Oczko (rozwiązanie)

Autor zadania: **Bartosz Kostka**
Opracowanie: **Adam Al-Hosam, Krzysztof Bartoszek**
Opis rozwiązania: **Bartosz Kostka**

Olimpiada Informatyczna Juniorów

W tym zadaniu mamy dane zasady gry w oczko. Rozdaliśmy już karty dla każdego gracza i teraz chcemy powiedzieć, kto jest zwycięzcą tej rundy.

Zadanie to rozwiążemy w dwóch krokach. Najpierw dla każdego gracza wyliczymy jego wynik (liczbę punktów). Następnie wyznaczymy zwycięzców biorąc pod uwagę wyniki graczy.

Zaczniemy zatem od zaimplementowania funkcji `wynik`, która dla ręki danego gracza (typem ręki będzie napis). Dla wszystkich kart poza asem sytuacja jest dość prosta, musimy dodać liczbę punktów odpowiadającą danej karcie (wszystkie karty powyżej dziesiątki (dziesiątka, walet, dama i król są warte 10 punktów, dwójka jest warta 2 punkty, trójka jest warta 3 punkty, itd.).

Zastanówmy się chwilę nad tym jak liczyć asy. Każdy as może być liczony jako 1 lub 11 – dla gracza wybierany jest korzystniejszy wariant. Załóżmy zatem, że na początku wszystkie asy będziemy liczyć jako 1. Po przeliczeniu wszystkich kart, jeżeli uznamy jednak, że mamy za mało punktów i asy powinniśmy dobrać jako 11, to dodamy po prostu 10 do wyniku. Warto zwrócić uwagę, że może tak się stać jedynie raz, kiedy mamy nie więcej niż 11 punktów (licząc wcześniej asy jako 1). Innymi słowy, jedynie jeden as może być policzony jako 11, ponieważ gdybyśmy dwa (lub więcej) asów policzyli jako 11, to wtedy przekroczylibyśmy magiczną liczbę 21.

Funkcja do zliczania punktów gracza (Python)

```python
# Obliczamy wynik punktowy dla danej reki.
def wynik(reka):
  # Deklarujemy zmienne na liczbę punktów oraz liczbę asów.
  punkty = 0
  asy = 0

  # Dla każdej karty.
  for karta in reka:
    # Sprawdzamy czy ta karta jest pomiędzy 2 a 9.
    # Zwróć uwagę, że karta jest typu napis,
    # zatem 2 i 9 także muszą być napisami.
    if '2' <= karta <= '9':
      # Dodajemy wartość tej karty, którą otrzymujemy
      # poprzez zrzutowanie napisu na typ int.
      punkty += int(karta)
    # W przeciwnim wypadku, jeżeli karta jest asem.
    elif karta == 'A':
      # Dodajemy jeden do liczby punktów (który zawsze musimy wziąć).
      punkty += 1
      # Oraz dodajemy jeden do liczby asów.
      asy += 1
    else:
      # W przeciwnym wypadku mamy dziesiątkę,
      # waleta, damę, lub króla.  Każda z tych
      # kart jest warta 10.
      punkty += 10

  # Teraz możemy wziąć większą wartość asa (11),
  # jeżeli nie przekroczymy 21.
  if asy > 0 and punkty <= 11:
    asy -= 1
    punkty += 10

  # Finalnie funkcja zwraca liczbę punktów
  # w optymalnym wariancie.
  return punkty
```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji i Nauki w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022”.

Funkcja do zliczania punktów gracza (C++)

```cpp
// Obliczamy wynik punktowy dla danej reki.
int wynik(const string& reka) {
  // Deklarujemy zmienne na liczbę punktów oraz liczbę asów.
  int punkty = 0;
  int asy = 0;

  // Dla każdej karty.
  for (char karta : reka) {
    // Sprawdzamy czy ta karta jest pomiędzy 2 a 9.
    // Zwróć uwagę, że karta jest typu napis,
    // zatem 2 i 9 także muszą być napisami.
    if ('2' <= karta and karta <= '9') {
      // Dodajemy wartość tej karty, którą otrzymujemy
      // poprzez usunięcie wartości litery '0'.
      punkty += karta - '0';
    } else if (karta == 'A') {
      // W przeciwnim wypadku, jeżeli karta jest asem.
      // Dodajemy jeden do liczby punktów (który zawsze musimy wziąć).
      punkty++;
      // Oraz dodajemy jeden do liczby asów.
      asy++;
    } else {
      // W przeciwnym wypadku mamy dziesiątkę,
      // waleta, damę, lub króla.  Każda z tych
      // kart jest warta 10.
      punkty += 10;
    }
  }

  // Teraz możemy wziąć większą wartość asa (11),
  // jeżeli nie przekroczymy 21.
  if (asy > 0 and punkty <= 11) {
    asy--;
    punkty += 10;
  }

  // Finalnie funkcja zwraca liczbę punktów
  // w optymalnym wariancie.
  return punkty;
}
```

Teraz pozostaje nam dopisać resztę programu, czyli na podstawie wyników wszystkich graczy, znaleźć zwycięzców. Możemy wszystkie wyniki zapisać do wektora/tablicy/listy. Wtedy aby znaleźć wynik najlepszego gracza wystarczy przeiterować się po wszystkich wynikach i znaleźć największy nieprzekraczający 21 (zwróć uwagę, że nie musi taki istnieć). Następnie możemy ponownie przeiterować się po wektorze/tablicy/liście i wypisać numery wszystkich graczy, którzy osiągnęli taki wynik.

Poniższe programy zawierają implementację takiego podejścia i zawierają wcześniej zaimplementowane funkcje `wynik`.

ocz.py

```python
# Obliczamy wynik punktowy dla danej reki.
def wynik(reka):
  # Deklarujemy zmienne na liczbę punktów oraz liczbę asów.
  punkty = 0
  asy = 0

  # Dla każdej karty.
  for karta in reka:
    # Sprawdzamy czy ta karta jest pomiędzy 2 a 9.
```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji i Nauki w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

```
10      # Zwróć uwagę, że karta jest typu napis,
11      # zatem 2 i 9 także muszą być napisami.
12      if '2' <= karta <= '9':
13        # Dodajemy wartość tej karty, którą otrzymujemy
14        # poprzez zrzutowanie napisu na typ int.
15        punkty += int(karta)
16      # W przeciwnim wypadku, jeżeli karta jest asem.
17      elif karta == 'A':
18        # Dodajemy jeden do liczby punktów (który zawsze musimy wziąć).
19        punkty += 1
20        # Oraz dodajemy jeden do liczby asów.
21        asy += 1
22      else:
23        # W przeciwnym wypadku mamy dziesiątkę,
24        # waleta, damę, lub króla.  Każda z tych
25        # kart jest warta 10.
26        punkty += 10
27
28    # Teraz możemy wziąć większą wartość asa (11),
29    # jeżeli nie przekroczymy 21.
30    if asy > 0 and punkty <= 11:
31      asy -= 1
32      punkty += 10
33
34    # Finalnie funkcja zwraca liczbę punktów
35    # w optymalnym wariancie.
36    return punkty
37
38
39  # Wczytujemy liczbę graczy.
40  N = int(input())
41
42  # Wczytujemy ręce graczy oraz zapisujemy
43  # ich optymalne wyniki w liście `wyniki`.
44  wyniki = []
45  for i in range(N):
46    reka = input()
47    wynik_dla_reki = wynik(reka)
48    wyniki.append(wynik_dla_reki)
49
50  # Szukamy najlepszego wyniku nie przekraczającego 21.
51  najlepszy_wynik = -1
52  for wynik in wyniki:
53    # Rozpatrujemy tylko wyniki niewiększe od 21.
54    if wynik <= 21:
55      if wynik > najlepszy_wynik:
56        najlepszy_wynik = wynik
57
58
59  # Teraz generujemy listę numerów zawodników,
60  # którzy są zwycięzcami.
61  zwyciezcy = []
62  for i in range(N):
63    if wyniki[i] == najlepszy_wynik:
64      # Zwróć uwagę, że zawodnicy są numerowani od 1 do N,
65      # a w programie i jest od 0 do N - 1, zatem musimy dodać jeden.
66      zwyciezcy.append(i + 1)
67
```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji i Nauki w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022”.

```python
# Wypisz liczbę zwycięzców.
print(len(zwyciezcy))
# A następnie ich numery.
for zwyciezca in zwyciezcy:
  print(zwyciezca, end='␣')
print()
```

ocz.cpp

```cpp
#include <iostream>
#include <vector>

using namespace std;

// Obliczamy wynik punktowy dla danej reki.
int wynik(const string& reka) {
  // Deklarujemy zmienne na liczbę punktów oraz liczbę asów.
  int punkty = 0;
  int asy = 0;

  // Dla każdej karty.
  for (char karta : reka) {
    // Sprawdzamy czy ta karta jest pomiędzy 2 a 9.
    // Zwróć uwagę, że karta jest typu napis,
    // zatem 2 i 9 także muszą być napisami.
    if ('2' <= karta and karta <= '9') {
      // Dodajemy wartość tej karty, którą otrzymujemy
      // poprzez usunięcie wartości litery '0'.
      punkty += karta - '0';
    } else if (karta == 'A') {
      // W przeciwnim wypadku, jeżeli karta jest asem.
      // Dodajemy jeden do liczby punktów (który zawsze musimy wziąć).
      punkty++;
      // Oraz dodajemy jeden do liczby asów.
      asy++;
    } else {
      // W przeciwnym wypadku mamy dziesiątkę,
      // waleta, damę, lub króla.  Każda z tych
      // kart jest warta 10.
      punkty += 10;
    }
  }

  // Teraz możemy wziąć większą wartość asa (11),
  // jeżeli nie przekroczymy 21.
  if (asy > 0 and punkty <= 11) {
    asy--;
    punkty += 10;
  }

  // Finalnie funkcja zwraca liczbę punktów
  // w optymalnym wariancie.
  return punkty;
}


int main() {
  // Wczytujemy liczbę graczy.
  int N;
  cin >> N;
```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji i Nauki w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

```
52
53    // Wczytujemy ręce graczy oraz zapisujemy
54    // ich optymalne wyniki w liście ‘wyniki‘.
55    vector<int> wyniki;
56    for (int i = 0; i < N; i++) {
57      string reka;
58      cin >> reka;
59      int wynik_dla_reki = wynik(reka);
60      wyniki.push_back(wynik_dla_reki);
61    }
62
63    // Szukamy najlepszego wyniku nie przekraczającego 21.
64    int najlepszy_wynik = -1;
65    for (int wynik : wyniki) {
66      // Rozpatrujemy tylko wyniki niewiększe od 21.
67      if (wynik <= 21)
68        if (wynik > najlepszy_wynik)
69          najlepszy_wynik = wynik;
70    }
71
72    // Teraz generujemy listę numerów zawodników,
73    // którzy są zwycięzcami.
74    vector<int> zwyciezcy;
75    for (int i = 0; i < N; i++) {
76      if (wyniki[i] == najlepszy_wynik) {
77        // Zwróć uwagę, że zawodnicy są numerowani od 1 do N,
78        // a w programie i jest od 0 do N - 1, zatem musimy dodać jeden.
79        zwyciezcy.push_back(i + 1);
80      }
81    }
82
83    // Wypisz liczbę zwycięzców.
84    cout << zwyciezcy.size() << "\n";
85    // A następnie ich numery.
86    for (int zwyciezca : zwyciezcy)
87      cout << zwyciezca << "␣";
88    cout << "\n";
89  }
```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji i Nauki w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022”.