

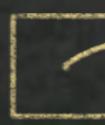
(Sentinel node)

لیست مسحی
لیست از $\underline{n} - 1$ عکان

و $\underline{1}$ را که از طرفی از سری \underline{n} عکان $\underline{-2}$

(Cursor) لیست \underline{n}

L.head

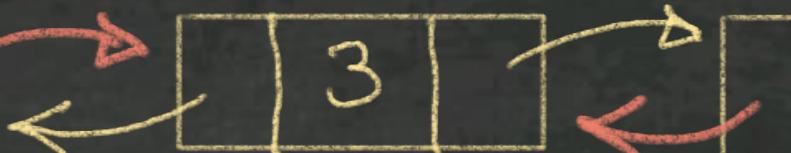


1 | 4 |

2 | 3 |

3 | 5 |

4 | 1 |



Delete(L, x)

if $x.\text{prev} \neq \text{NIL}$

$x.\text{prev}.\text{next} = x.\text{next}$

else $L.\text{head} = x.\text{next}$

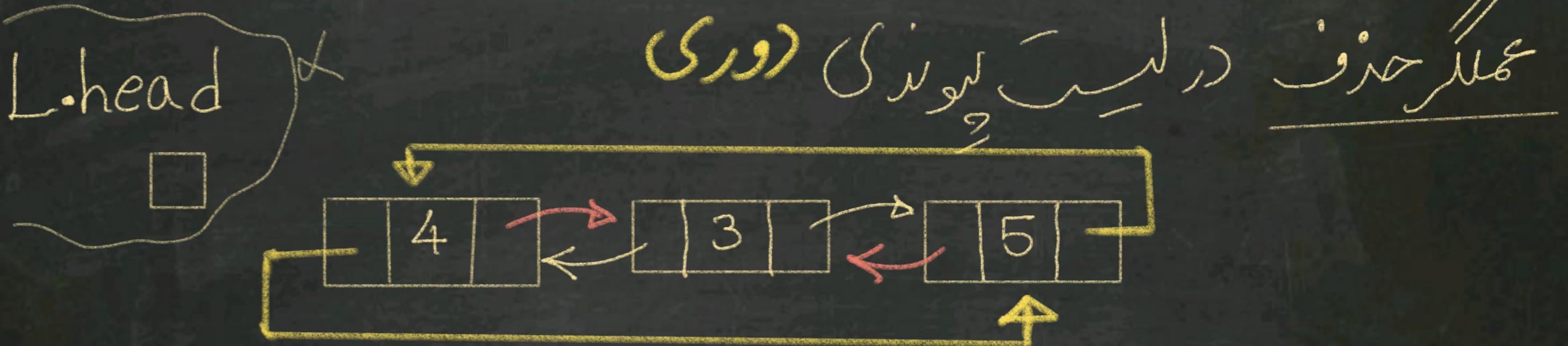
if $x.\text{next} \neq \text{NIL}$

$x.\text{next}.\text{prev} = x.\text{prev}$

else $L.\text{tail} = x.\text{prev}$

x \square $\xrightarrow{\text{Sign } \leftarrow ?}$ sign // 1

\square L.tail



if $x.\text{prev} \neq \text{NIL}$

$x.\text{prev}.\text{next} = x.\text{next}$

else $L.\text{head} = x.\text{next}$

if $x.\text{next} \neq \text{NIL}$

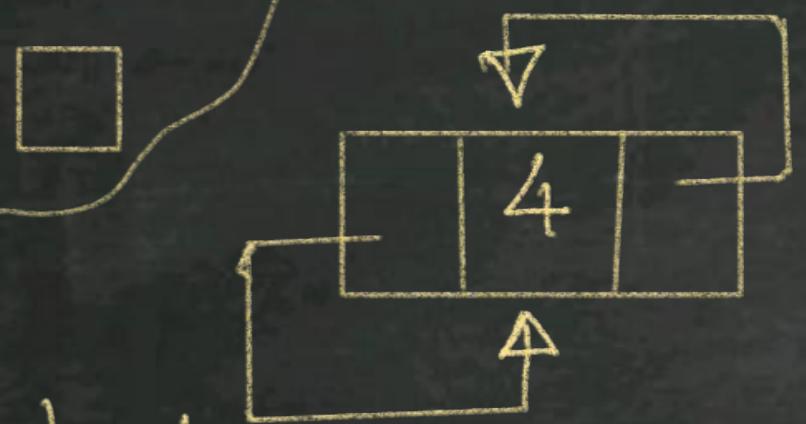
$x.\text{next}.\text{prev} = L.\text{prev}$

else $L.\text{tail} = x.\text{Prev}$

سیگنال
جایگزین
کوچک
بزرگ
میانی
کوچک
بزرگ
میانی
کوچک
بزرگ
میانی

وری سیلیکن، جو نویس

L.head



Delete(L, x)

if $x.\text{prev} \neq \text{Nil}$

$$x.\text{prev}.\text{next} = x.\text{next}$$

✓ $\rightarrow a = a$

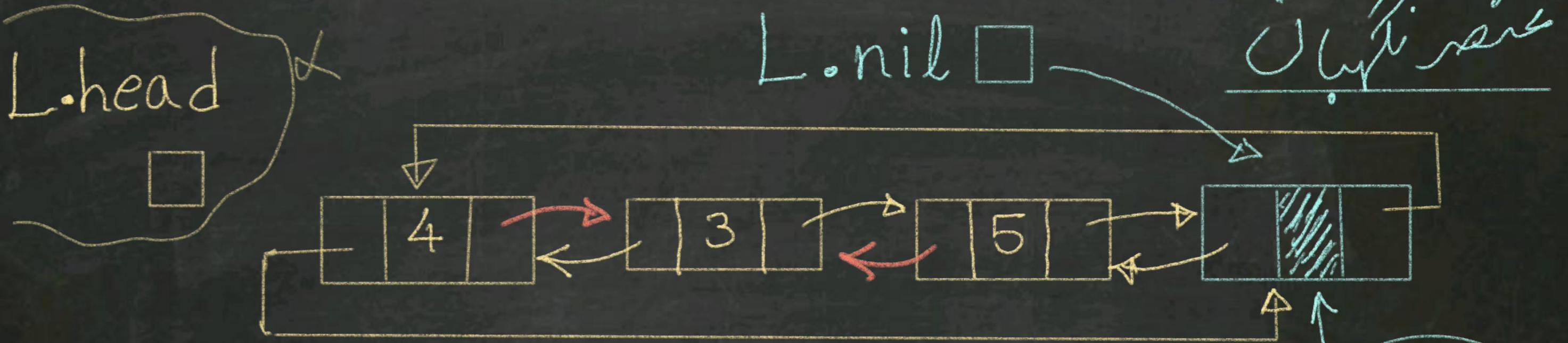
else $L.\text{head} = x.\text{next}$

if $x.\text{next} \neq \text{Nil}$

$$x.\text{next}.\text{prev} = x.\text{prev}$$

✓ $\rightarrow a = a$

else $L.\text{tail} = x.\text{Prev}$



~~if $x.\text{prev} \neq \text{NIL}$~~

$x.\text{prev}.\text{next} = x.\text{next}$

~~else $L.\text{head} = x.\text{next}$~~

~~if $x.\text{next} \neq \text{NIL}$~~

$x.\text{next}.\text{prev} = x.\text{prev}$

~~else $L.\text{tail} = x.\text{prev}$~~

✓ class List{

node *nil;

List(){

 nil = new node;

}

 nil->prev=nil->next=nil;

 node *head(){

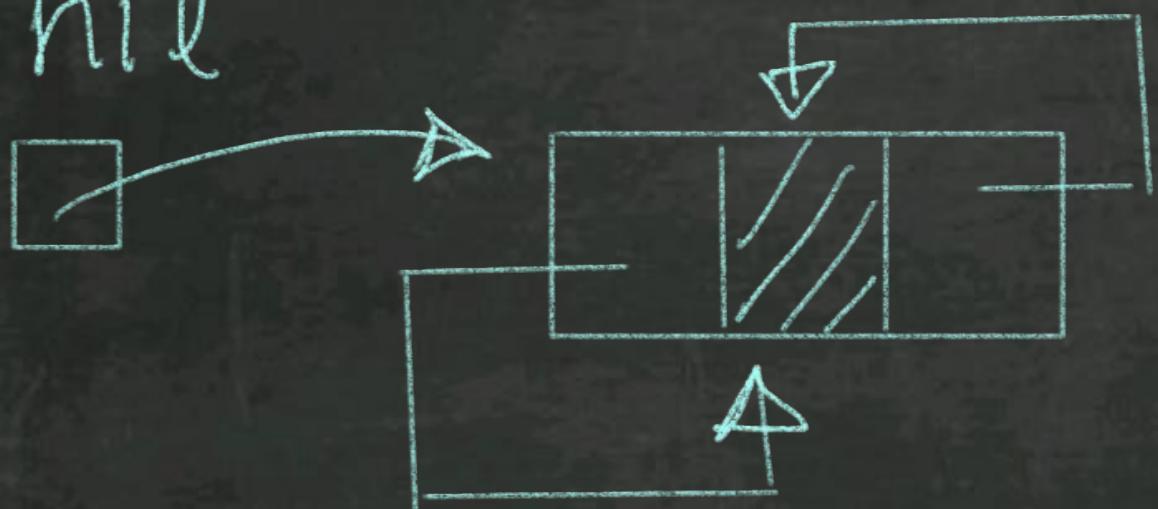
 return nil->next;

}

بایه می باشد
Data
List

بـ سـ بـ لـ كـ فـ رـ دـ لـ لـ مـ لـ كـ لـ جـ لـ سـ لـ مـ

L.nil



أـ لـ وـ حـ صـ

لـ سـ كـ طـ وـ اـ رـ

لـ سـ كـ طـ وـ اـ رـ

isEmpty(){

return nil->next == nil;

}

برای یک لیست کوئنری (وری) با عصرهای لینک

۲

Print(L)

$x = L.\text{head}()$

while $x \neq L.\text{nil}$

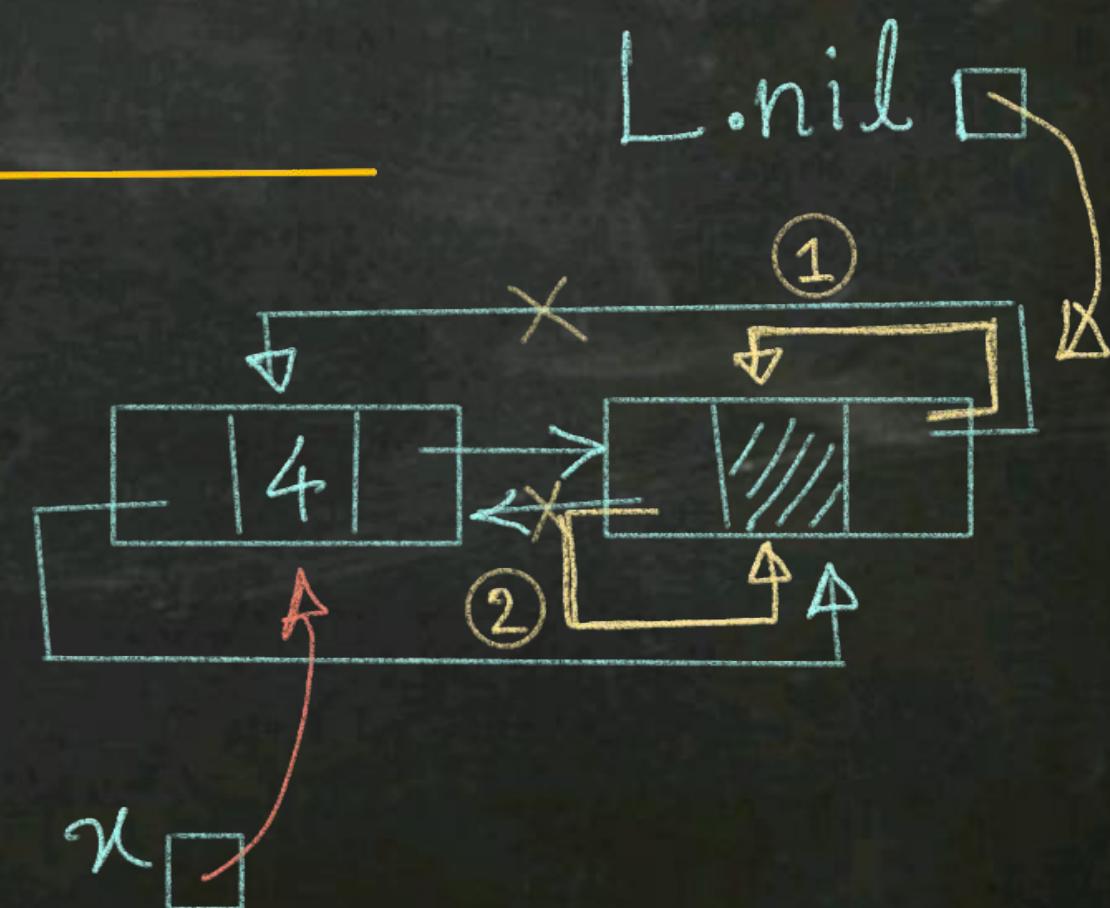
Print(x.data)

$x = x.\text{next}$

Delete(L, x)

1 $x.\text{prev}.\text{next} = x.\text{next}$

2 $x.\text{next}.\text{prev} = x.\text{prev}$



2

```

class node{
    Type data;
    int prev;
    int next;
}

```

CURSOR
VS
Pointer

پیکر کو سطح اور لایه لایه پیشی کرنے کی تکنیک

کسی دلیل کی وجہ سے (ف) و مذکورہ طوفان

تمام کو سطح خود برناہم الگام سورہ

لز طوفی اسی (هار زینت) اڑائے بزرگ

- بجای اشارہ کر، اندرس (اصح نہیں)

مولفہ کی مختلف یہ اڑائے بزرگ را در لطفی کریں

static node mem[100000];

• كل القيم في mem

STATIC global variable

mem

	0	1	2	3	4	5	6	7	8	9
Prev	5	6	-1	/\//\	0	2	-1	/\//\	1	/\//\
data	c	y	a	/\//\	d	b	x	/\//\	z	/\//\
next	4	8	5	/\//\	-1	0	1	/\//\	-1	/\//\

$L_1 \cdot \text{head} = 2$

$L_2 \cdot \text{head} = 6$

MemSize = 10

~) فرآیندی را که در آن از آن داده شده است باز خواهد گشت.

Print(L)

$x = L \cdot \text{head}$

while $x \neq -1$

-1

→ NIL

Print(mem[x] · data)

$x = \text{mem}[x] \cdot \text{next}$

نیاز اوری . کوچک و مدرک حافظه با نیاز کوچک خود بینانه انجام می‌شود .

عملدر در ارتباط با مدرک حافظه میزان است :

InitializeMem

(۱) معرفی اولیه حافظه

Allocate ^۹ / کمیسیون فضای حافظه کی کیجیع

^۹ / ^{۱۰} ^{۱۱} ^{۱۲} ^{۱۳} افزایش درین فضای حافظه کمیسیون را داد

Free

mem

	0	1	2	3	4	5	6	7	8	9
Prev	5	6	-1	0	0	2	-1	0	1	0
data	c	y	a	0	d	b	x	0	z	0
next	4	8	5	7	-1	0	1	9	-1	-1

Allocate()

if free == -1

 throw exception("out of mem");

x=free

free = mem[free].next

int free 7

return x

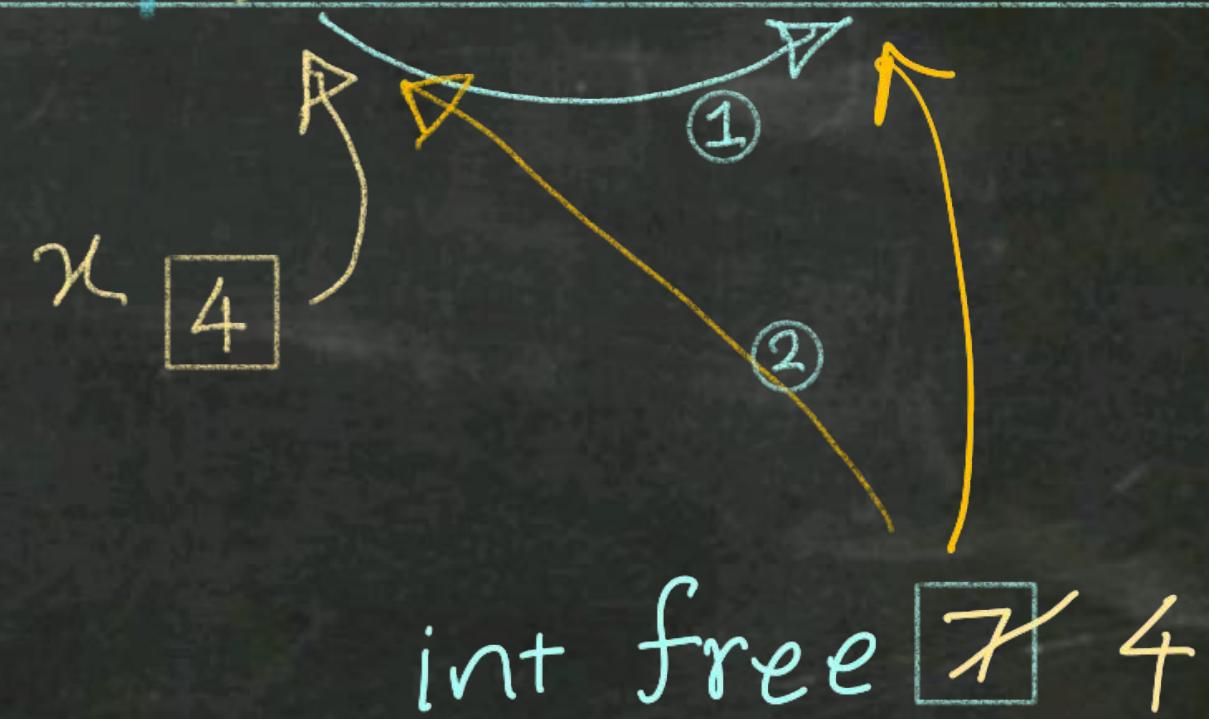
mem

	0	1	2	3	4	5	6	7	8	9
Prev	5	6	-1	-1	0	2	-1	0	1	0
data	c	y	a	h	d	b	x	z	w	v
next	4	8	5	-1	7	0	1	9	-1	-1

Free(x)

1 mem[x].next = free

2 free = x



InitializeMem()

لیبرایری رفعات در این سری سروع

for $i = 1$ to MemSize-1

mem[i-1].next = i

mem[MemSize-1].next = -1

free = 0

mem

	0	1	2	3	4	5	6	7	8	9
Prev	/\//	/\//	/\//	/\//\//	/\//\//	/\//\//	/\//\//	/\//\//	/\//\//	/\//\//
data	/\//\//	/\//\//	/\//\//	/\//\//	/\//\//	/\//\//	/\//\//	/\//\//	/\//\//	/\//\//
next	1	2	3	4	5	6	7	8	9	-1

free 0 