

1. (1 bod) Ponašanje programa za signal SIGINT postavlja se u Pythonu sa:

```
signal.signal(signal.SIGINT, obrada_signala).
```

Što ako se to ne napravi (ne pozove prethodna funkcija)?

- signal će se ignorirati ako bude upućen dretvi
- signal će se zadržati – čekati će da se u dretvi tom funkcijom definira drukčije

c. signal će prekinuti izvođenje dretve – to je pretpostavljeno ponašanje za SIGINT

d. signal će ponovno pokrenuti program – to je pretpostavljeno ponašanje za SIGINT

2. (2 boda) Nakon pokretanja programa s desne strane, signal SIGINT upućen mu je u trenucima $t=2.5$ i $t=4$ (od pokretanja programa). Što će ispisati program od početka do završetka rada, uz ta dva signala (nadopuniti ispis)?

Pretpostaviti da signalom prekinuta operacija `time.sleep(1)` nakon obrade signala neće nastaviti sa spavanjem neprospavagog vremena (ako je zadano

vrijeme već proteklo). Npr. ako je `time.sleep(1)` prekinuta signalom nakon pola sekunde spavanja, nakon obrade signala (koji traje više sekundi) izvođenje programa će se nastaviti kodom nakon `time.sleep(1)` (neće sada odspavati još pola sekunde).

```
import signal, time
def obrada(signum):
    print("SIG-P")
    for i in range(1,6):
        print("SIG-:" + str(i))
        time.sleep(1)
    print("SIG-K")

print("GP-pocetak")
signal.signal(signal.SIGINT, obrada)
for i in range(1,6):
    print("GP-:" + str(i))
    time.sleep(1)
print("GP-kraj")
```

t	ispis	
0	GP-pocetak	
0	GP-1	
1	GP-2	
2	GP-3	
2.5	SIG-P	- SIGINT - poziva se obrada
2.5	SIG-1	
3.5	SIG-2	
4	SIG-P	- SIGINT - poziva se obrada
4	SIG-1	
5	SIG-2	
6	SIG-3	
7	SIG-4	
8	SIG-5	
9	SIG-K	
9	SIG-3	
10	SIG-4	
11	SIG-5	
12	SIG-K	
12	GP-4	
13	GP-5	
14	GP-KRAJ	