

Mövzu 2

Tam ədədlərlə iş

1. Ədədin rəqəmləri ilə iş

1.1. Tam ədədin rəqəmləri ilə hesablamalar.

- 1) Tam ədədləri 10-luqlara ayırmaq olar. Məsələn 2371 ədədini $2371 = 2 \cdot 10^3 + 3 \cdot 10^2 + 7 \cdot 10^1 + 1 \cdot 10^0$ (adətən daha sadə şəkildə $2371 = 2 \cdot 10^3 + 3 \cdot 10^2 + 7 \cdot 10 + 1$ şəkildə yazırıq) kimi yazmaq olar. Burada 2, 3, 7, 1 bizim ədədin rəqəmləridir. Biz növbəti mövzularda bəzən N tam ədədini $N = edcba$ kimi də yazacağıq ki, burada **a, b, c, d, e** ədədin rəqəmlərini göztərir (nümunədə beşrəqəmli ədəddir).
- 2) Əgər c – yüzlüklərin, b – onluqların, a – isə təklikləri bildirirsə, onda onlar $N = 100 \cdot c + 10 \cdot b + a$ ədədini təşkil edir. Məsələn, əgər **a=4, b=6, c=2** verilibsə onda biz $N=462$ ədədini əldə etmək istəyirik, bunun üçün $N = 100 \cdot c + 10 \cdot b + a$ şəkildə hesablamaq gərəkdir.
- 3) N ədədinin sonuna a rəqəminin əlavə edilməsi. $N = N \cdot 10 + a$. Məsələn, əgər $N = 34$ olarsa **a=7**, $N \cdot 10 + a = 347$, yəni biz 34 ədədinin sonuna 7 rəqəmini əlavə etdik.
- 4) Python-da ədədin rəqəmlərini ayırmaq və ya pozmaq üçün “%” və tam bölmə üçün isə “//” əməllərindən istifadə edilir.
Diqqət. Rəqəmləri ayırmaq yalnız müsbət ədədlərin üzərində yerinə yetirəcəyik. (bax. 2.1 Mövzu 1)
Rəqəmlərin ayrılıb silinməsinə aid nümunələr.
 - a) $72396 \% 10 = 6$; $72396 \% 100 = 96$; $72396 \% 1000 = 3966$; $72396 \% 10000 = 2396$;
 - b) $72396 // 10 = 7239$; $72396 // 100 = 723$; $72396 // 1000 = 72$; $72396 // 10000 = 7$;
 - c) Bizə $N = 458932$ ədədi verilib.

Nə etməli	Python
Sağdan 1-ci rəqəmi seçmək	$a = N \% 10$
Sağdan 2-ci rəqəmi seçmək	$b = N // 10 \% 10$
Sağdan 3-cü rəqəmi seçmək	$c = N // 100 \% 10$
Sağdan 4-cü rəqəmi seçmək. Yəni sağdan ilk 3 rəqəmi silərək 4-cü rəqəmi seçmək.	$d = N // 1000 \% 10$
Sağdan 5-ci rəqəmi seçmək.	$c = N // 10000 \% 10$
Sağdan 6-cı rəqəmi seçmək.	$d = N // 100000 \% 10$
Yəni əgər dəqiq 6-rəqəmli ədədirsə sadəcə olaraq belə yazmaq olar: $d = N // 100000$	$d = N // 100000 \% 10$

- 5) 4-cü bənddə qeyd etdiyimiz kimi “%” əməliyyatı bölmədən alınan qalıqın hesablanması üçündür (yalnız 10-a bölmək deyil). Məsələn, $10 \% 3 = 1$, $15 \% 4 = 3$, $14 \% 3 = 2$, $15 \% 3 = 0$. Əgər a ədədinin b-yə bölünməsindən alınan qalır 0-dırsa onda a ədədi b-yə qalıqsız bölünür, yəni b ədədi a-nın bölənidir. Belə də deyə bilərik ki, a ədədi b-yə tam bölünür.

Məsələn, əgər $a \% 2 = 0$ olarsa bu o mənaya gəlir ki, **a** ədədi 2-yə qalıqsız bölünür, deməli **a** cüt ədəddir. Ədədin cüt olub-olmadığını elə beləcə də yoxlayırıq.

1.2. Mətn şəklindəki (formatında) tam ədədlərlə iş.

1) **Tutaq ki, bizə $s = '123'$ ədədi verilib** (sətir olaraq), onda hər bir rəqəmi (sətrin simvolları kimi) belə seçmək olar: $u = s[0]$, $v = s[1]$, $w = s[2]$, əgər biz onları ədəd olaraq istəsək onda: $a = \text{int}(s[0])$, $b = \text{int}(s[1])$, $c = \text{int}(s[2])$ yerinə yetirərək onları tam ədədə çeviririk.

Ədədin özünü əldə etmək üçün isə belə edə bilərik:

a) $N = 100*a + 10*b + c$

b) $N = \text{int}(u + v + w)$ (burada $u + v + w$ rəqəmləri bildirən simvolları birləşdirir və alınan simvolu ədədə çevirir)

2) **Tutaq ki, bizə $s = '123'$ ədədi verilib** (sətir olaraq). s sətirini siyahıya çevirək:

```
>>> s1=list(s)
```

```
>>> print(s1)
```

```
['1', '2', '3']
```

İndi biz **s1** siyahısının elementləri ilə işləyə bilərik. Yadımızda saxlayaq ki, **s1** siyahısının elementlərini dəyişdirmək (element əlavə edib silmək) mümkün olsa da s sətirinin elementlərini dəyişdirmək mümkün deyil (yəni sətirlər dəyişəbilməyən veriləndir (datadır)).



Python-da siyahılara aid lazımlı məlumatlar

Tutaq ki, bizə $s = [7, 3, -4, 4, 8]$ siyahısı verilib.

		Примеры
s.append(x)	Siyahının sonuna element əlavə edir	$s = [7, 3, -4, 4, 8]$ $s.append(100)$ $print(s)$ $[7, 3, -4, 4, 8, 100]$
s.extend(L)	s siyahısının sonuna L siyahısını birləşdirərək onu genişləndirir genişləndirib.	$s = [7, 3, -4, 4, 8]$ $s1=[200,300,400]$ $s.extend(s1)$ $print(s)$ $[7, 3, -4, 4, 8, 200, 300, 400]$
s.insert(i, x)	i -saylı mövqeyə x elementini əlavə edir.	$s = [7, 3, -4, 4, 8]$ $s.insert(3, -100)$ $print(s)$ $[7, 3, -4, -100, 4, 8]$
s.remove(x)	Qiyməti x olan ilk rast gəldiyi elementi silir. Belə element olmazsa ValueError xətası çıxır.	$s = [7, 3, -4, 4, 8, -4, 3]$ $s.remove(-4)$ $print(s)$ $[7, 3, 4, 8, -4, 3]$
s.pop(i)	i saylı mövqedəki elementi siyahıdan silir və qaytarır. Əgər mövqe(indeks) göstərilməyibsə sonuncu elementi silib qaytarır.	$s = [7, 3, -4, 4, 8, -4, 3]$ $a=s.pop(2)$ $print(s)$ $print(a)$ $[7, 3, 4, 8, -4, 3]$ -4
s.index(x)	Qiyməti x olan ilk elementin indeksini	$s = [7, -4, 3, -4, 4, 8, -4, 3]$ $print(s.index(-4))$ 1

	qaytarır (axtarış əvvəldən sona doğru)	
s.index(x, i)	Qiyməti x olan ilk elementin indeksini qaytarır (axtarış i nömrəli elementdən başlayaraq sona qədər aparılır)	print(s.index(-4,4) 6
s.index(x, i,j)	Qiyməti x olan ilk elementin indeksini qaytarır (axtarış i və j nömrəli indekslər arasında aparılır)	s = [7,-4, 3, -4, 4, 8, -4, 3] print(s.index(-4,2,7) 3
s.count(x)	Qiyməti x olan elementlərin sayını qaytarır.	s = [7, 3, -4, 4, 8, -4, 3] k=s.count(-4) print(k) 2
s.sort()	Siyahını artma sırasına görə nizamlayır/çəşidləyir.	s = [7, 3, -4, 4, 8, -4, 3] s.sort() print(s) [-4, -4, 3, 3, 4, 7, 8]
s.sort(reverse = True)	Siyahını azalma sırasına görə nizamlayır/çəşidləyir.	s = [7, 3, -4, 4, 8, -4, 3] s.sort(reverse = True) print(s) [8, 7, 4, 3, 3, -4, -4]
sorted(s)	Artma sırasına görə nizamlanmış/çəşidlənmiş siyahını qaytarır. s siyahısının özü dəyişməz qalır.	s = [7, 3, -4, 4, 8, -4, 3] s1=sorted(s) print(s) [7, 3, -4, 4, 8, -4, 3] print(s1) [-4, -4, 3, 3, 4, 7, 8]
sorted(s, reverse = True)	Azalma sırasına görə nizamlanmış/çəşidlənmiş siyahını qaytarır. s siyahısının özü dəyişməz qalır.	s = [7, 3, -4, 4, 8, -4, 3] s1=sorted(s, reverse = True) print(s) [7, 3, -4, 4, 8, -4, 3] print(s1) [8, 7, 4, 3, 3, -4, -4]
s.reverse()	Siyahını tərsinə çevirir.	s = [7, 3, -4, 4, 8, -4, 3] s.reverse() print(s) [3, -4, 8, 4, -4, 3, 7]
s.copy()	Siyahının surətini qaytarır.	s = [7, 3, -4, 4, 8] s1=[200,300,400] s1=s.copy() print(s1) [7, 3, -4, 4, 8]
s.clear()	Siyahını təmizləyir (bütün elementlərini silir)	s = [7, 3, -4, 4, 8, -4, 3] s.clear() print(s) []
max(s), min(s)	Maksimal və minimal elementləri tapır.	s = [7, 3, -4, 4, 8, -4, 3] print(max(s),min(s))

		8 -4
sum(s)	Elementlərin cəmini hesablayır.	s = [7, 3, -4, 14] print(sum(s)) 20

2. Ədədin rəqəmləri üzərində əməllərə aid nümunələr

Aşağıdakı nümunələrin hər birində eyni misalı həm ədədlər həm də sətir şəklində verilmiş ədədlər, həm də siyahılarla həll edəcəyik.

2.1. Üçrəqəmli natural ədəd verilib. Onun rəqəmlərinin cəmini tapın.

Nö	Həlli	İzahı
1	1 n=int(input()) 2 a= n//100%10 3 b=n//10%10 4 c= n%10 5 s=a+b+c 6 print(s)	Ədədin daxil edilməsi Sağdan ilk rəqəmin seçilməsi Sağdan ikinci rəqəmin seçilməsi Sağdan sonuncu rəqəmin seçilməsi Rəqəmlərin cəminin hesablanması Nəticənin çapı
	1 n=int(input()) 2 a,b,c=n//100, n//10%10, n%10 3 print(a+b+c)	Yuxarıdakı ilə eyni proqramdır. Sadəcə olaraq 2,3,4-cü sətirin yerinə bir sətirdə yazılış qaydası göstərilib.
2	1 s=input() 2 a,b,c=int(s[0]),int(s[1]),int(s[2]) 3 print(a+b+c)	Ədəd sətir şəklində daxil edilir. Hər bir rəqəm ədədə çevrilir. Cəm çap olunur
	1 s=input() 2 print(int(s[0])+int(s[1])+int(s[2]))	Eyni proqramdır, sadəcə olaraq əlavə dəyişənlər istifadə edilməyib.
3	1 s=list(input()) 2 s=[int(a) for a in s] 3 print(sum(s))	list funksiyası giriş verilənini simvollarıdan ibarət siyahıya çevirir – '0', '1', '2',... '9' Siyahının hər bir elementi int() funksiyasının köməyi ilə tam ədədə çevrilir sum(s) rəqəmlərin cəmini hesablayır.
	1 print(sum([int(a) for a in input()]))	Eyni proqram cəmi bir sətirdə yazılıb.
Yuxarıdakı 6 proqramın hər birinə 245 ədədini daxil etsəniz nəticədə 11 ədədi alacaqsınız.		

2.2. Natural ədəd verilib. Bu ədədi təşkil edən rəqəmlərin ən böyüyünü tapın.

Nö	Həlli	İzahı
1	n=int(input()) maxd=0 while n>0: q=n%10 if q>maxd: maxd=q n=n//10 print(maxd)	Ədədin daxil edilməsi maxd dəyişənində maksimal rəqəmi saxlayacağıq. n>0 olduğu müddətdə sonuncu rəqəmi ayıraraq silirik. n ədədinin sonuncu rəqəmini seçirik Əgər seçilmiş ədəd maxd-dən böyükdürsə maxd-yə onu mənimsədirik Sonuncu rəqəmi silirik Nəticəni çap edirik

2	s=list(input())	list funksiyası ədədi siyahıya – ‘0’, ‘1’, ‘2’,...’9’
	maxd=int(max(s)) print(maxd)	max(s) funksiyası maksimum ədədi tapır Nəticəni çap edirik
	print(int(max(list(input()))))	Eyni həll bir sətirdə
Yuxarıdakı 6 proqramın hər birinə 2454032 ədədini daxil etsəniz nəticədə 5 ədədi alacaqsınız.		

2.3. Natural ədəd verilib. Onu təşkil edən rəqəmlər arasında ən böyüyünün sayını tapın. Yəni maksimal ədədi tapıb onun ədəddə neçə dəfə təkrarlandığını müəyyənləşdirin. Məsələn, 273527347536 ədədində 7 rəqəmi ən böyükdür və ədəddə üç dəfə təkrarlanır.

Nö	Həlli	İzahı
1	<pre>n=int(input()) maxd=0 k=1 while n>0: q=n%10 if q>maxd: maxd=q k=1 else: if q==maxd: k=k+1 n=n//10 print(maxd,k)</pre>	<p>Ədədin daxil edilməsi maxd dəyişəninə maksimal ədədi k dəyişəninə isə sayını yazacağıq n>0 olduqca son rəqəmi atırıq n-in son rəqəmini seçirik Əgər rəqəm maxd-dən böyükdürsə onda maxd-yə bu ədədi mənimsədirik və k=1, yəni yeni maxd üçün 1 olur əks halda əgər maxd cari rəqəmdirə k-nı bir vahid artırırıq son rəqəmi silirik nəticəni çap edirik</p>
2	<pre>n= list(input()) maxd=max(n) k=n.count(maxd) print(int(maxd), k)</pre>	<p>list funksiyası ədədi rəqəmlərdən ibarət simvollara çevirir – ‘0’, ‘1’, ‘2’,...’9’ max(n) maksimal elementi tapır n.count(maxd) maxd-yə bərabər rəqəmlərin sayını tapır nəticəni çap edir.</p>
Bu iki proqrama 273527347536 ədədi daxil etsəniz 7 3 cavabını alacaqsınız.		

2.4. N (N>=1000) natural ədədi verilib. Bu ədəddə minlik və onluq mövqedə duran rəqəmlərin yerini dəyişin. Məsələn, 2341**5674** ədədini 2341**7654** ilə əvəzləmək lazımdır.

Nö	Həlli	İzahı
1	<pre>n=int(input()) m=n//10000 a=n//1000%10 b=n//100%10 c=n//10%10 d=n%10 x=m*10000+c*1000+b*100+a*10+d print(x)</pre>	<p>Ədədin daxil edilməsi – məsələn 23415674 Son 4 rəqəmi pozub 2341 ədədini əldə edirik 5-i seçirik 6-nı seçirik 7-ni seçirik 4-ü seçirik Onlardan yeni ədəd yaradıırıq 2341*10000+7*1000+6*100+5*10+4 Nəticəni çap edirik</p>
	<pre>n=int(input()) m,a=n//10000,n//1000%10 b,c,d=n//100%10,n//10%10,n%10</pre>	Eyni proqramdır sadəcə olaraq operatorları birləşdirib yığcam şəkildə yazmışıq.

	<code>print((((m*10+c)*10+b)*10+a)*10+d)</code>	
2	<pre>n=input() l=len(n) m=n[:l-4]+n[l-2]+n[l-3]+n[l-4]+n[l-1] print(int(m))</pre>	<p>Ədədin sətir kimi daxili ədədin uzunluğunun tapılması yeni ədədin sətir şəklində formalaşdırılması alınmış mətni tam ədədə çeviririk çap edirik</p>
3	<pre>n=list(input()) l=len(n) n[l-4],n[l-2]=n[l-2],n[l-4] m="".join(n) print(int(m))</pre>	<p>list funksiyası rəqəmləri siyahı şəklində qaytarır – ‘0’, ‘1’, ‘2’,...’9’ rəqəmlərin sayı 5 və 7-nin yerini dəyişirik (23415674) ".join(n) funksiyası ilə siyahının elementlərini birləşdiririk alınmış mətni tam ədədə çeviririk çap edirik</p>

2.5. Verilmiş N ($N < 100000$) ədədini aşmayan və təşkil olunduğu rəqəmlərə bölünən natural ədədlərin sayını tapın.

No	Həlli	İzahı
1	<pre>n=int(input()) k=0 for i in range(1,n+1): m=i p=0 while m>0: q=m%10 if q==0 or i%q!=0: p=1 break m=m//10 if p==0: k+=1 print(k)</pre>	<p>1-dən n-ə qədər (n daxil olmaqla) ədədlərə baxırıq Yoxlama üçün $p=0$ şərtini götürürük Hər bir ədədi təşkil edən rəqəmlər üçün onun sıfıra bərabər olmasını və ədədin həmin rəqəmə bölünməməsini yoxlayırıq. Bu hal baş versə $p=1$ mənimsədiyə while dövründən çıxırıq. əgər while dövründən çıxmamışıqsa deməli $p=0$ olaraq qalıb, onda k-nın qiymətini bir vahid artırırıq</p>
2	<pre>n=int(input()) k=0 for i in range(1,n+1): m=i; p=0 while m>0: q=m%10 if q==0 or i%q!=0: p=1; break m=m//10 if p==0: k+=1 print(k)</pre>	<p>Eyni proqramdır. Sadəcə görə bilərik ki, bəzi operatorları bir sətirdə qruplaşdırmaq da mümkündür.</p>

2.6. N natural ədədi verilib. Ədədi təşkil edən tək rəqəmləri silin, lakin yerdə qalan rəqəmlərin sırası dəyişməsin. Məsələn, 527014 ədədi 204 ədədinə çevrilsin.

Nö	Həlli	İzahı
1	<pre>n=int(input()) m=0; p=1 while n>0: q=n%10 if q%2==0: m=p*q+m p=p*10 n=n//10 print(m)</pre>	Hər bir rəqəmi yoxlayırıq. əgər cütdürsə onu $m=p*q+m$ düsturu ilə m-ə əlavə edirik, burada p 10-nun qüvvətidir.
2	<pre>s=input() l=len(s) n="" for i in range(l): if int(s[i])%2==0: n=n+s[i] print(int(n))</pre>	Soldan başlayaraq bütün rəqəmləri yoxlayırıq və əgər cütdürsə onu n mətninə əlavə edirik.
3	<pre>s=list(input()) n=[a for a in s if int(a)%2==0] print(int("".join(n)))</pre>	Rəqəmlərdən ibarət siyahı Yalnız cüt ədədlərdən ibarət yeni siyahı yaradırıq. ".join(n) ilə siyahını bir mətnə birləşdiririk Mətn şəkilli ədədi tam ədədə çeviririk Çap edirik
Yuxarıdakı hər üç proqramı icra edib 527014 ədədini daxil etsək nəticədə 204 ədədini əldə edərik.		

2.7. N natural ədədi verilib. Ədədi təşkil edən rəqəmləri elə düzmək lazımdır ki, mümkün ən böyük ədəd alınsın. Məsələn, 25317623 ədədi verilibsə onda onun rəqəmlərini 76533221 şəklində düzməklə ən böyük ədəd alırıq.

Nö	Həlli	İzahı
	<pre>s= list(input()) s=sorted(s,reverse=True) n=int("".join(s)) print(n)</pre>	Siyahını azalan sıra ilə çeşidləyin/nizamlayın
	<pre>print(int("".join(sorted(list(input()),reverse=True))))</pre>	Eyni proqram cəmi bir sətirdə.

2.8. $m < n$ ($100 \leq m, n \leq 999$) natural ədədləri verilib. Verilmiş m və n ədədləri (özləri də daxil olmaqla) aralığındakı bütün üçrəqəmli Armstrong ədədlərini tapın. Üçrəqəmli ədəd o vaxt Armstrong ədədi sayılır ki, onun rəqəmlərinin kubları cəmi özünə bərabər olsun. Məsələn: $153 = 1^3 + 5^3 + 3^3$.

Nö	Həlli	İzahı
	<pre>a,b=map(int, input().split()) for i in range(a,b+1):</pre>	İki ədədin bir sətirdə daxil edilməsi

	<pre>a,b,c=i//100, i//10%10, i%10 if a**3+b**3+c**3 == i: print(i)</pre>	<p>a və b aralığındakı bütün ədədləri özləri də daxil olmaqla yoxlayırıq rəqəmləri seçirik əgər rəqəmlərin kübləri cəmi ədədin özünə bərabədirsə çap edirik</p>
--	---	---

2.9. Rəqəmlərinin cəmi k olan bütün N-rəqəmli ədədləri tapın.

Nö	Həlli	İzahı
1	<pre>n,k=map(int, input().split()) a=10**(n-1) for i in range(a,10*a): s=0 m=i while m>0: s = s + m%10 m=m//10 if s==k: print(i,end=' ')</pre>	<p>Bütün n-rəqəmli ədədləri yoxlayırıq</p> <p>Rəqəmlərin cəmini hesablayırıq</p> <p>Cəmi k ilə müqayisə edirik, əgər k-ya bərabədirsə ədədi çap edirik.</p>
2	<pre>n,k=map(int, input().split()) a=10**(n-1) for i in range(a,10*a): s=[int(b) for b in list(str(i))] if sum(s)==k: print(i,end=' ')</pre>	<p>ədədi siyahıya çeviririk. sum(s) – siyahının elementləri cəmini hesablayıb müqayisə edirik</p>
Əgər 3 və 5 ədədlərini daxil etsək 104 113 122 131... 500 nəticəsini əldə edərək		

Nö	Həlli	İzahı