

Problem: Count how many vowels in a string

Draft: look at each letter,  
increment a counter if letter is {a,e,i,o,u} vowel.

```
for x  
    accumulator.  
counter = 0  
for letter in s :  
    if letter in {'a','e','i','o','u'} :  
        counter += 1
```

Problem: given a number n , generate a list with n random numbers in the range [0..1000]

Draft

- ask for n
- start with an empty list
- repeat n times  
Pick a random number  
add it to the list.

```
index → for i in range()  
  
n = int(input('n?'))  
l = []  
for i in range(n) :  
    x = random.randint(0,1000)  
    l.append(x)
```

Problem: Generate a random list of n numbers, find the largest number in the list which is divisible by 7. If no such number exists print '0'

Draft: look at each element.  
if element is divisible by 7 AND  
the largest so far:  
remember it!

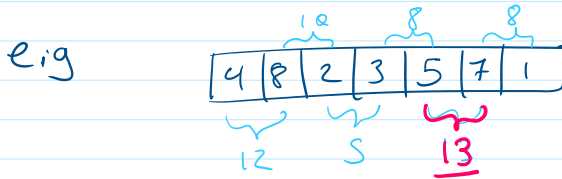
```
best_x = 0
```

```

for x in l :
    if x % 7 == 0 and x > best_x :
        best_x = x

```

Problem: From a list of n random numbers, find the two consecutive numbers that with maximum sum.



Draft:

look at each element x in the list  
 sum x with the next element  
 remember the pair that adds to the most.

for each position i in the list (except the last one)

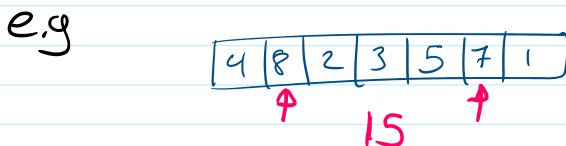
- sum the element at position i with the element at position i+1
- remember the pair that adds to the most

```

best_sum = 0
best_pair = (-1,-1)
for i in range( len(l)-1 ) :
    s = l[i] + l[i+1]
    if s > best_sum :
        best_sum = s
        best_pair = (l[i],l[i+1])
print(best_pair)

```

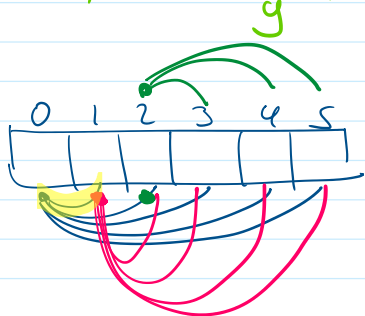
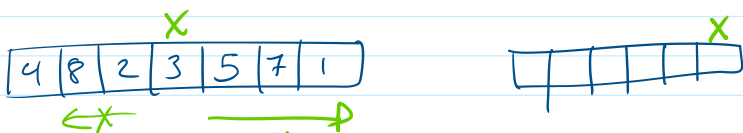
Problem: From a list of n random numbers, find the two numbers with maximum sum.



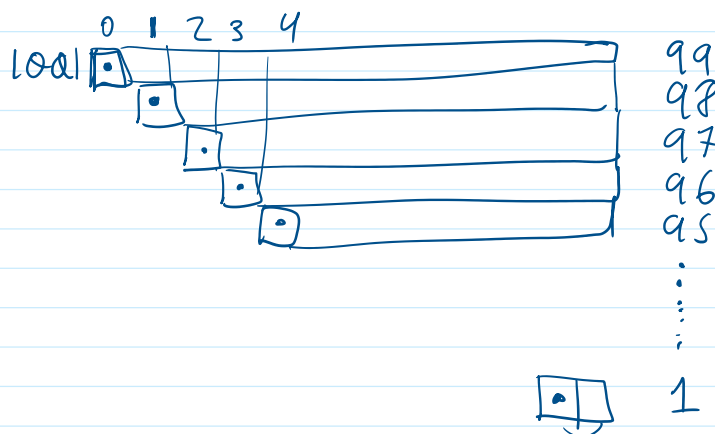
for every number x in the list  
 for every number y in the list.  
ahead of x

- compute sum
- remember best pair.

- compute sum
- remember best pair.



x)



$$1 + 2 + 3 + 4 + 5 + \dots + 97 + 98 + 99 = 100 \cdot 50 = \underline{5000}$$

Problem: Print the first n prime numbers.

prime: a number that is divisible only by 1 and by itself

draft:

- pick a number x
- check if any number smaller than x, divides x
- if so, x is not prime.
- else x is prime, print x, increment count.
- pick the next x, repeat.

draft #2:

counter = 0  
x = 2

```

counter = 0
x = 2
while counter < n :
    # test x for primality.
    z = 2
    while z < x and z does not divides x :
        increment z.
    if z == x
        print (x)
        increment count.
    increment x

```

```

n = int(input('n? '))
counter = 0
x = 2
while counter < n :
    #print(f'{x=}')
    # test x for primality
    z = 2
    while x % z != 0 :
        #print(f'{z=}')
        z += 1
    if z == x :
        print(x, 'is prime')
        counter += 1
    x += 1

```

### Problem: Payment Calculator:

Example: Bob buys a computer, \$1,000  
at a 1.5% monthly interest.

Bob pays \$50 every month.

How much will Bob pay for his computer?

	interest	debt	balance
	1.5%	\$1,000	
1	\$15	\$1,015	\$965
2	\$14.8	\$979.8	\$929.8
3			

write a function that given  $\rightarrow$  dept  
 $\rightarrow$  interest  
 $\rightarrow$  payment

Write a function that given  $\leftarrow$  interest  
computes total amount paid.  $\rightarrow$  payment

Draft:

input: dept, interest, payment  
rate.

total-payment = 0

while dept > 0:

int = dept \* interest-rate

balance = dept + int - payment.

total-payment = payment.

print, item in the loop:

return total-payment.

(code)

— 0 —