strings
    scanf("%s", A);
```

```c
    // Read the input strings
    scanf("%s", A);
    scanf("%s", B);

    // Check compatibility
    if (are_compatible(A, B)) {
        printf("YES\n");
    } else {
        printf("NO\n");
    }

    return 0;
}
```

	Input	Expected	Got	
✗	abaca cdbda	YES	NO	✗

Some hidden test cases failed, too.

Your code must pass all tests to earn any marks. Try again.

Show differences

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

int main() {
    int N;
    scanf("%d", &N);

    char passwords[100][15];
    for (int i = 0; i < N; i++) {
        scanf("%s", passwords[i]);
    }

    for (int i = 0; i < N; i++) {
        for (int j = i + 1; j < N; j++) {
            if (strcmp(passwords[i], passwords[j]) == 0) continue;

            int len = strlen(passwords[i]);
            int is_palindrome = 1;
            for (int k = 0; k < len; k++) {
                if (passwords[i][k] != passwords[j][len - k - 1]) {
                    is_palindrome = 0;
                    break;
                }
            }

            if (is_palindrome) {
                printf("%d %c\n", len, passwords[i][len / 2]);
                return 0;
            }
        }
    }
```

```
23                  }
24              }
25
26              if (is_palindrome) {
27                  printf("%d %c\n", len, passwords[i][len / 2]);
28                  return 0;
29              }
30          }
31      }
32
33      return 0;
34  }
```

	Input	Expected	Got	
✓	4 abc def feg cba	3 b	3 b	✓

Passed all tests! ✓

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

int main() {
    int N;
    scanf("%d", &N);

    char best_restaurant[21];  // To store the best restaurant's name, max 20 chars +
    int max_points = -1;

    for (int i = 0; i < N; i++) {
        char restaurant_name[21];
        int points;

        // Input the restaurant name and points
        scanf("%s %d", restaurant_name, &points);

        // Determine if we need to update the best restaurant
        if (points > max_points || (points == max_points && strcmp(restaurant_name, be
            max_points = points;
            strcpy(best_restaurant, restaurant_name);
        }
    }

    // Output the best restaurant's name
    printf("%s\n", best_restaurant);

    return 0;
}
```

	Input	Expected	Got	
✓	3 Pizzeria 108 Dominos 145 Pizzapizza 49	Dominos	Dominos	✓

Passed all tests! ✓

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int isValidMobileNumber(char number[]) {
    // Check if the length of the number is 10
    if (strlen(number) != 10) {
        return 0;
    }

    // Check if the number contains only digits and does not start with zero
    if (number[0] == '0') {
        return 0;
    }

    for (int i = 0; i < 10; i++) {
        if (!isdigit(number[i])) {
            return 0;
        }
    }

    return 1;
}

int main() {
    int T;

    scanf("%d", &T);

    char number[20]; // Assuming maximum length of input can be 20
```

```
31
32      for (int i = 0; i < T; i++) {
33
34          scanf("%s", number);
35
36          if (isValidMobileNumber(number)) {
37              printf("YES\n");
38          } else {
39              printf("NO\n");
40          }
41      }
42
43      return 0;
44  }
```

	Input	Expected	Got	
✓	3 1234567890 0123456789 0123456.87	YES NO NO	YES NO NO	✓

Passed all tests! ✓