Taks-1

```
# input:1 10
# hint :9+1 9+3 9+5 9+7 9+9  9+11 ...
# ouput:10 12 14 16 18 20 22 24 26 28
```

Task-2

```
# input:5
# hint 5^1+1^1 5^2+2^2 5^3+3^3
# output:6 29 152 881 6250
```

Task-3

```
# st="ECE4EEE3CIV2CSE4"
# ouput:ECEECEECEECE
#       EEEEEEEEE
#       CIVCIV
#       CSECSECSECSE
```

In [6]:

```
#Task-3
    # st="ECE4EEE3CIV2CSE4"
    # ouput:ECEECEECEECE
    #       EEEEEEEEE
    #       CIVCIV
    #       CSECSECSECSE

st="ECE4EEE3CIV2CSE4"
branches=[]
values=[]
es=''
for evry_chr in st:
    if evry_chr.isalpha():
        es=es+evry_chr
    else:
        branches.append(es)
        values.append(evry_chr)
        es=''
print(branches)
print(values)


```

```
['ECE', 'EEE', 'CIV', 'CSE']
['4', '3', '2', '4']
```

In [7]:
```
for i in range(len(branches)):
    print(branches[i]*int(values[i]))
```

```
ECEECEECEECE
EEEEEEEEE
CIVCIV
CSECSECSECSE
```

In [8]:
```
for i in range(len(branches)):
    r=branches[i][::-1]
    print(r*int(values[i]))
```

```
ECEECEECEECE
EEEEEEEEE
VICVIC
ESCESCESCESC
```

In [19]:
```
    # input:1 10
    # hint :9+1 9+3 9+5 9+7 9+9  9+11 ...
    # ouput:10 12 14 16 18 20 22 24 26 28
a=9
sum=0
for i in range(1,20):
     if i%2!=0:
            sum=a+i
            print(sum,end=" ")
```

```
10 12 14 16 18 20 22 24 26 28
```

In [17]:
```
# Task-2
    # input:5
    # hint 5^1+1^1 5^2+2^2 5^3+3^3
    # output:6 29 152 881 6250
i=int(input())
for j in range(1,10):
    print(i**j+j**j)
```

```
5
6
29
152
881
6250
62281
901668
17167841
389373614
```

Dictionary

- Dictionary is Collection of Hetrogenous Data Set
  - ex:chr,int,spl char,..etc
    - It is an mutale (we can read elements,remove,modify)

- Here Data in form of Key and value pair
- All keys are unique but not values
- reprasentation is data type is dict
- we can insilize the dict with {}

In [20]:
```
d={"key 1":'value1',"Key 2":"Value2","key 3":"Value3"}
```

In [22]:
```
d1={"cbit":2008,"python":1989,"apssdc":2014}
d1
```

Out[22]:
```
{'cbit': 2008, 'python': 1989, 'apssdc': 2014}
```

In [23]:
```
print(dir(dict))
```

```
['__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc_
_', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__gt_
_', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len_
_', '__lt__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
'__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'cle
ar', 'copy', 'fromkeys', 'get', 'items', 'keys', 'pop', 'popitem', 'setdefaul
t', 'update', 'values']
```

In [24]:
```
#Geeting all keys
d1.keys()

```

Out[24]:
```
dict_keys(['cbit', 'python', 'apssdc'])
```

In [25]:
```
#values
d1.values()
```

Out[25]:
```
dict_values([2008, 1989, 2014])
```

In [26]:
```
d1['Vbit']=2008  #Adding New elements in to dict
```

In [27]:
```
d1
```

Out[27]:
```
{'cbit': 2008, 'python': 1989, 'apssdc': 2014, 'Vbit': 2008}
```

In [28]:
```
d1['python']=1991 #Updating dict value
```

In [29]:
```
d1
```

Out[29]:
```
{'cbit': 2008, 'python': 1991, 'apssdc': 2014, 'Vbit': 2008}
```

In [30]:
```
d1['vbit']=2000
```

In [31]:
```
d1
```

Out[31]:
```
{'cbit': 2008, 'python': 1991, 'apssdc': 2014, 'Vbit': 2008, 'vbit': 2000}
```

In [33]:
```
d1.pop('vbit')#Removing paticular Element from Dict
```

Out[33]:
```
2000
```

In [34]:
```
d1
```

Out[34]:
```
{'cbit': 2008, 'python': 1991, 'apssdc': 2014, 'Vbit': 2008}
```

In [35]:
```
d1.popitem()
```

Out[35]:
```
('Vbit', 2008)
```

In [37]:
```
d1
```

Out[37]:
```
{'cbit': 2008, 'python': 1991, 'apssdc': 2014}
```

In [38]:
```
#Displaying all the Keys and Values wihch present in the dict


# cbit-->2008
# python-->1991
# apssdc-->2014
for k,v in d1.items():
    print(k,"-->",v)
```

```
cbit --> 2008
python --> 1991
apssdc --> 2014
```

In [ ]:
```
#Tasks : Using Functions
    #1.Create a contact dict.Then add the contact details
        #like name and number to dict
            #note:if  name in dict disply name already exist
    #2.Update the Contact which present in the dict
        #note:if not name in dict disply name not exist
    #3.Delete the contact name and number from dict
        #note:if not name in dict disply name not exist
    #4.List out the all contacts from dict
        #Note:if contacts is empty disply "empty list"

```

In [39]:
```
contacts={}
```

In [ ]:
```
 #1.Create a contact dict.Then add the contact details
        #like name and number to dict
            #note:if  name in dict disply name already exist
```

In [59]:
```
#Funtion to add a contact to dict
def addContact(name,number):
    if name not in contacts:
        contacts[name]=number
        print("Contact added Successfully.")
    else:
        print("Already name Exist:",name)

name=input("enter the Name")
number=int(input("Enter the number"))
addContact(name,number)
```

```
enter the Namehanuman
Enter the number1234
Contact added Successfully.
```

In [47]:
```
contacts
```

Out[47]:
```
{'muni': 111111, 'vijay': 4544}
```

In [50]:
```
#Funtion to update  a contact to dict
def updatecontact(name):
    if name in contacts:
        number=int(input("Enter the number"))
        contacts[name]=number
        print("Contact updated Successfully.")
    else:
        print("Not avalilabe :",name)

name=input("enter the Name")

updatecontact(name)
```

```
enter the Namemuni
Enter the number3333333
Contact updated Successfully.
```

In [55]:
```
#Funtion to update  a contact to dict
def deleteContact(name):
    if name in contacts:
        contacts.pop(name)
        print("deleted Successfully.")
    else:
        print("Not avalilabe :",name)

name=input("enter the Name")

deleteContact(name)
```

```
enter the Namevijay
deleted Successfully.
```

In [56]:
```
contacts
```

Out[56]:
```
{}
```

In [60]:
```
#Funtion to update  a contact to dict
def displayAllContacts(d):
    if len(contacts)<=0:
        print("Empty contacts list ")
    else:
        for k,v in contacts.items():
            print(k,"-->",v)
displayAllContacts(contacts)
```

```
hanuman --> 1234
```

In [ ]:
```
#Task:
   # 1.
#    input: {'cse':5,"ece":4,"eee":3,"civ":2,"mech":1}
#     output:csecsecsecsecse
#            eceeceeceece
#             eeeeeeeee
#             civciv
#             mech
 # 2.
    # Find the Freqvency char and count
    #input:data='weareincbitproddaturlearingpythonprogramming'
#             data1="hihellocbitccccccc"
#         o/p:c-->8
```

In [64]:
```
#Task:
   # 1.
#    input: {'cse':5,"ece":4,"eee":3,"civ":2,"mech":1}
#     output:csecsecsecsecse
#            eceeceeceece
#             eeeeeeeee
#             civciv
#             mech
b={'cse':5,"ece":4,"eee":3,"civ":2,"mech":1}
v=list(b.values())
v
```

Out[64]:
```
[5, 4, 3, 2, 1]
```

In [69]:
```
for k,v in b.items():
    print(k*v)
```

```
csecsecsecsecse
eceeceeceece
eeeeeeeee
civciv
mech
```

In [95]:
```
# data1="hahellllllllllllllllllllllllllooooooooooocbitccccccc"
data1='weareincbitproddaturlearingpythonprogramming'
un=[]
di={}
for i in data1:#For getting Unique keys
    if i not in un:
        un.append(i)
for j in un:   #Adding the no of repetations
        di[j]=data1.count(j)
m=max(di.values()) #max values from dict
for k,v in di.items():#getting for both max key and Value
    if m==v:
        print(k,"-->",v)
```

```
r --> 6
```

In [3]:
```
# 8 6 1
c=list(map(int,input().split()))
sorted(c)
```

```
8 6 1
```

Out[3]:
```
[1, 6, 8]
```

In [ ]:
```
# 5
# 85 25 65 21 84
    55514%10
    no
```

In [7]:
```
n=int(input())
c=list(map(int,input().split()))

```

```
75841
679 29980 47815 72976 87375 53880 15953 45632 18034 48300 93609 21568 51752 8
4044 37900 95328 75110 90176 49575 99600 96008 76128 58868 12864 26875 80788
65666 32928 80792 65400 78014 3232 75006 11348 2950 73728 8928 9576 42511 888
00 66879 8772 4940 91456 83800 53404 70883 16928 84865 45000 83283 79008 6968
0 96040 2425 44224 12846 78284 43267 91200 42298 22736 37444 67616 98400 4062
8 11521 39344 27747 58800 84043 72480 65880 97252 23750 87088 69982 63628 469
53 51200 1035 680 46088 68384 85600 88024 12111 66048 67136 24000 35743 8064
96845 59612 4125 10272 84212 52792 71657 40000 51026 44308 53837 8320 57750 4
2864 54975 58704 27129 33500 65096 70432 84587 83928 35950 21440 53536 880 75
998 12800 89755 63116 23038 56960 81250 80664 70547 63776 8578 9600 82342 237
44 23228 97100 25900 47488 78512 33064 32006 0 39921 45868 98694 67968 94250
61756 43428 50384 43583 72500 31888 65952 97462 36632 10600 97056 91855 77992
2284 64800 54739 43112 93856 75600 16425 19708 40082 4864 7135 6800 83605 755
68 86606 7196 18125 86944 48910 88632 71195 50400 9765 12092 84075 36544 4650
54656 15621 34160 72942 47300 55505 25664 70510 18072 59775 36416 21178 41096
69477 0 95017 17728 67074 54688 24225 22636 14371 30464 94529 9700 6811 50176
63354 39148 25 33888 76398 5732 6820 22800 99367 47856 87882 18048 86875 4951
6 62818 77440 57182 65800 96350 80832 25624 86944 67150 39088 40617 13116 668
```

In [5]:
```
99999**9999
```

Out[5]:
```
90484601407448825888563042346605458233021361015111900453967466926159666809167
```

In [1]:
```
10**9
```

Out[1]:
```
1000000000
```

In [20]:
```
n=int(input())
c=list(map(int,input().split()))
c
```

```
5
85 25 65 21 84
```

Out[20]:
```
[85, 25, 65, 21, 84]
```

In [27]:
```
n=int(input())
n1=list(map(int,input().split()))
re=[]
for i in range(len(n1)):
    remi=n1[i]%10
    re.append(remi)
print(re)
for j in range(len(re)):
    r=re[j]
if r%10==0:
    print("Yes")
else:
    print("No")
```

```
5
85 25 65 21 84
[5, 5, 5, 1, 4]
No
```

Regular Expressions

- It is one language .It will match the some seqvence of patters
- It will returns the 100% output is accurate
- Short from re
- we need to import the re(regular expressions)

In [29]:
```
import re

print(dir(re))
```

```
['A', 'ASCII', 'DEBUG', 'DOTALL', 'I', 'IGNORECASE', 'L', 'LOCALE', 'M', 'MULTI
LINE', 'Match', 'Pattern', 'RegexFlag', 'S', 'Scanner', 'T', 'TEMPLATE', 'U',
'UNICODE', 'VERBOSE', 'X', '_MAXCACHE', '__all__', '__builtins__', '__cached_
_', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__',
'__version__', '_cache', '_compile', '_compile_repl', '_expand', '_locale', '_p
ickle', '_special_chars_map', '_subx', 'compile', 'copyreg', 'enum', 'error',
'escape', 'findall', 'finditer', 'fullmatch', 'functools', 'match', 'purge', 's
earch', 'split', 'sre_compile', 'sre_parse', 'sub', 'subn', 'template']
```

match:Will return the only first char match

Search:If get one success it will terminate

FindAll:It will return all cases

In [32]:
```
re.match('a','apssdc')
```

Out[32]:
```
<re.Match object; span=(0, 1), match='a'>
```

In [33]:
```
re.match('a','pssdca')
```

In [34]:
```
re.search('s','apssdc')
```

Out[34]:
```
<re.Match object; span=(2, 3), match='s'>
```

In [35]:
```
re.search('s','sapssdc')
```

Out[35]:
```
<re.Match object; span=(0, 1), match='s'>
```

In [36]:
```
#Finall
re.findall('s','apssdc')
```

Out[36]:
```
['s', 's']
```

symbols and meanings

1. ^-->Exactly starting letter with
2. $-->Exactly ending with
3. []-->data set
4. {}-->range Set

5. .--->one char/digt/splChar ...etc
6. |--->either this or that
7. - --> with out null
8. *--->multiply

In [ ]:
```
#s=cbit1234&^JHGJHG
# Get the all digits
#Mobile number validation
    #case:1-->Starting with 6 or 7 or 8 or 9
        #^[6-9][0-9]{9}$
    #case:2-->Starting with 0(zero)
        #^[0][6-9][0-9]{9}$
    #Case 3:starting with +91
        #^[+][9][1][6-9][0-9]{9}$
#Final Pattern for Mobile number validation
#^[+][9][1][6-9][0-9]{9}$|^[0][6-9][0-9]{9}$|^[6-9][0-9]{9}$
```

In [46]:
```
li=['9876543210','652434247','1213713501','1327531122','0987654321']
pattern='^[+][9][1][6-9][0-9]{9}$|^[0][6-9][0-9]{9}$|^[6-9][0-9]{9}$'
for i in li:
    if re.match(pattern,str(i)):
        print(i)
```

```
9876543210
```

In [45]:
```
import re
n=input()
pattern='^[+][9][1][6-9][0-9]{9}$|^[0][6-9][0-9]{9}$|^[6-9][0-9]{9}$'
if re.match(pattern,n):
    print("yes valid number",n)
else:
    print("Invalid number goto home")
```

```
131
Invalid number goto home
```

In [ ]:
```
#Task for Email id validation
    #muni.apssdc@gmail.com
    #muneiah.t@apssdc.in
    #example123@gmail.com
    #student@ksrm.org
    #student@rguktrkv.ac.in
    #example@yahoo.com
    #example@hotmail.com
    #muneiah@outlook.com

```