
PPPD - Lab. 09

Copyright ©2021 M. Śleszyńska-Nowak i in.

Zadanie punktowane, lab 09, grupa A, 2021/2022, autor: Małgorzata Śleszyńska-Nowak

Uwaga: w rozwiązaniu zadania nie można korzystać z `append`, `extend`, `pop`, `remove`, `slice`, indeksowania ujemnego i operacji na listach typu `+`, `+=` itp.

Temat: Alergeny

Treść zadania

Zbiór danych `alerg.csv` zawiera informacje o daniach w pewnej restauracji i alergenach, które te dania zawierają. Każdy wiersz odpowiada jednemu daniu, każda kolumna jednemu alergenowi. Jeżeli i -te danie zawiera j -ty alergen to `dane[i][j] = 1`, w przeciwnym przypadku `dane[i][j] = 0`.

Plik `.csv` załadujesz przy pomocy następującej funkcji:

```
import csv

def wczytaj_dane(sciezka):
    M = []
    f = open(sciezka)
    for row in csv.reader(f):
        for i in range(len(row)):
            row[i] = int(row[i]) # konwersja z str na int
        list.append(M, row) # == A.append(row)
    f.close()
    return M
```

Od teraz `dane` jest macierzą, w której liczba wierszy odpowiada liczbie dań, liczba kolumn odpowiada liczbie alergenów, w każdym polu `[i][j]` jest zmienna typu `int` o wartości 0 albo 1.

Zadanie

Zadanie składa się z 5 niezależnych od siebie etapów, każdy za 2 punkty, które mogą być rozwiązywane w dowolnej kolejności.

1. `wypisz(dane)` (2 punkty)

Napisz funkcję `wypisz(dane)`, która wypisze w odpowiednim formacie macierz z danymi.

Wymagania:

- w pierwszym wierszu: indeksy alergenów (przesunięte w prawo aby były nad odpowiednimi kolumnami)
- w każdym następnym wierszu: najpierw indeks dania i dwukropek, potem wartości z macierzy odpowiadające danemu daniu

Wzór:

```

    0 1 2 3 4 5 6 7 8 9 10
0: 0 0 1 0 1 0 1 0 0 0 1
1: 1 0 0 0 1 0 0 0 0 1 0
2: 0 0 0 0 0 0 0 0 0 0 0
3: 0 0 1 0 1 0 1 0 1 0 1
4: 0 1 0 1 0 0 0 1 1 0 1
5: 0 0 0 0 1 0 0 0 0 0 1
6: 0 0 1 1 1 0 0 0 1 1 0
7: 0 0 0 0 0 1 1 1 0 0 0
8: 0 0 0 0 0 0 0 0 0 0 0

```

2. `szukaj_bezpiecznych_dan(dane, lista)` (2 punkty)

Napisz funkcję `szukaj_bezpiecznych_dan(dane, lista)`, która przyjmuje macierz z danymi `dane` oraz listę z indeksami alergenów `lista`, których chcemy uniknąć (`lista` nie musi być posortowana). Funkcja ma zwrócić listę z indeksami dań (w kolejności rosnącej), które nie zawierają żadnego alergenu z `lista`.

UWAGA: funkcja nie może modyfikować swoich argumentów.

Przykład działania:

```

print("dania bez 0, 5, 10")
print(szukaj_bezpiecznych_dan(dane, [0, 5, 10]))

```

```

dania bez 0, 5, 10
[2, 6, 8]

```

3. `dodaj_nowe_danie(dane, lista)` (2 punkty)

Napisz funkcję `dodaj_nowe_danie(dane, lista)`, która przyjmuje macierz z danymi `dane` oraz listę z indeksami alergenów `lista`, które występują w nowym daniu (`lista` nie musi być posortowana). Funkcja ma zwrócić macierz z dodanym nowym daniem. Danie dodajemy w ostatnim wierszu, indeksy pozostałych dań nie mogą zostać zmienione.

UWAGA: funkcja nie może modyfikować swoich argumentów.

Przykład działania:

```

print("dodane nowe danie z alergenami 1, 2, 3, 8")
dane = dodaj_nowe_danie(dane, [1, 2, 3, 8])
wypisz(dane)

```

```

dodane nowe danie z alergenami 1, 2, 3, 8
    0 1 2 3 4 5 6 7 8 9 10
0: 0 0 1 0 1 0 1 0 0 0 1
1: 1 0 0 0 1 0 0 0 0 1 0
2: 0 0 0 0 0 0 0 0 0 0 0
3: 0 0 1 0 1 0 1 0 1 0 1
4: 0 1 0 1 0 0 0 1 1 0 1
5: 0 0 0 0 1 0 0 0 0 0 1
6: 0 0 1 1 1 0 0 0 1 1 0
7: 0 0 0 0 0 1 1 1 0 0 0
8: 0 0 0 0 0 0 0 0 0 0 0
9: 0 1 1 1 0 0 0 0 1 0 0

```

4. `usun_najgorsze(dane, k)` (2 punkty)

Napisz funkcję `usun_najgorsze(dane, k)`, która przyjmuje macierz z danymi `dane` oraz liczbę naturalną `k`. Funkcja ma zwrócić macierz z danymi, w której zostaną usunięte dania zawierające co najmniej `k` alergenów. Kolejność pozostałych dań ma zostać niezmienniona.

UWAGA1: funkcja może korzystać tylko z $O(\text{len}(\text{dane}))$ dodatkowej pamięci, oprócz pamięci zajmowanej przez wynikową macierz.

UWAGA2: funkcja nie może modyfikować swoich argumentów.

Przykład działania:

```
print("dania po usunięciu tych z co najmniej 5 alergenami")
dane = usun_najgorsze(dane, 5)
wypisz(dane)
```

```
dania po usunięciu tych z co najmniej 5 alergenami
  0 1 2 3 4 5 6 7 8 9 10
0: 0 0 1 0 1 0 1 0 0 0 1
1: 1 0 0 0 1 0 0 0 0 1 0
2: 0 0 0 0 0 0 0 0 0 0 0
3: 0 0 0 0 1 0 0 0 0 0 1
4: 0 0 0 0 0 1 1 1 0 0 0
5: 0 0 0 0 0 0 0 0 0 0 0
6: 0 1 1 1 0 0 0 0 1 0 0
```

5. zapisz_informacje(dane, sciezka) (2 punkty)

Napisz funkcję `zapisz_informacje(dane, sciezka)`, która przyjmuje macierz z danymi `dane` oraz ścieżkę do pliku `.txt` `sciezka`. Funkcja ma zapisać informacje odnośnie danych (liczba dań, liczba alergenów, maksymalna liczba alergenów w jednym daniu, liczba dań z ... alergenami) do pliku w odpowiednim formacie (podanym niżej).

UWAGA: funkcja nie może modyfikować swoich argumentów.

Przykład działania:

```
zapisz_informacje(dane, "zapis.txt")
```

Wymagany format:

```
liczba dan: 7
liczba uwzględnionych alergenów: 11
maksymalna liczba alergenów w jednym daniu: 4
liczba dan z 0 alergenami: 2
liczba dan z 1 alergenami: 0
liczba dan z 2 alergenami: 1
liczba dan z 3 alergenami: 2
liczba dan z 4 alergenami: 2
```

Uwaga

- Jeśli rozwiązanie nie spełnia postawionych wymagań (np. korzysta z zabronionych konstrukcji, modyfikuje argumenty funkcji), funkcja jest oceniana na 0 punktów.
- Jeśli program się nie kompiluje (interpretuje), ocena jest zmniejszana o połowę.
- Jeśli kod programu jest niskiej jakości (nieestetycznie formatowanie, mylące nazwy zmiennych itp.), ocena jest zmniejszana o 2p.