**Complete**

Reward: https://codefights.com/img/coins_new.png **3000**

Given a positive integer, find the next larger integer that can be formed using only the digits from the original number.

**Input (I)** → integer :

**Output** → integer :

Next larger integer with the same digits and -1 if it doesn't exist

<https://codefights.com/feed/LfJ2PXWgnQ3vhqgRY>

[izmmisha](https://codefights.com/profile/izmmisha) 's solution

int p,D,d, nextSameDigits(int I) {

for(d = 1; D=I/d%10; d\*=10, p = D)

if(D < p)

return I+(p-D)\*d\*0.9;

return -1;

}

**----MI SOLUCION ACEPTADA----**

#include <iostream> // std::cout

#include <algorithm> // std::next\_permutation, std::sort

#include <vector>

#include <limits>

int nextSameDigit(int I){

std::vector<int> digits;

int aux = I;

while(aux > 0) {

digits.push\_back(aux%10);

aux/=10;

}

std::reverse(digits.begin(), digits.end());

int dif = INT\_MAX;

int min\_dif = dif;

int next\_large = -1;

do {

//lo convierto a entero

int perm=0;

for(int i = 0; i < digits.size(); i++) {

perm = perm \* 10 + digits[i];

}

int dif = perm - I;

if( perm > I && dif < min\_dif ) {

min\_dif = dif;

next\_large = perm;

}

//printf("\n%d ", perm);

} while(std::next\_permutation( digits.begin(), digits.end()));

return next\_large;

}

int main() {

int next = nextSameDigit(115);

printf("%d ", next);

return 0;

}