

1. Да се напише Python скрипта со име „prva.py3“ која како аргументи добива име на влезна датотека и опционален втор аргумент. Во влезната датотека има мени, така што во секој ред одделно има по едно јадење од менито во формат `reden_broj ime_na_jadenje kategorija_na_jadenje cena`. Секој дел е одделен помеѓу себе со празно место. Доколку корисникот не внесе втор аргумент, тогаш се печати целото мени, но доколку внесе аргумент, тогаш ги печати само оние јадења чија категорија е иста со вториот аргумент (ги филтрира по категорија внесена како втор аргумент од командна линија). Потоа скриптата дозволува да се внесе од тастатура три јадења од менито преку внесување на нивниот реден број. На крај скриптата треба да ги отпечати имињата на одбраните јадења од корисникот (само оние кои ги има на менито што е претходно отпечатено) и вкупната сума за наплата (вкупната цена).

Пример:

```
./python3 meni.py3 dorucek
```

На екран:

```
1 omlet dorucek 150
7 poshirani dorucek 120
20 mekici dorucek 70
25 angliski dorucek 350
```

Од тастатура:

```
1 10 25
```

На екран:

```
Omlet
```

```
Angliski
```

```
Vkupno: 500
```

2. Да се напише SHELL скрипта која како аргумент добива даден збор. Shell скриптата треба да ги најде сите текстуални датотеки во кои се содржи тој збор (во тековниот директориум и поддиректориумите). Секоја датотека која ќе се најде, се повикува одделно Python скриптата од првата задача, каде како прв аргумент се праќа името на датотеката, додека пак како втор аргумент се праќа зборот (што се добива како аргумент на Shell скриптата). Резултатот од извршувањето на секоја Python скрипта се печати на екран.
3. Да се напише програма во C која како аргумент добива име на непостоечка датотека. Програмата најпрво креира дете процес. Родител процесот треба да генерира 1000 случајни броеви со помош на функцијата `rand()` (од библиотеката `stdlib.h`) и истите ги запишува на екран (кои со помош на редирекција се запишуваат во датотеката) и родител процесот завршува. Дете процесот проверува дали може да ја отвори датотеката, и ако не може заспива 1 секунда. Потоа проверува пак, и заспива пак 1 секунда, се додека не успее да ја отвори датотеката. Чим успее да ја отвори, чита 1000 броеви од тастатура (редиректирани од датотеката) и ги сместува во низа. Потоа, дете процесот, со помош на 4 нитки (секоја нитка пребарува по 250 броеви), го наоѓа најголемиот број и го печати на екран.