

29. Кафе

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Около Петинoгo университета недавно открылось новое кафе, в котором действует следующая система скидок: при каждой покупке более чем на 100 рублей покупатель получает купон, дающий право на один бесплатный обед (при покупке на сумму 100 рублей и меньше такой купон покупатель не получает).

Однажды Пете на глаза попался преЙскурант на ближайшие N дней. Внимательно его изучив, он решил, что будет обедать в этом кафе все N дней, причем каждый день он будет покупать в кафе ровно один обед. Однако стипендия у Пети небольшая, и поэтому он хочет по максимуму использовать предоставляемую систему скидок так, чтобы его суммарные затраты были минимальны. Требуется найти минимально возможную суммарную стоимость обедов и номера дней, в которые Пете следует воспользоваться купонами.

Формат ввода

В первой строке входного файла записано целое число N ($0 \leq N \leq 100$). В каждой из последующих N строк записано одно целое число, обозначающее стоимость обеда в рублях на соответствующий день. Стоимость — неотрицательное целое число, не превосходящее 300.

Формат вывода

В первой строке выдайте минимальную возможную суммарную стоимость обедов. Во второй строке выдайте два числа K_1 и K_2 — количество купонов, которые останутся неиспользованными у Пети после этих N дней и количество использованных им купонов соответственно.

В последующих K_2 строках выдайте в возрастающем порядке номера дней, когда Пете следует воспользоваться купонами. Если существует несколько решений с минимальной суммарной стоимостью, то выдайте то из них, в котором значение K_1 максимально (на случай, если Петя когда-нибудь ещё решит заглянуть в это кафе). Если таких решений несколько, выведите любое из них.

Пример

Ввод	Вывод
5	235
35	0 1
40	5
101	
59	
63	

Язык

GNU GCC 12.2 C++20

Набрать здесь

Отправить файл

```
1 #include <algorithm>
2 #include <iostream>
3 #include <iterator>
4 #include <vector>
5
6 int main() {
7     int n;
8     std::cin >> n;
9     if (n == 0) {
10         std::cout << "0\n" << 0 << " " << 0;
11         return 0;
12     }
13     int m = n + 1;
14     std::vector<std::vector<int>> dp(n, std::vector<int>(m, 30001));
15     std::vector<int> in(n);
16     for (int i = 0; i != n; ++i) {
17         std::cin >> in[i];
18     }
19     if (in[0] <= 100) {
20         dp[0][0] = in[0];
21     } else {
22         dp[0][1] = in[0];
23     }
```

```
24
25     for (int i = 1; i != n; ++i) {
26         for (int j = 0; j != m; ++j) {
27             if (in[i] > 100){
28                 if (j > 0){
29                     dp[i][j] = std::min(dp[i][j], dp[i - 1][j - 1] + in[i]);
30                 }
31                 if (j < m - 1) {
32                     dp[i][j] = std::min(dp[i][j], dp[i - 1][j + 1]);
33                 }
34             } else {
35                 if (j < m - 1) {
36                     dp[i][j] = std::min(dp[i][j], dp[i - 1][j + 1]);
37                 }
38                 dp[i][j] = std::min(dp[i][j], dp[i - 1][j] + in[i]);
```

[Отправить](#)[Предыдущая](#)[Следующая](#)