

Большое ФБЦ-число. 10-11 класс

В задаче рассмотрим факториальную буквенно-цифровую систему счисления (сокращённо ФБЦ-систему) из задачи "ФБЦ".

Составьте программу, которая принимает на вход в первой строке десятичное число N – положительное натуральное число ($1 \leq N \leq 50000$) – длину последовательности символов, и во второй строке – саму эту последовательность, составленную из N символов из таблицы ASCII. В последовательности не используются управляющие символы, такие как перевод строки и другие. Программа находит максимальное ФБЦ-число, запись которого можно составить из символов последовательности. При составлении записи числа можно использовать символы из введённой последовательности в количестве, которое не больше количества их вхождений во введённую последовательность. Например, если цифра 0 входит в последовательность дважды, то в составленной записи искомого числа 0 встречается не более чем два раза. Нельзя менять регистр символов (т. е. делать строчные буквы заглавными или наоборот).

Программа выводит в единственной строке запись найденного числа в ФБЦ-системе без значащих нулей. Если введённые символы таковы, что из них не получается составить запись какого-либо числа в ФБЦ-системе, то программа выводит единственную строку с числом -1 .

Формат ввода: В первой строке содержится десятичное число N ($1 \leq N \leq 50000$). Во второй строке содержится последовательность, составленная из N символов из таблицы ASCII, но без управляющих символов, таких как перевод строки и другие.

Формат вывода: Если среди введённой последовательности недостаточно символов, подходящих для записи ФБЦ-числа, то программа в первой строке выводит -1 и ничего более не выводит. Иначе в единственной строке выводится максимальное ФБЦ-число, запись которого можно составить из символов последовательности. При составлении записи числа используются символы из введённой последовательности в количестве, которое не больше количества их вхождений во введённую последовательность. В записях ФБЦ-чисел используются только десятичные цифры ($0, \dots, 9$), строчные латинские буквы (a, \dots, z) и заглавные латинские буквы (A, \dots, Z). Выводится запись найденного числа в ФБЦ-системе без значащих нулей.

Пример №1:

ввод:

65

01020304056789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz

вывод:

YXWVUTSRQPONMLKJIHGFEDCBAzyxwvutsrqponmlkjihgfedcba9876543210

Пример №2:

ввод:

2

22

вывод:

-1

Пример №3:

ввод:

15

*aA334422551100

вывод:

a554433221100

Решение

В решении можно запрограммировать следующие подзадачи: 1) определение количества вхождений каждого из символов – цифр и букв, используемых в записи ФБЦ-чисел, $0, \dots, 9, a, \dots, z, A, \dots, Z$ в данную последовательность; 2) определение максимально возможной длины искомого ФБЦ-числа; 3) определение искомого числа при известной его длине. Для решения первой подзадачи достаточно одного прохода по последовательности. Для решения второй подзадачи можно использовать жадный алгоритм. С учётом правил записи ФБЦ-чисел следует суммировать количество вхождений сначала цифры 0, потом 1 и т. д., останавливаясь, если числовое значение цифры превышает текущую сумму более чем на 1. Для решения третьей подзадачи также можно использовать жадный алгоритм. Зная длину искомого числа, начиная со старшего его разряда, следует помещать в разряд цифру, которая является наибольшей среди допустимых правилами

записи ФБЦ-чисел. После использования цифры, количество её вхождений уменьшается на 1. Далее осуществляется переход к соседнему более младшему разряду. По окончании обработки выводится составленное число (или -1 при неудачном поиске).

Код возможного решения

```
program MAXFBC1011 (input, output);
const   MAXN = 61;
type    fbcnumber = array [1..MAXN] of char;
var      ANSWER : fbcnumber;
         N : word;
         DIGITS : array [0..MAXN] of word;
         C : char;
         I, J : byte;
function num2char(I : byte) : char;
begin   if (I >= 0) and (I <= 9) then num2char := chr(I + ord('0'))
        else if (I >= 10) and (I <= 35) then num2char := chr(I - 10 + ord('a'))
        else if (I >= 36) and (I <= MAXN) then num2char := chr(I - 36 + ord('A'))
        else num2char := '/';
end;
function char2num(C : char) : byte;
begin   if (C >= '0') and (C <= '9') then char2num := ord(C) - ord('0')
        else if (C >= 'a') and (C <= 'z') then char2num := ord(C) - ord('a') + 10
        else if (C >= 'A') and (C <= 'Z') then char2num := ord(C) - ord('A') + 36
        else char2num := 255
end;
procedure readsequence(N : word);
var      C : char;
         I : word;
         J : byte;
begin    for I := 1 to N do begin
            read(C);
            J := char2num(C);
            if (J <> 255) then DIGITS[J] := DIGITS[J] + 1
        end;
end;
function calclenght() : word;
var      L, I : word;
         C : char;
begin    L := 0;
         C := '0';
         I := char2num(C);
         while (I <> 255) and ((I - 1) <= L) and (L < MAXN) and (C <= '9') do
             begin L := L + DIGITS[I]; C := succ(C); I := char2num(C) end;
         C := 'a';
         I := char2num(C);
         while (I <> 255) and ((I - 1) <= L) and (L < MAXN) and (C <= 'z') and (C >= 'a') do
             begin L := L + DIGITS[I]; C := succ(C); I := char2num(C) end;
         C := 'A';
         I := char2num(C);
         while (I <> 255) and ((I - 1) <= L) and (L < MAXN) and (C <= 'Z') do
             begin L := L + DIGITS[I]; C := succ(C); I := char2num(C) end;
         if L > MAXN then L := MAXN;
         calclenght := L;
end;
begin    readln(N);
         for I := 0 to MAXN do DIGITS[I] := 0;
```

```

readsequence(N);
J := calclenght();
if J = 0 then writeln(-1)
else begin
  for I := 1 to MAXN do ANSWER[I] := '0';
  I := MAXN;
  while J > 0 do begin
    if I > J then I := J;
    while (I >= 0) and (I <= MAXN) and (DIGITS[I] = 0)
    do I := I - 1;
    if (I >= 0) and (I <= MAXN) and (DIGITS[I] > 0) then begin
      ANSWER[J] := num2char(I);
      DIGITS[I] := DIGITS[I] - 1;
    end;
    J := J - 1
  end;
  I := MAXN;
  while (ANSWER[I] = '0') and (I > 1) do I := I - 1;
  while I >= 1 do begin write(ANSWER[I]); I := I - 1 end;
  writeln
end;
end.

```