

## Čo máme na obed?

Začiatok semestra je tu a študenti opäť čelia najdôležitejšej otázke dňa: Čo si dám dnes na obed? Výber jedla zo širokej ponuky našich školských jedální je veľká výzva sama osebe, no keď niekto chce mať vyváženú stravu s potrebným množstvom živín a zároveň musí brať do úvahy hranice pre dotáciu, tak rozhodnutie je skoro nemožné. Vašou úlohou v tejto úlohe zadania je vytvoriť riešenie, ktoré pomôže stravníkom vybrať si to najlepšie riešenie, čo jedáleň v daný deň ponúka.

Vstupom riešenia bude zoznam ponúkaných jedál vo forme csv súboru, ako môžete vidieť na príklade nižšie (anglické názvy sa používajú pre jednoduchšiu prácu a výpisy počas vývoja). Prvý riadok súboru popisuje atribúty zahrnuté v našej databáze, a to:

- meal – meno jedla
- category – typ jedla; v našej implementácii budú existovať polievky (soup), hlavné jedlá (main dish), prílohy (sidedish) a dezerty (dessert)
- calories – počet kalórií v danom jedle na 100g (alebo 100ml v prípade polievok)
- protein – množstvo bielkovín v jedle na 100g (na 100ml pre polievky)
- fat – množstvo tuku v jedle na 100g (na 100ml pre polievky)
- carbs – množstvo sacharidov v jedle na 100g (na 100ml pre polievky)
- amount – veľkosť porcie v gramoch alebo mililitroch pre polievky
- price – cena porcie.

```
meal,category,calories,protein,fat,carbs,amount,price
boiled potatoes,sidedish,93,2.18,0.22,21.29,280,1.10
boiled rice,sidedish,206,4.47,4.33,38.6,190,0.60
fried chicken,main dish,369,21.9,18.3,28.7,180,2.60
goulash soup,soup,63,3.82,3.1,5.36,350,0.6
banana,dessert,90,1.2,0.24,19.8,120,0.25
```

Obed stravníka pozostáva z polievky, hlavného jedla, prílohy a dezertu. Úlohou vášho kódu je vybrať čo najlacnejšiu kombináciu. Aby úloha nebola príliš jednoduchá, pridáme ďalšie ohraničenia, a to v množstve kalórií, bielkovín, tukov a sacharidov, ktoré náš stravník chce mať vo svojom obede. Tieto intervaly budú dané v ďalšom súbore. Príklad obsahu takéhoto súboru vidíte nižšie.

```
calories,50,1000
protein,5,300
fat,5,300
carbs,5,300
price,4.00
```

Formát je txt s pravidlami csv, avšak tentokrát bez hlavičky. Každý riadok popisuje limity pre daný ukazovateľ. V tomto konkrétnom príklade hľadá stravník kombináciu jedál s celkovým množstvom kalórií medzi 50 a 1000 kcal; medzi 5 a 300 g bielkovín; 5 a 300 g tukov; 5 a 300 g sacharidov. Posledný riadok vyjadruje rozpočet stravníka. Môže byť zadaný iba horný limit (v našom príklade maximálne 4 eurá) alebo aj dolný aj horný limit (napríklad

price, 2.50, 4.00). Ak stravník nezadal dolný limit, budeme to považovať za 0. Poradie ohraňení v súbore bude vždy rovnaké a súbor musí obsahovať hodnoty pre všetky ukazovatele.

Po úspešnom načítaní ponuky a ohraňení prichádza samotný výber. Cieľom je vybrať takú kombináciu polievky, hlavného jedla, prílohy a dezertu, aby sme ostali v definovanom intervale pre každý ukazovateľ (kalórie, bielkoviny, tuky, sacharidy) a pritom sme minimalizovali celkovú cenu v rámci ohraňení (teda cena nemôže byť menšia ako dolný limit a väčšia ako horný limit).

Pri výpočte týchto údajov využite informácie načítané z ponuky jedál, pričom dávajte si pozor na veľkosť porcie. Napríklad pre varené zemiaky (1. riadok v našom príklade) platí, že 100g obsahuje 93 kalórií, avšak porcia má 280g, teda množstvo kalórií v porcii je  $2,8 * 93 = 260,4$  kcal. Rovnakým spôsobom vieme vypočítať hodnoty pre ostatné jedlá v našej ponuke: 391,4 kcal pre ryžu; 664,2 kcal pre vyprážaný rezeň; 220,5 kcal pre gulášovú polievku a 108 kcal pre banány.

V takomto prípade máme možnosť vybrať si z dvoch príloh: zemiaky alebo ryža. Ak ale máme limitované množstvo kalórií pre celý obed na 1300 kcal, tak vidíme, že musíme vybrať zemiaky (celkovo 1253,1 kcal na obed), keďže s ryžou by sme túto hodnotu už presiahli (1384,1 kcal) hoci by sme vykombinovali lacnejší obed.

Na druhej strane, zemiaky obsahujú menej bielkovín a teda ak nášmu stravníkovi viac záleží na množstve bielkovín a nastavil väčšiu hodnotu dolného limitu bielkovín v obede, tak si musí vybrať ryžu. Ak by všetky ohraňenia spĺňal aj obed so zemiakmi aj obed s ryžou, vybrali by sme lacnejší variant, teda ryžu.

Na základe týchto poznatkov a pravidiel, potom môžeme implementovať náš systém pre výber obedu.

### Úloha 1 – 0,25 bodov

Implementujte metódu `load_meals`, ktorá má jeden parameter a to cestu k súboru, ktorý obsahuje ponuku jedál v jedálni vo formáte `csv`. Prvý riadok v tomto súbore predstavuje hlavičku s názvami atribútov, pričom každý ďalší riadok popisuje jednu položku. Príklady takýchto `csv` súborov nájdete v priečinku `1a_samples`.

Funkcia `load_meals` vracia jednu hodnotu, a to `pandas dataframe`.

### Úloha 2 – 0,5 bodov

Implementujte metódu `split_into_categories`, ktorá rozdelí `dataframe` načítaný metódou `load_meals` na štyri menšie `dataframey` s rôznymi typmi jedál: polievky, hlavné jedlá, prílohy a dezerty. Návrátové hodnoty funkcie sú tieto štyri `dataframey`.

### Úloha 3 – 2 body

Implementujte metódu `load_intervals`, ktorá načíta ohraňenia pre ukazovatele kalórie, bielkoviny, tuky, sacharidy a cena zo súboru `txt`. Jej parametrom je zároveň cesta k tomuto súboru. Každý riadok v tomto súbore reprezentuje limity niektorého ukazovateľa s tromi hodnotami: názov ukazovateľa, dolný limit a horný limit. V prípade ceny (`price`) dolný limit

môže chýbať, v takomto prípade ho považujte za 0. Funkcia vráti jeden dictionary s načítanými údajmi, kde kľúčom je názov ukazovateľa a hodnotou je dvojica hodnôt s dolným a horným limitom. Limitné hodnoty ceny sú uložené ako desatinné čísla (`float`), ostatné číselné hodnoty sú celé čísla (`int`). Funkcia vracia dictionary iba v prípade, ak všetky načítané údaje sú platné. V opačnom prípade sa vygeneruje výnimka (kontrolu rieši funkcia `check_intervals`).

Implementujte metódu `check_intervals`, ktorá skontroluje platnosť načítaných údajov. Funkcia dostane jeden parameter – dictionary načítaný funkciou `load_intervals`. Funkcia `check_intervals` nemá návratovú hodnotu, v prípade neplatných údajov vygeneruje výnimku a to nasledovne:

- `TypeError` so správou *Intervals must be loaded as dictionary* ak parameter nereprezentuje dictionary
- `KeyError` so správou *Missing expected key XY* ak chýba niektorý z očakávaných kľúčov `calories`, `protein`, `fat`, `carbs`, `price` (názov kľúča použite namiesto XY)
- `TypeError` so správou *Interval limits should be loaded as tuples* ak pod niektorým kľúčom nie je uložená hodnota typu tuple
- `ValueError` so správou *Interval limits should be loaded as tuples of two values* ak pod kľúčom je uložený tuple avšak nie s dvomi hodnotami
- `TypeError` so správou *XY limits should be set as T* ak ohraničenia neboli definované správnym typom: pre `price` to musí byť `float`, pre ostatné ukazovatele celé čísla (namiesto XY použite meno ukazovateľa a namiesto T očakávaný typ `float/int`)
- `ValueError` so správou *Upper limit cannot be smaller than lower limit* ak horný limit je menší ako dolný limit pri niektorom ukazovateli

#### Úloha 4 – 0,5 bodov

Implementujte funkciu `calculate_stats`, ktorá vypočíta výživové hodnoty pre jednu porciu jedla. Ako parameter dostane riadok z dataframeu, t.j. `Series` s načítanými údajmi. V tomto riadku sú výživové hodnoty – kalórie, bielkoviny, tuky, sacharidy – dané pre 100g daného výrobku, tieto hodnoty potrebujete prepočítať pre porciu, ktorej veľkosť je daná atribútom `amount`.

Funkcia vracia štyri hodnoty: množstvo kalórií, bielkovín, tukov a sacharidov v porcii jedla `meal`.

#### Úloha 5 – 1 bod

Implementujte funkciu `evaluate_lunch`, ktorá zistí, či kombinácia jedál – obed – spĺňa ohraničenia dané intervalmi jednotlivých ukazovateľov. Funkcia vráti hodnotu `True`, ak obed je prijateľný, ak niektorý limit nebol dodržaný, tak vráti `False`. Parametre funkcie sú nasledovné:

- `soup` – vybraná polievka z ponuky – riadok z dataframeu typu `Series`
- `main` – vybrané hlavné jedlo z ponuky – riadok z dataframeu typu `Series`
- `side` – vybraná príloha z ponuky – riadok z dataframeu typu `Series`
- `dessert` – vybraný dezert z ponuky – riadok z dataframeu typu `Series`
- `intervals` – slovník reprezentujúci limity – formát podľa načítaných údajov v `load_intervals`

**Poznámka:** V prípade ceny môže dôjsť kvôli zaokrúhľovaniu k malým rozdielom. Napríklad celková cena môže byť vypočítaná 3.6000000000000005, ktorá by ale mala byť akceptovaná, ak horný limit bol nastavený na 3,60 EUR. Použite toleranciu menej ako 0,001 (takto rozdiel 1 centu sa už nebude akceptovať).

### Úloha 6 – 0,25 bodov

Implementujte funkciu `get_lunch_price`, ktorá vypočíta celkovú cenu obedu, ktorý obsahuje polievku, hlavné jedlo, prílohu a dezert. Vybrané položky dostane funkcia ako parameter (rovnako ako `evaluate_lunch`). Funkcia vracia jednu hodnotu: celkovú cenu obeda ako desatinné číslo (`float`).

### Úloha 7 – 0,5 bodov

Implementujte funkciu `generate_combinations`, ktorá vráti zoznam všetkých možných obedov, ktoré sa dajú nakombinovať z ponuky. Funkcia dostane ako parameter štyri dataframey, a to dataframe obsahujúci všetky možné polievky, hlavné jedlá, prílohy a dezerty. Každá kombinácia musí mať jednu polievku, jedno hlavné jedlo, jednu prílohu a jeden dezert. Kombináciu reprezentujte ako n-ticu `pandas Series`, teda riadkov reprezentujúcich vybranú položku z ponuky.

### Úloha 8 – 1 bod

Implementujte funkciu `find_best_meal`, ktorá nájde najlacnejší možný obed, ktorý spĺňa podmienky na živiny stanovené stravníkom. Funkcia má nasledujúce parametre:

- `soups` – zoznam dostupných polievok – `pandas Dataframe`
- `main`s – zoznam dostupných hlavných jedál – `pandas Dataframe`
- `sides` – zoznam dostupných príloh – `pandas Dataframe`
- `desserts` – zoznam dostupných dezertov – `pandas Dataframe`
- `intervals` – slovník reprezentujúci limity – formát podľa načítaných údajov v `load_intervals`

Funkcia vracia dve hodnoty:

- najlacnejší možný obed spĺňajúci podmienky na živiny – zoznam štyroch `pandas Series`, teda zoznam s vybranou polievkou, hlavným jedlom, prílohou a dezertom; ak sa nedá nakombinovať obed, ktorý by spĺňal podmienky, hodnota je `None`
- celková cena obedu – desatinné číslo (`float`); ak sa nedá nakombinovať obed, ktorý by spĺňal podmienky, hodnota je nekonečno (`infinity`)

Kostra riešenia obsahuje ešte funkciu `main` a hlavnú funkciu s ukázkovým príkladom použitia. Tieto funkcie môžete ľubovoľne využívať na testovanie, pri hodnotení sa však očakáva že vaše funkcie budú spolupracovať ako je uvedené vo funkcii `main`. Pri riešení môžete vytvoriť ľubovoľné pomocné funkcie a môžete použiť hotové riešenia z ľubovoľného štandardného modulu jazyka Python. Použitie knižnice `pandas` je povolené.

Vaše riešenia môžete otestovať aj pomocou sady testov v súbore `sample_tests_1a.py`. Pri hodnotení vášho riešenia použijeme podobné testy, avšak ich bude viac.

Približná dĺžka riešenia: *cca. 160 riadkov kódu bez komentárov*.