Welcome to the Back End Engineering Test - you have 3 hours to complete it. The first three tasks are mandatory, the fourth and the fifth are optional, do them if you have enough time. Please return only the source code of the solution to me by copying it from your favourite IDE, and pasting it to the Google Docs document I am sending you.

Tasks can be solved in any of the programming languages Java, Go, JavaScript, Python, Scala or C / C ++, without the use of a framework. Please do not use functions or libraries that solve a given problem in one or two lines of code. Comments within the code are welcome.

We would appreciate it if you would write an explanation of your idea on how you would solve the task in English, before you start writing the actual code.

Good luck!

1. Write a method or a function that finds an element in a linked list, that is **n** places away from the end of the list, ideally with just one pass through the list.

2. Write a method or a function that merges two sorted arrays, A and B, into A. The array A is initialized with a sufficient number of elements so that all values from A and B can be accommodated. Values of 0 represent "empty" elements, placed at the end of the array A, that can be filled in. The array A should contain all the elements from A and B in sorted order. The sort order of the array A should not be changed.
EXAMPLE:
Input A = [-1, 15, 0, 0, 0, 0] (effectively A is [-1, 15])
B = [4, 3, 2, 1]
The result is A = [-1, 1, 2, 3, 4, 15]

3. Write a method or a function that finds the first character that has exactly 2 occurrences in the input string.
EXAMPLE:
ABAACCBA-> B
ABC> ''
ABBCCA-> A

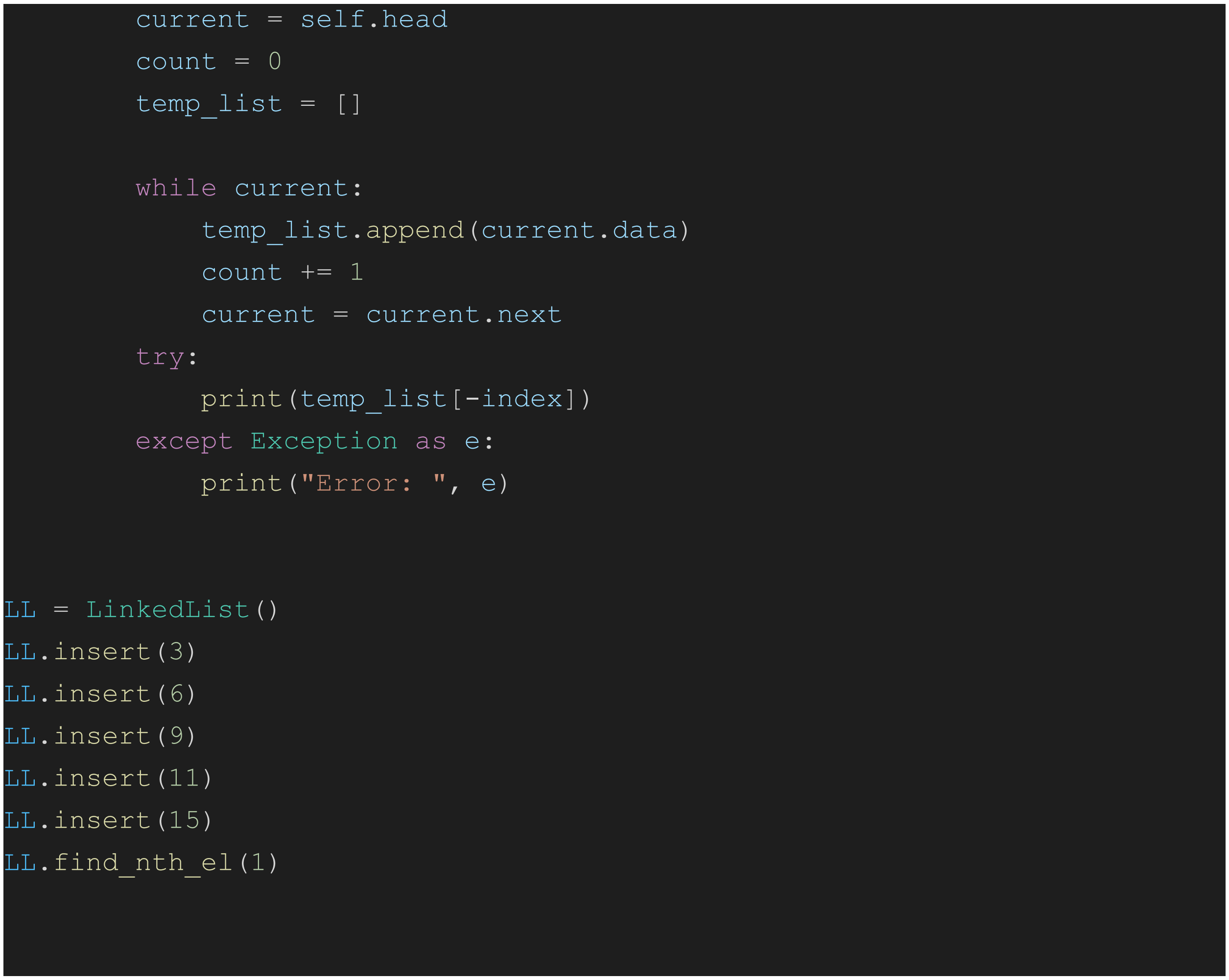
4. Write a method or a function that extracts 530000 elements from the input array containing 900000 elements, which are evenly distributed in the input set.
- as an illustration, let's say that the input data set contains 9 elements (x1, x2, ..., x9).

- possible solutions would be: (x1, x3, x5, x7, x9), (x1, x3, x4, x6, x8, x9) ... where a larger number of solutions would be a set of 5 elements (a certain number of solutions would have 6 elements).
So it is important that all the elements of the resulting set are **evenly** distributed in the input set and that there is an **appropriate** number of them.

5. Write a method or a function that returns words in reverse order for a given text. Words are considered to be groups of letters separated by a whitespace character. The first word of a new sentence should start with a capital letter and sentences should be separated with full stop ("."). All other words except the first word of the input sentence should retain their initial form. Please do not use built-in string helper methods, like for example split, replace or reverse.
EXAMPLE:
Agree with you the council does. Your apprentice Skywalker will be.
Does council the you with agree. Be will Skywalker apprentice your.

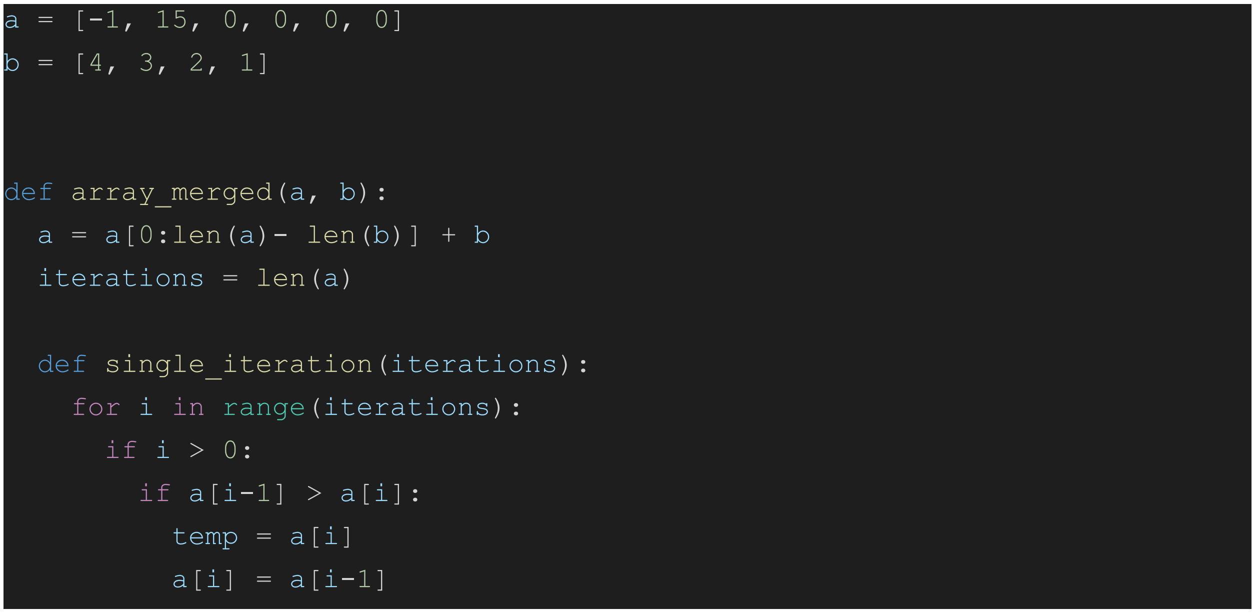
Task1:

```
class Node:
    def __init__(self, data, next=None):
        self.data = data
        self.next = next

class LinkedList:
    def __init__(self):
        self.head = None

    def insert(self,data):
        newNode = Node(data)
        if self.head:
            current= self.head
            while(current.next):
                current = current.next
            current.next = newNode
        else:
            self.head = newNode

    def find_nth_el(self, index):
```



```
        current = self.head
        count = 0
        temp_list = []

        while current:
            temp_list.append(current.data)
            count += 1
            current = current.next
        try:
            print(temp_list[-index])
        except Exception as e:
            print("Error: ", e)


LL = LinkedList()
LL.insert(3)
LL.insert(6)
LL.insert(9)
LL.insert(11)
LL.insert(15)
LL.find_nth_el(1)
```

Task2:



```
a = [-1, 15, 0, 0, 0, 0]
b = [4, 3, 2, 1]



def array_merged(a, b):
  a = a[0:len(a)- len(b)] + b
  iterations = len(a)

  def single_iteration(iterations):
    for i in range(iterations):
      if i > 0:
        if a[i-1] > a[i]:
          temp = a[i]
          a[i] = a[i-1]
          a[i-1] = temp

    return a

  for i in range(len(a)):
    a = single_iteration(iterations)
    iterations -= 1

  return a


print(array_merged(a, b))
```

Task3:

```
a = 'ABAACCBA'    # -> B
b = 'ABC'        # -> ''
c = 'ABBCCA'     #-> A

def find_occur(input, occur):
    dict_char = {}
    for i in input:
        if i not in dict_char:
            dict_char[i] = 1
        else:
            dict_char[i] += 1

    for key,value in dict_char.items():
        if value == occur:
            return key

print(find_occur(a, 2))
```

Task4:

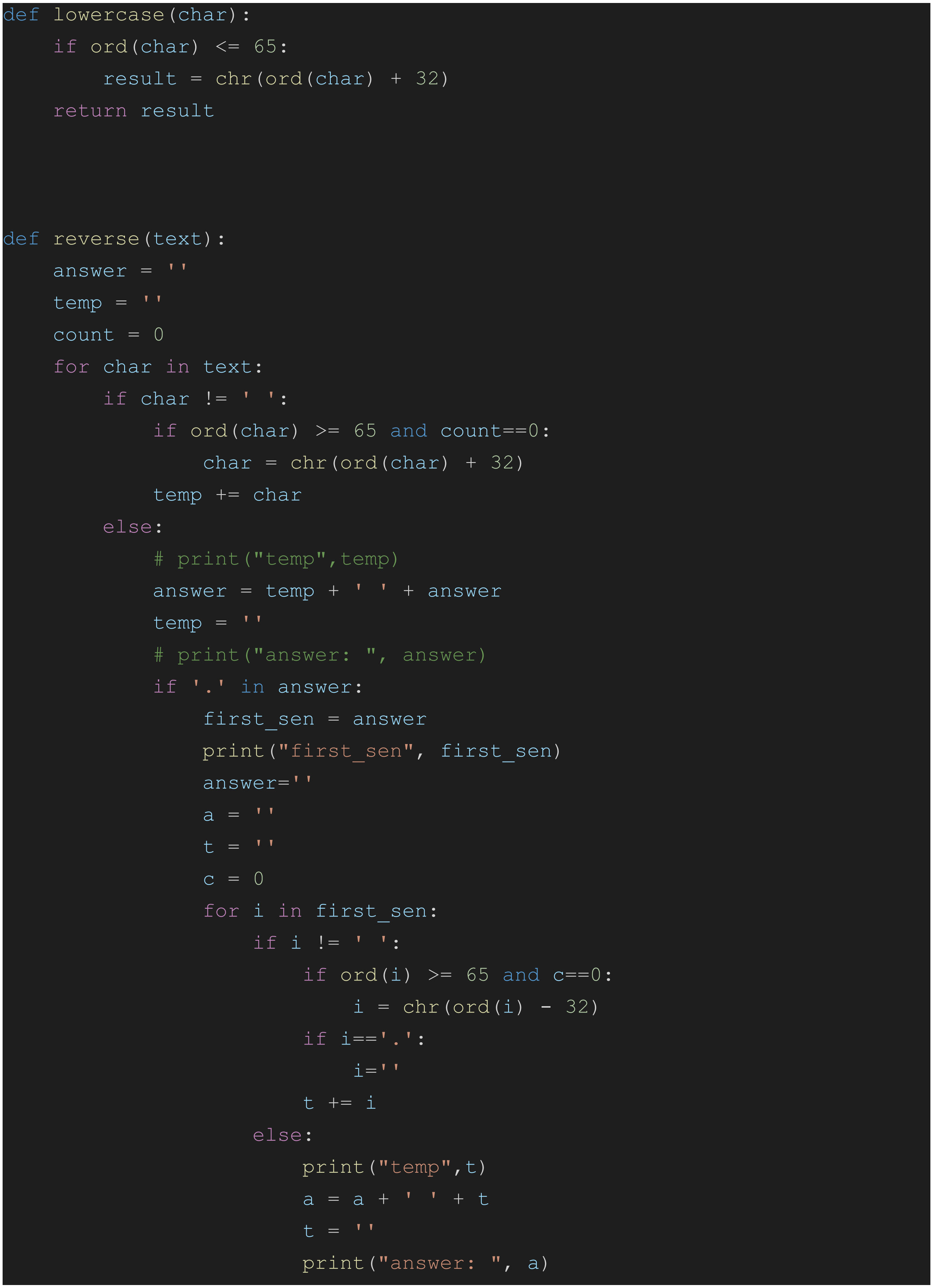
```
from math import ceil
lst = list(range(900000))
```

```
def extract_evenly(lst, extract_number):
  elem_number = len(lst)
  div = 2
  divs = [2]
  lst_out = []

  if elem_number/2 > extract_number:
    return lst[::2]
  else:
    while elem_number % extract_number != 0:
      res0 = elem_number/div
      extract_number = extract_number - res0
      div = int(ceil(elem_number / extract_number))
      divs.append(div)

  for i in divs:
    lst_out += lst[::i]

  return lst_out

print(len(extract_evenly(lst, 530000)))
```

Task5:

Note: Task5 is not finished completely. Minor changes needs to be implemented.

```
A = 'Agree with you the council does. Your apprentice Skywalker will be.'
# Does council the you with agree. Be will Skywalker apprentice your.


def uppercase(char):
    if ord(char) >= 65:
        result = chr(ord(char) - 32)
    return result
```



```
def lowercase(char):
    if ord(char) <= 65:
        result = chr(ord(char) + 32)
    return result



def reverse(text):
    answer = ''
    temp = ''
    count = 0
    for char in text:
        if char != ' ':
            if ord(char) >= 65 and count==0:
                char = chr(ord(char) + 32)
            temp += char
        else:
            # print("temp",temp)
            answer = temp + ' ' + answer
            temp = ''
            # print("answer: ", answer)
            if '.' in answer:
                first_sen = answer
                print("first_sen", first_sen)
                answer=''
                a = ''
                t = ''
                c = 0
                for i in first_sen:
                    if i != ' ':
                        if ord(i) >= 65 and c==0:
                            i = chr(ord(i) - 32)
                        if i=='.':
                            i=''
                        t += i
                    else:
                        print("temp",t)
                        a = a + ' ' + t
                        t = ''
                        print("answer: ", a)
                    c+=1
                a=a+'. '
        count+=1
    answer = temp + ' ' + answer
    answer = a + answer
    print(answer)

reverse(A)
```