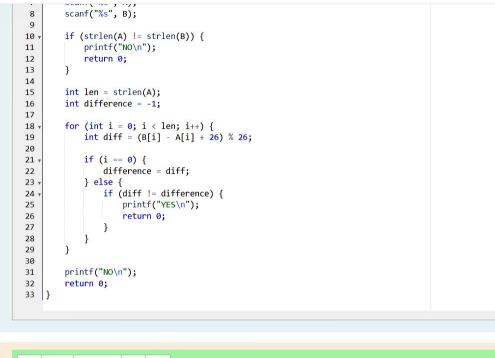


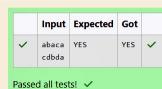
Constraints

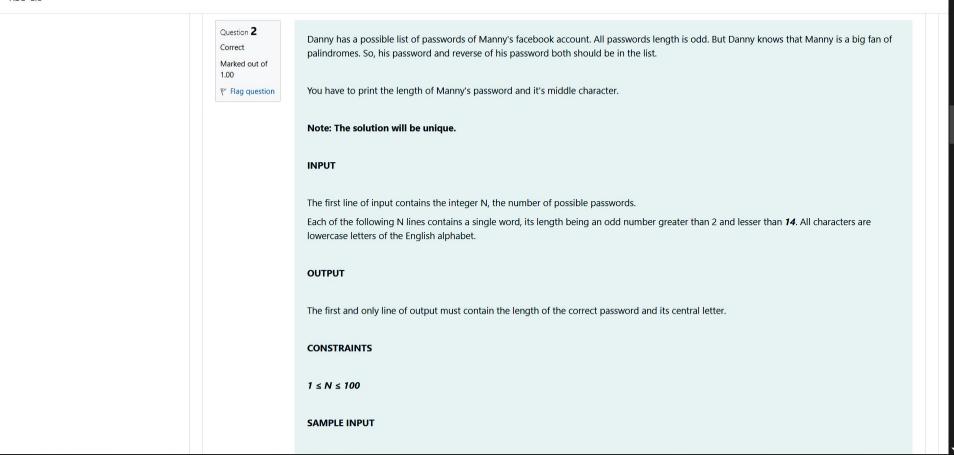


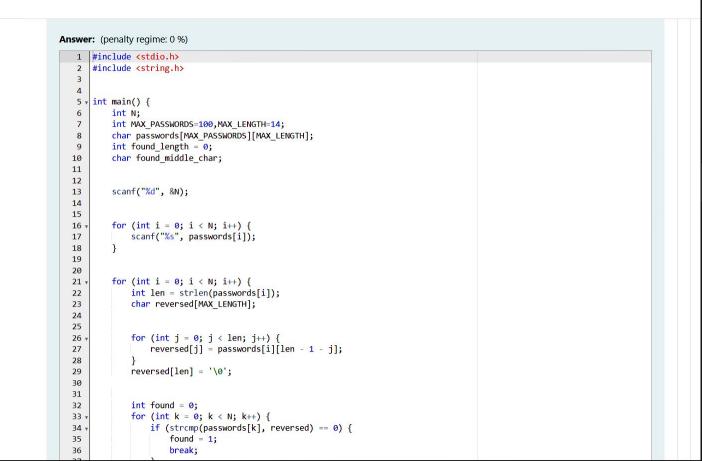
chan A[1000001] D[1000001].

```
Answer: (penalty regime: 0 %)
   1 #include <stdio.h>
      #include <string.h>
   4 v int main() {
          char A[1000001], B[1000001];
          scanf("%s", A);
          scanf("%s", B);
   9
  10
          if (strlen(A) != strlen(B)) {
              printf("NO\n");
  11
  12
              return 0;
  13
  14
          int len = strlen(A);
  15
          int difference = -1;
  16
  17
  18 •
          for (int i = 0; i < len; i++) {
              int diff = (B[i] - A[i] + 26) \% 26;
  19
  20
  21 •
              if (i == 0) {
  22
                  difference = diff;
  23 •
               } else {
                  if (diff != difference) {
  24 •
                      printf("YES\n");
  25
  26
                      return 0;
  27
  28
  29
  30
          printf("NO\n");
  31
  32
          return 0;
  33 }
```









```
15
16
        for (int i = 0; i < N; i++) {
            scanf("%s", passwords[i]);
17
18
19
20
21
       for (int i = 0; i < N; i++) {
22
            int len = strlen(passwords[i]);
23
            char reversed[MAX LENGTH];
24
25
26
            for (int j = 0; j < len; j++) {
27
               reversed[j] = passwords[i][len - 1 - j];
28
            reversed[len] = '\0';
29
30
31
32
            int found = 0;
            for (int k = 0; k < N; k++) {
33 •
                if (strcmp(passwords[k], reversed) == 0) {
34
35
                    found = 1;
36
                    break;
37
38
39
40
            if (found) {
41
                found length = len;
42
43
               found middle char = passwords[i][len / 2];
                break;
44
45
46
47
       printf("%d %c\n", found length, found middle char);
48
49
50
       return 0;
51 }
```

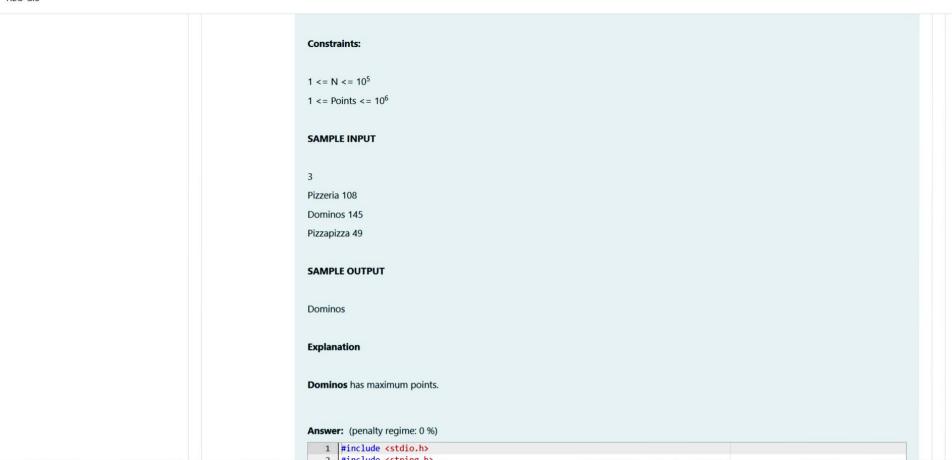
31 32

33 •

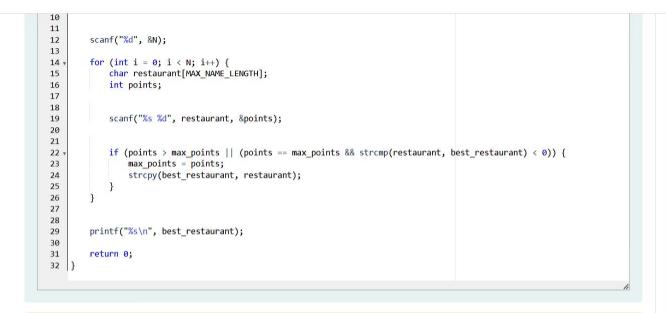
int found = 0;

for (int k = 0; k < N; k++) {





```
Answer: (penalty regime: 0 %)
     #include <stdio.h>
      #include <string.h>
       int main() {
   5 v
          int N;
          int MAX NAME LENGTH=21;
          char best_restaurant[MAX_NAME_LENGTH];
          int max_points = -1;
  10
  11
  12
          scanf("%d", &N);
  13
  14
          for (int i = 0; i < N; i++) {
              char restaurant[MAX NAME LENGTH];
  15
              int points;
  16
  17
  18
              scanf("%s %d", restaurant, &points);
  19
  20
  21
              if (points > max points || (points == max points && strcmp(restaurant, best restaurant) < 0)) {
  22 *
                  max points = points;
  23
                  strcpy(best_restaurant, restaurant);
  24
  25
  26
  27
  28
          printf("%s\n", best_restaurant);
  29
  30
  31
          return 0;
  32 }
```





REC-CIS	
Question 4 Correct Marked out of 1.00	These days Bechan Chacha is depressed because his crush gave him list of mobile number some of them are valid and some of them are invalid. Bechan Chacha has special power that he can pick his crush number only if he has valid set of mobile numbers. Help him to determine the valid numbers.
F Flag question	You are given a string "S" and you have to determine whether it is Valid mobile number or not. Mobile number is valid only if it is of length 10, consists of numeric values and it shouldn't have prefix zeroes.
	Input:
	First line of input is T representing total number of test cases.
	Next T line each representing "S" as described in in problem statement.
	Output:
	Print "YES" if it is valid mobile number else print "NO".
	Note: Quotes are for clarity.
	Constraints:
	$1 <= T <= 10^3$
	sum of string length <= 10 ⁵
	SAMPLE INPUT

REC-CIS	
	Output:
	Print "YES" if it is valid mobile number else print "NO".
	Note: Quotes are for clarity.
	Constraints:
	$1 <= T <= 10^3$
	sum of string length <= 10 ⁵
	SAMPLE INPUT
	3
	1234567890
	0123456789
	0123456.87
	SAMPLE OUTPUT
	YES
	NO
	NO

