

## Վարժություն 1.9

Ունենք inc և dec ֆունկցիաները: inc-ը վերադարձնում է արգումենտին գումարած 1 արժեքը, իսկ dec-ը արգումենտից հանած 1 արժեքը

```
>>> def inc(a):  
    return a + 1
```

```
>>> def dec(a):  
    return a - 1
```

Կիրառելով պարամետրերի փոխարինման մոդելը, պարզաբանեք հետևյալ ֆունկցիաներից յուրաքանչյուրից առաջացող գործընթացները, դրանցից որն է ռեկուրսիվ, որը պոչավոր ռեկուրսիվ, պատասխանը հիմնավորեք

```
>>> def add1(a, b):  
    if a == 0:  
        return b  
    else:  
        return inc(add1(dec(a), b))
```

```
>>> def add2(a, b):  
    if a == 0:  
        return b  
    else:  
        return add2(dec(a), inc(b))
```

Օրինակի համար դիտարկեք add1(4, 5) և add2(4, 5) կանչերը

**Սկզբում դիտարկենք  $add1(a, b)$  ֆունկցիան:** Մինչև ֆունկցիայի աշխատանքին անդրադառնալը նշեմ որ այն **ռեկուրսիվ** է, այսինքն այն դեպքը, երբ ֆունկցիան կանչում է ինքն իրեն: Սակայն ռեկուրսիվ ֆունկցիաների հետ աշխատելիս պետք է հաշվի առնել մի կարևոր հանգամանք, որ այն պետք է ունենա բազային դեպք: Բազային դեպքը այն պայմանն է, որով տեղի է հաշվողական գործընթացի ավարտը այլապես ծրագիրը կընկներ անվերջ ռեկուրսիվ ցիկլի մեջ: Մեր պարագայում այս ֆունկցիայի բազային դեպքը  $a == 0$  -ն է:

```
>>> def add1(a, b):
    if a == 0:
        return b
    else:
        return inc(add1(dec(a), b))
```

Այժմ դիտարկենք ֆունկցիայի աշխատանքը նշված արգումենտների(պարամետրերի) դեպքում կիրառելով պարամետրերի փոխանցման մոդելը:

Ֆունկցիայի կանչ՝ խնդրի պայմանին համապատասխան՝ առաջին արգումենտը կստանա 4 իսկ երկրորդ՝ 5 արժեք:

a	b	dec(a)	add1(dec(a),b)	inc(add1(dec(a),b))
4	5	3	add1(3,5)	add1(3,5) + 1
3	5	2	add1(2,5)	add1(2,5) + 2
2	5	1	add1(1,5)	add1(1,5) + 3
1	5	0	add1(0,5)	$b + 4 = 5 + 4 = 9$

Յուրաքանչյուր ֆունկցիայի համար ստեղծվում են լոկալ environment- ներ: Դրանք իրենցից ներկայացնում են stack-ներ որտեղ պահվում են բոլոր փոփոխականների ընդունած արժեքները: Դրանք ստեղծվում են ֆունկցիայի կանչի ժամանակ և «մահանում» ֆունկցիայի հետ: Ինչպես նկատեցինք b-արժեքը մնաց անփոփոխ:

Այժմ կառուցենք լոկալ միջավայրեր:

**Դիտարկենք  $add2(a, b)$  ֆունկցիան:** Այս ֆունկցիան **պոչավոր ռեկուրսիվ է:** Ոնենք երկրորդ արգումենտ(b), որի հետ կատարվում են գործողությունները, իսկ որպես վերջնարդյունք ֆունկցիան կվերադարձնի հենց այդ արգումենտը(b): Ամեն քայլի ժամանակ լուրջալայրեր չեն ստեղծվում քանի որ նախորդ լուրջալայրում տվյալները այլևս անհրաժեշտ չեն:

```
def add 2(a, b):
    if a == 0:
        return b
    else:
        return add2(dec(a), inc(b))
```

a	dec(a)	b	inc(b)	add2(dec(a),inc(b))
4	3	5	6	add2(3,6)
3	2	6	7	add2(2,7)
2	1	7	8	add2(1,8)
1	0	8	9	add2(0,9) = 9