

ZADANIE IGR: ROZWIĄZANIE

(PYTHON)



Olimpiada
Informatyczna
Juniorów

W tym zadaniu podana jest najpierw liczba zawodników n , a potem ich kolejne wyniki. Wczytajmy najpierw liczbę:

```
n = input()
```

Czas na drugi wiersz – oddzielone spacjami wyniki.
Instrukcja `input()` oznacza po prostu wiersz, za to
`input().split()` – rozbicie tego wiersza na poszczególne
kawałki względem znaku spacji. Jeśli na wejściu będzie napis
"8 2 6 1", to `input().split()` będzie oznaczało tablicę
["8", "2", "6", "1"].

Elementy tablicy `input().split()` są jednak napisami, a nie liczbami. Aby je szybko i wygodnie zamienić na liczby, użyjemy charakterystycznej dla Pythona konstrukcji, która nazywa się *list comprehension*, czyli *odwzorowanie listy*:

```
wyniki = [int(x) for x in input.split()]
```

Czyli: weź wszystkie napisy `x` z listy `input.split()`, dla każdego wykonaj `int(x)` i całość zgromadź w nową listę, którą następnie zapisz do zmiennej `wyniki`.

W Pythonie jest bardzo naturalny i szybki sposób na uporządkowanie listy w kolejności rosnącej:

```
wyniki_s = sorted(wyniki)
```

Czyli po prostu: do zmiennej `wyniki_s` wpisz tę samą listę, co `wyniki`, tyle że posortowaną.

Nam bardziej przyda się lista posortowana malejąco, czyli w odwróconym porządku:

```
wyniki_s = sorted(wyniki, reverse = True)
```

Teraz wystarczy zwyczajnie wypisać pierwsze trzy elementy posortowanej listy – będą to szukane trzy najlepsze wyniki. Całość programu wygląda następująco:

```
n = input()
wyniki = [int(x) for x in input().split()]

wyniki_s = sorted(wyniki,reverse=True)

print(wyniki_s[0])
print(wyniki_s[1])
print(wyniki_s[2])
```

Alternatywnym podejściem (bez użycia `sorted`) jest wyszukać w liście `wyniki` największy element:

```
kandydat = 0
for w in wyniki:
    if w > kandydat:
        kandydat = w
```

Dla każdego elementu z listy sprawdzamy, czy jest większy od aktualnie najlepszego kandydata – jeśli tak, to on zostaje nowym kandydatem.

Znaleziony najlepszy wynik wypisujemy, a potem usuwamy go z listy:

```
kandydat = 0
for w in wyniki:
    if w > kandydat:
        kandydat = w
print(kandydat)
wyniki.remove(kandydat)
```


Wystarczy teraz całą operację powtórzyć 3 razy, aby znaleźć złotego, srebrnego i brązowego medalistę.

```
n = input()
wyniki = [int(x) for x in input().split()]

for i in range(3):
    kandydat = 0
    for w in wyniki:
        if w > kandydat:
            kandydat = w
    print(kandydat)
    wyniki.remove(kandydat)
```

Które z podanych rozwiązań jest lepsze? Ciężko odpowiedzieć na to pytanie. Pierwsze – z instrukcją `sorted` – jest zgodne z „duchem” Pythona, jest krótsze i wygodniejsze. W tym zadaniu obie metody są podobnie efektywne i obie zostaną zaakceptowane przez system sprawdzający.

Ale na dłuższą metę używanie np. odwzorowań list, struktur słowników i innych instrukcji charakterystycznych dla Pythona ma tendencję do pogarszania złożoności obliczeniowej programu, czasem w sposób trudny do przewidzenia. W dalszych etapach Olimpiady, kiedy złożoność rozwiązania będzie miała kluczowe znaczenie, trzeba się będzie dobrze zastanowić nad każdą taką instrukcją.