```python
# 3N+1
i,j=input("enter the numbers seprated by space").
split()
m=int(i)
j=int(j)
maxlist=[]
max=0
number=j
for i in range(m,j+1):
    number=i
    list_of_number=[i]
    while(number>1):
        if number%2==0:
            number=number/2
        else:
            number=number*3+1
        list_of_number.append(number)
        if max<len(list_of_number):
            max=len(list_of_number)
            maxlist=list_of_number

print("number having max list =",maxlist[0])
print(maxlist)
print(m,j,len(maxlist))


"""                                          OUTPUT
C:\Users\HP\AppData\Local\Programs\Python\Python310\
python.exe D:/other/python/cp/3n+1.py
enter the numbers seprated by space 1 10
number having max list = 9
[9, 28, 14.0, 7.0, 22.0, 11.0, 34.0, 17.0, 52.0, 26.0
, 13.0, 40.0, 20.0, 10.0, 5.0, 16.0, 8.0, 4.0, 2.0, 1
.0]
1 10 20

Process finished with exit code 0
"""
```

```
 1 # cardwar
 2 #input of both players cards
 3 deck1=list(input("enter the deck 1 cards").split())
 4 deck2=list(input("enter the deck 2 cards").split())
 5 rank_of_deck1=[]
 6 rank_of_deck2=[]
 7
 8 # it removes the first  element of the argument(deck
   ) given and returns that removed card
 9
10 def dequeue(deck):
11     if len(deck)==0:
12         return False
13     value=deck[0]
14     del deck[0]
15     return value
16
17
18 # check if given deck is empty or not
19 #returns  true if deck is empty and false vice verse
20
21 def check_empty(deck):
22     if len(deck)==0:
23         return True
24     else:
25         return False
26
27
28 ncards = 52
29 nsuits = 4
30 values = "23456789TJQKA"
31 suits = "cdhs"
32
33 #this fucntion find rank of cards
34 def rankofcards(value,suit):
35     for i in range(0, int(ncards / nsuits)):
36         if values[i] == value:
37             for j in range(0, nsuits):
38                 if suits[j] == suit:
39                     return i * nsuits + j
40
```

```
41 #this give value of the card for ex=(2,3,4,5 etc)
42 def value(rank):
43     return rank // nsuits
44
45
46 #note that
47 # the input deck are not used here ex=(4d,3d,7h)
48 #the ranks are calculated of each card and that rank
   list is used as arguemnts in this war functionn
49
50 def war(rank_of_deck1,rank_of_deck2):
51     steps=0
52     # a blank list (collected_cards)is used for
   temporary storing the  top cards of each players
53     collected_Cards=[]
54     # while condition check if any one pf player have
    empty deck and repeats itself until one of then have
    lost all cards
55     while(not check_empty(rank_of_deck1) and not
   check_empty(rank_of_deck2)):
56         steps+=1
57         if(steps>200):
58             print("match tied")
59             break
60         # the top card of each palyer is deleted from
    their deck and stored in the (collected card) list
61         # and in variable card1 and card 2
62         card1=dequeue(rank_of_deck1)
63         collected_Cards.append(card1)
64
65         card2=dequeue(rank_of_deck2)
66         collected_Cards.append(card2)
67         # now we hav rank of both top card
68         #using value function we nor calculate value
   of card
69         # for ex if rank is 9 its value will be 9//4=
   2 and compare them
70         # not if value are same both if will not
   execute and next iteration will occur
71         # if both are same again next 2 card will be
   drawn and  comapred
```

```
 72
 73         if value(card1) > value(card2):
 74             # if card1 is> card2 means palyer 1 have
    higher card so he will now take the collected cards
    thus
 75             # we transfer the collected cards to him
    so palyer 1 will have more card than palyer 2
 76             # note:- add card at end of queue(first
   in first out)
 77             transfer(collected_Cards,1)
 78         if value(card1) < value(card2):
 79             transfer(collected_Cards,2)
 80     # after loop
 81     # check who has lost all cards and empty deck
   holder is loser
 82     if check_empty(rank_of_deck1)==True and
   check_empty(rank_of_deck2)==False:
 83         print("palyer 2 wins")
 84     if check_empty(rank_of_deck1)==False and
   check_empty(rank_of_deck2)==True:
 85         print("player 1 wins")
 86
 87
 88
 89 def transfer(collected_cards,deck_number):
 90     if deck_number==2:
 91         for i in range(0,len(collected_cards)+1):
 92             a=dequeue(collected_cards)
 93             if a!=False:
 94                 rank_of_deck2.append(a)
 95     else:
 96         for i in range(0, len(collected_cards) + 1):
 97             a = dequeue(collected_cards)
 98             if a!=False:
 99                 rank_of_deck1.append(a)
100
101
102
103
104 for i in deck1:
105     rank_of_deck1.append(rankofcards(i[0],i[1]))
```

```
106 for i in deck2:
107     rank_of_deck2.append(rankofcards(i[0],i[1]))
108 war(rank_of_deck1,rank_of_deck2)
109
110
111 """                 OUTPUT
112 C:\Users\HP\AppData\Local\Programs\Python\Python310\
    python.exe D:/other/python/cp/manascardwar.py
113 enter the deck 1 cards8d 8c 9c 7c 5d 4c Js Qc 5s Ts
    Jc Ad 7d Kh Tc 3s 8s 2d 2s 5h 6d Ac 5c 9h 3d 9d
114 enter the deck 2 cards4d Ks As 4h Jh 6h Jd Qs Qh 6s
    6c 2c Kc 4s Ah 3h Qd 2h 7s 9s 3c 8h Kd 7h Th Td
115 match tied
116 Process finished with exit code 0
117 ca
118
119
120 C:\Users\HP\AppData\Local\Programs\Python\Python310\
    python.exe D:/other/python/cp/manascardwar.py
121 enter the deck 1 cards4d Ks As 4h Jh  Qs
122 enter the deck 2 cards8d 8c 9c 7c 5d 4c
123 player 1 wins
124
125 Process finished with exit code 0
126 """
```

```python
#CORPORATE RENAME
dict1={}
changed_values=[]
while True:
    string1=input()
    list1=string1.split(" to ")
    if len(list1[0])<40 and len(list1[-1])<40:
        dict1[list1[0]]=list1[-1]
        changed_values.append(list1[0])
    else:
        input_string=string1
        break

for i in changed_values:
    if i in input_string:
        input_string = input_string.replace(i,dict1[i
])

print(input_string)

"""                              output
Anderson Consulting to Accenture
Enron to Dynegy
DEC to Compaq
TWA to American
Anderson Accounting begat Anderson Consulting, which
offered advice to Enron before it DECLARED bankruptcy
,which made AndersonConsulting quite happy it changed
 its namein the first place"""
```

File - D:\other\python\cp\editor2.py

```python
# Grpahical Editor

output_matrix = []
commands_list = []

while True:
    commands = list(map(str, input().split()))
    commands_list.append(commands)
    if commands[0] == "X":
        break

for commands in commands_list:
    if commands[0] == "I":
        row = int(commands[2])
        column = int(commands[1])
        for i in range(row):
            a = []
            for j in range(column):
                a.append("O")
            output_matrix.append(a)

    elif commands[0] == "L":
        output_matrix[int(commands[2]) - 1][int(commands[1]) - 1] = commands[3]

    elif commands[0] == "V":
        column_number = int(commands[1]) - 1
        for i in range(int(commands[2]) - 1, int(commands[3])):
            output_matrix[i][column_number] = commands[4]

    elif commands[0] == "H":
        row_number = int(commands[3]) - 1
        for i in range(int(commands[1]) - 1, int(commands[2])):
            output_matrix[row_number][i] = commands[4]

    elif commands[0] == "F":
        for i in range(row):
```

```
            for j in range(column):
                output_matrix[i][j] = commands[3]

    elif commands[0] == "S":
        file_name = ""
        file_name = commands[1]
        print(file_name)
        for i in range(row):
            for j in range(column):
                print(output_matrix[i][j], end=" ")
            print()

    elif commands[0] == "X":
        break
    else:
        continue

""" OUTPUT
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe D:/other/python/cp/editor2.py
I 5 6
L 2 3 A
S one.bmp
G 2 3 J
F 3 3 J
V 2 3 4 W
H 3 4 2 Z
S two.bmp
X
one.bmp
O O O O O
O O O O O
O A O O O
O O O O O
O O O O O
O O O O O
two.bmp
J J J J J
J J Z Z J
J W J J J
J W J J J
```

```
77 J J J J J
78 J J J J J
79
80 Process finished with exit code 0
81
82 """
83
```

```
 1 #output
 2 week=["sun","mon","tue","wed","thu","fri","sat"]
 3
 4 number_of_parties=int(input("enter number of parties"
   ))
 5 cycle=int(input("enter cycle for counting"))
 6 working_days_missed=0
 7 hartal_days_list=[]
 8 for i in range(0,number_of_parties):
 9     hartal_days_list.append(int(input()))
10 hartal_cycle=[]
11 count_cycle=[]
12 for i in range(0,cycle):
13     hartal_cycle.append(week[i%7])
14     count_cycle.append(0)
15
16 for i in hartal_days_list:
17     increment=i
18     while(True):
19         count_cycle[i-1]=1
20         if hartal_cycle[i-1]=="fri" or hartal_cycle[i
   -1]=="sat":
21             count_cycle[i-1]=0
22         i+=increment
23         if i>cycle:
24             break
25 for i in count_cycle:
26     if i==1:
27         working_days_missed+=1
28 print(hartal_cycle)
29 print(count_cycle)
30
31 print("total working  days missed",
   working_days_missed)
32
33
34
35 """                              OUTPUT
36 C:\Users\HP\AppData\Local\Programs\Python\Python310\
   python.exe D:/other/python/cp/hartal.py
37 enter number of parties1
```

```
enter cycle for counting14
3
['sun', 'mon', 'tue', 'wed', 'thu', 'fri', 'sat', '
sun', 'mon', 'tue', 'wed', 'thu', 'fri', 'sat']
[0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0]
total working  days missed 3

Process finished with exit code 0
"""


```

```python
#HOW MANY FIBS
a,b=input().split(" ")
a=int(a)
b=int(b)
series=[]
firstnumber=1
secondnumber=2
thirdnumber=3
while(thirdnumber<=b):
    thirdnumber=firstnumber+secondnumber
    if(thirdnumber>=a) and (thirdnumber<=b):
        series.append(thirdnumber)
    firstnumber=secondnumber
    secondnumber=thirdnumber

print((series))
print(len(series))

"""                    OUTPUT
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe "D:/other/python/cp/how many fibs.py"
10 150
[13, 21, 34, 55, 89, 144]
6

Process finished with exit code 0

"""
```

```python
#RANK
values = "23456789TJQKA"
suits = "cdhs"
nsuits = 4

def rank_card(value, suit):
    try:
        return values.index(value) * nsuits + suits.index(suit)
    except:
        return False

card=input()
rank=rank_card(card[0],card[1])
print(rank)

"""                         OUTPUT
  C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe D:/other/python/cp/rank.py
3c
4

Process finished with exit code 0
"""
```

```python
#reverse and add
list_of_numbers=[]
countlist=[]
palindromelist=[]

numberofinput=int(input())
for i in range(0,numberofinput):
    list_of_numbers.append(int(input()))

for a in list_of_numbers:
    count=0
    while(True):
        temp=a
        a=int(str(a)[::-1])
        if temp==a:
            palindromelist.append(a)
            countlist.append(count)
            break
        count += 1
        a=a+temp

for i in range(0,len(countlist)):
    print(countlist[i]," ", palindromelist[i])

    """                    OUTPUT
    C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe "D:/other/python/cp/how many fibs.py"
10 150
[13, 21, 34, 55, 89, 144]
6

Process finished with exit code 0
"""
```

File - D:\other\python\cp\trip.py

```python
# the trip
solution_list = []
while True:
    cost_spend = []
    total_expenditure = 0
    avg_cents = 0
    difference = 0
    give_money = 0
    take_money = 0
    number_of_children = int(input())
    if number_of_children == 0:
        for i in solution_list:
            print("$", i)
        break
    else:
        #input of money spend by students
        for i in range(0, number_of_children):
            cost_spend.append(float(input()))

        #convertion to cents
        for i in range(0, number_of_children):
            cost_spend[i] = cost_spend[i] * 100
            total_expenditure += cost_spend[i]

        #average in cents
        avg_cents = total_expenditure / number_of_children

        #calcuate total money to be given
        #and total money to be taken by each student
        for i in range(0, number_of_children):
            difference = cost_spend[i] - avg_cents
            if difference > 0:
                take_money += int(difference) / 100
            else:
                give_money += abs(int(difference) / 100)

    solution_list.append(max(take_money, give_money))

""" OUTPUT
```

```
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe D:/other/python/cp/trip.py
3
10.00
20.00
30.00
4
15.00
15.01
3.00
3.01
0
$ 10.0
$ 11.99

Process finished with exit code 0
"""
```

```python
def search_the_word(i, j, row_index_add, column_index_add, word):
    word_length = len(word)
    for index in range(word_length):
        if grid[i][j] == word[index]:
            if index != word_length - 1:
                if i < m - 1 and j < n - 1:
                    if (i == 0 and row_index_add == -1) or (j == 0 and column_index_add == -1):
                        break
                    else:
                        i += row_index_add
                        j += column_index_add
                else:
                    break
        else:
            break
    else:
        return True
    return False


def search_first_letter(word):
    adding = (0, 1, -1)
    for i in range(m):
        for j in range(n):
            if grid[i][j] == word[0]:
                for row_add in adding:
                    for column_add in adding:
                        if not (row_add == 0 and column_add == 0):
                            if search_the_word(i, j, row_add, column_add, word):
                                return i, j


no_of_cases = int(input())
output = []
for case_no in range(no_of_cases):
    case_output = []
    m, n = map(int, input().split())
```

```
    grid = []
    for i in range(m):
        grid.append(input().lower())
    k = int(input())
    word_list = []
    for i in range(k):
        word_list.append(input().lower())
    for word in word_list:
        y, x = search_first_letter(word)
        case_output.append(str(y + 1) + ' ' + str(x + 1))
    output.append(case_output)

for case_output in output:
    for each in case_output:
        print(each)
    print()


    """                                 OUTPUT
    C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe D:/other/python/cp/waldorf.py
1
8 11
abcDEFGhigg
hEbkWalDorf
FtyAwaldORm
FtsimrLqsrc
byoArBeDeyv
Klcbqwikomk
strEBGadhrb
yUiqlxcnBjf
4
Waldorf
Bambi
Betty
Dagbert
2 5
2 3
1 2
7 8
```

```
77
78
79 Process finished with exit code 0
80 """
```

File - D:\other\python\cp\querty.py

```
#WERTY
key = {'2': '1', '3': '2', '4': '3', '5': '4', '6': '5', '7': '6', '8': '7', '9': '8', '0': '9', '-': '0', '=': '-',
       'W': 'Q', 'E': 'W', 'R': 'E', 'T': 'R', 'Y': 'T', 'U': 'Y', 'I': 'U', 'O': 'I', 'P': 'O', '[': 'P', ']': '[',
       '\\': ']','S': 'A', 'D': 'S', 'F': 'D', 'G': 'F', 'H': 'G', 'J': 'H', 'K': 'J', 'L': 'K', ';': 'L', '\'': ';',
       'X': 'Z', 'C': 'X', 'V': 'C', 'B': 'V', 'N': 'B', 'M': 'N', ',': 'M', '.': ',', '/': '.'}

a=input("enter the string")
a=a.upper()

for i in a:
    if i==' ':
        print(" ", end="")
    else:
        print(key[i], end="")


"""                             OUTPUT
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe D:/other/python/cp/querty.py
enter the string,smsd smo; npnsfr
MANAS ANIL BOBADE
Process finished with exit code 0
"""
```