Όνομα Άσκησης: Διπλοί Παλίνδρομοι (Dual Palindromes)

Πηγή: Usaco

Εκφώνηση

Παλινδρομικός, ή παλίνδρομος, καλείται ένας αριθμός ο οποίος όταν τα ψηφία του διαβάζονται από το τελευταίο προς το πρώτο προκύπτει πάλι ο ίδιος αριθμός με αυτόν. Π.χ. ο αριθμός πχ. 12321. Οι παλίνδρομοι αριθμοί δεν μπορούν να ξεκινούν, ούτε να καταλήγουν σε μηδέν, έτσι ο αριθμός 0220 δεν είναι παλίνδρομος. Ο αριθμός 21 δεν είναι παλίνδρομος στο δεκαδικό σύστημα, αν όμως μετατραπεί στο δυαδικό (10101) τότε είναι . Να γράψετε πρόγραμμα που να διαβάζει δύο αριθμούς στο δεκαδικό σύστημα:

- N (1 <= N <= 15)
- S (0 < S < 10000)

και να βρίσκει τους πρώτους N αριθμούς αυστηρώς μεγαλύτερους από το S που είναι παλίνδρομοι όταν εκφραστούν σε δύο ή περισσότερες βάσεις B (2 <= B <= 10) όπως π.χ. στο δυαδικό ή στο τριαδικό σύστημα.

Δεδομένα εισόδου (dualpal.in)

Δύο ακέραιοι αριθμοί N και S

Δεδομένα εξόδου (dualpal.out)

N αριθμοί στο δεκαδικό σύστημα (βάση: 10)ταξινομημένοι σε αύξουσα σειρά οι οποίοι είναι παλίνδρομοι όταν εκφραστούν σε δύο ή περισσότερες διαφορετικές βάσεις.

Παράδειγμα εισόδου

```
3 25
```

Παράδειγμα εξόδου

```
26
27
28
```

Επεξήγηση

Χρησιμοποιούμε brute force. Μετατρέπουμε όλους τους αριθμούς μετά το S σε B βάσεις μέχρι να βρούμε N παλίνδρομους.

Hints

Μετατρέψτε τους αριθμούς σε strings.

Λύση

```
#include <fstream>
#include <string>
using namespace std;

string num_to_string(int N, int B) {
  if (N == 0)
    return "0";

  char numbers[] = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  string result = "";

  do {
    result.push_back(numbers[N % B]);
    N /= B;
  } while (N);

  return string(result.rbegin(), result.rend());
}

bool is_palindrome(const string &snum) {
  string r(snum.rbegin(), snum.rend());

  return r == snum;
}

int main() {

  ifstream fin("dualpal.in");
  ofstream fout("dualpal.out");

  int N, S;
  int pal_nums;

  fin >> N >> S;

  for (int i = S + 1; ; i++) {
    pal_nums = 0;
    for (int j = 2; j <= 10; j++) {
      if (is_palindrome(num_to_string(i, j))) {
        pal_nums++;
        if (pal_nums == 2)
          break;
      }
    }

    if (pal_nums >= 2) {
      fout << i << endl;
      N -= 1;
```

```
      if (!N)
        break;
    }
  }

  fin.close();
  fout.close();

  return 0;
}
```