

1. Да се напише Python скрипта со име „prva.py3“ која како аргументи добива две влезни датотеки. Првата датотека е код напишан во Python, додека пак втората датотека е документација која треба да се вметне во кодот. Документацијата е организирана така што секоја линија од датотеката најпрво има број, а потоа текст. Скриптата треба да ги спои двете датотеки, така што втората датотека (документацијата) треба да ја додаде во програмскиот код (првата датотека). Додавањето се прави така што бројката на почеток на секоја линија во документацијата кажува на која линија во кодот треба да биде истата вметнатата, а потоа, на крајот на линијата во кодот, се додава линијата од документацијата, започнувајќи со #. Новоформириот код се печати на екран.

Пример:

```
./prg dat.py3 dok.co
```

Dat.py3:

```
for i in range(len(X)):
    for j in range(len(X[0])):
        result[i][j] = X[i][j] + Y[i][j]
```

for r in result:

```
    print(r)
```

dok.co

```
6 #print the result
```

```
2 #iterate columns
```

```
1 #iterate rows
```

```
3 #sum up
```

На екран:

```
for i in range(len(X)):#iterate rows
```

```
    for j in range(len(X[0])):#iterate columns
```

```
        result[i][j] = X[i][j] + Y[i][j]#sum up
```

for r in result:

```
    print(r)#print the result
```

2. Да се напише SHELL скрипта која треба да ги најде сите датотеки (во тековниот директориум и поддиректориумите) кои имаат наставка (екстензија) „.py3“. Потоа, скриптата треба за секоја датотека со наставка „.py“ да најде датотека со исто име како и датотеката само со екстензија „.co“. (Пример, ако најде датотека со име prvadat.py3, пребарува дали има датотека со име prvadat.co (во тековниот директориум и поддиректориумите). Скриптата, за секој пар датотеки што ќе ги најде датотека со наставка „.py“ и датотека со исто име само со наставка „.co“, ја повикува Python скриптата од првата задача и како аргументи им ги праќа ове две датотеки.
3. Да се напишат две програми во C кои ќе работат со процеси, нитки и споделена (shared) меморија. Двете програми комуницираат помеѓу себе со споделена (заедничка) меморија со клуч 12345. Втората програма, внесува од тастатура низа од 1000 броеви и истите ги запишува во заедничката меморија, а потоа, со помош на функцијата rand() (од библиотеката stdlib.h) случајно избира една од нитките на првата програма да го пребара својот број. Една нитка не може да се избере повеќе пати, така што треба да се грижи втората програма на следно бирање да избере некоја од преостанатите нитки.

Првата програма, дозволува да се внесат  $X$  броеви од тастатура ( $X$  исто така се внесува од тастатура и може да е со вредност најмногу 10). Потоа, програмата креира  $X$  нитки, така што секоја нитка ќе го пребарува соодветниот број внесен од тастатура. Секоја нитка, чека да и дојде редот (да биде избрана од втората програма) за да го пребара бројот во низата од 1000 броеви. На секоја секунда проверува дали она е на ред да пребарува, и притоа доколку е избрана, пребарува низ низата од 1000 броеви и печати на екран колку пати е пронајден соодветниот број. Откако ќе заврши, сигнализира на втората програма дека може да избере следна нитка за извршување.