

CHANGE ALL THE CODES TO AVOID PLAGIARISM :-

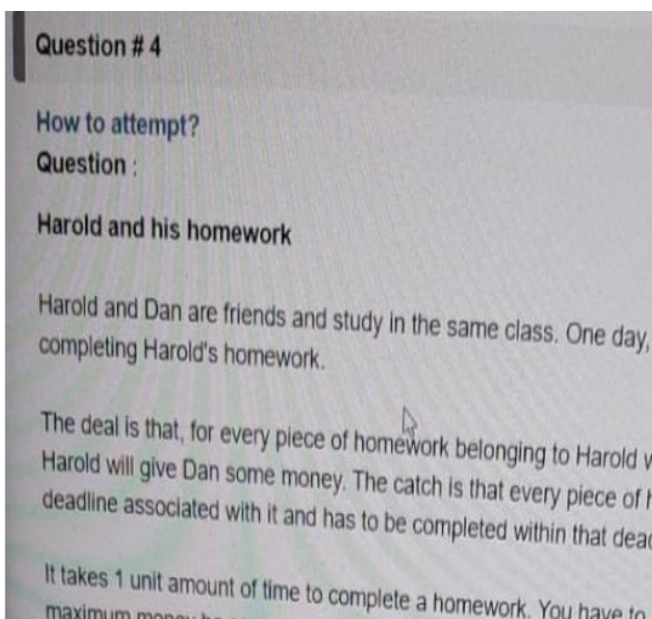
1. Harold and his homework:

```
language: PYTHON3 Compiler: Python 3.6

input1 : int
input2 : int[]
input3 : int[]

Expected return type : int
...

# Read only region end
# Write code here
#m=list(map(int,.split()))
m=input1
#d=list(map(int,input2.split()))
d=input2
dic={}
t=input1
for i in range(0,t):
    if d[i] not in dic.keys():
        dic[d[i]]=m[i]
    else:
        dic[d[i]]=max(m[i],dic[d[i]])
return sum(dic.values())
```



NOTE: Here, m=input2
d=input3

2. Halindrome:

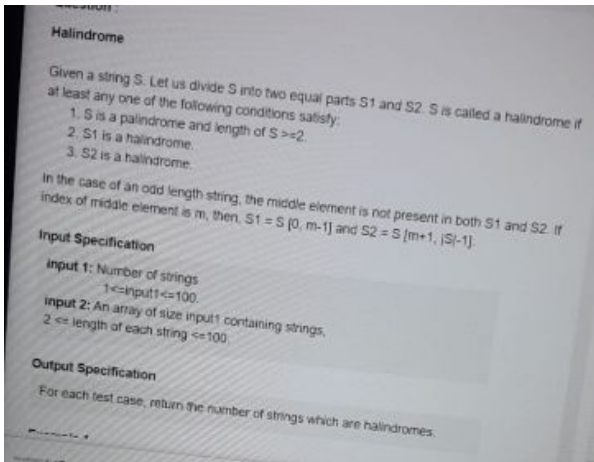
(change it as per the question)

CHANGE ALL THE CODES TO AVOID PLAGIARISM :-

```
def ispal(s):
    rev = ''.join(reversed(s))
    if s == rev:
        return True
    return False

def ishal(s):
    if len(s) < 2:
        return False
    if len(s) == 2 and not ispal(s):
        return False
    if len(s) >= 2 and ispal(s):
        return True
    else:
        if len(s) >= 4:
            s1 = s[0:len(s)//2]
            s2 = s[(len(s)//2)+1:] if len(s)%2 != 0 else s[len(s)//2:]
            if ishal(s1) or ishal(s2):
                return True

n = int(input())
a = list(input().split(", "))
c = 0
for i in range(n):
    if ishal(a[i]):
        c += 1
print(c)
```



Abraham :

```
n=int(input())
c=0;
while n/2 >= 1:
    c+=1
    n=n/2
if(n==2):
    a=(2*(2**c))-1
else:
    a=(2**c)-1
print(a)
```

//Abraham in python

9:30 am

Planting Trees:

CHANGE ALL THE CODES TO AVOID PLAGIARISM :-

```
n=int(input())
k=int(input())

trees=[2]

for i in range(n):
    trees_planted=list(range(0,(trees.pop(0)+1)%k))
    trees+=trees_planted
#print(trees)
if len(trees)==0:
    print(1)
else:
    print(len(trees))
```

10:14 am

Frequency co:

CHANGE ALL THE CODES TO AVOID PLAGIARISM :-

```
Language: PYTHON3 Compiler : Python 3.6
# Read only region end
# Write code here
dic={}
for i in input1:
    dic[i]=dic.get(i,0)+1
l=[]
for i in dic.items():
    l.append(i)
l.sort()
#print(l)
newl=[]
for i in l:
    newl.append(i[0])
    newl.append(str(i[1]))
news=''.join(newl)
return news
```

Monica and flavours some errors:

```
expected return type : int
...
# Read only region end
l=[]
input3=list(input3)
for i in range(0,input2):
    if input3[i] not in l:
        #print(input3[i])
        l.append(input3[i])
#print(l)
return len(l)
```

Social Network

CHANGE ALL THE CODES TO AVOID PLAGIARISM :-

```
1
2 #SOCIAL NETWORK
3 |
4 print("Enter Range:",end=" ")
5 n=int(input())
6 a=[]
7
8 a=list(range(n+1))
9 for i in range(2,int(n**(1/2))+1):
10     s=0
11     if(i!=-1):
12         k=0
13         f=1
14         while(f):
15             s=i*(i+k)
16             if(s<n+1):
17                 a[s]=-1
18                 k=k+1
19             else:
20                 f=0
21 c=0
22 for i in a[2:]:
23     if i!=-1:
24         c=c+1
25
```

Shell ×

```
Python 3.7.6 (bundled)
>>> %Run 'Social Network.py'

Enter Range: 10
Total: 3
```

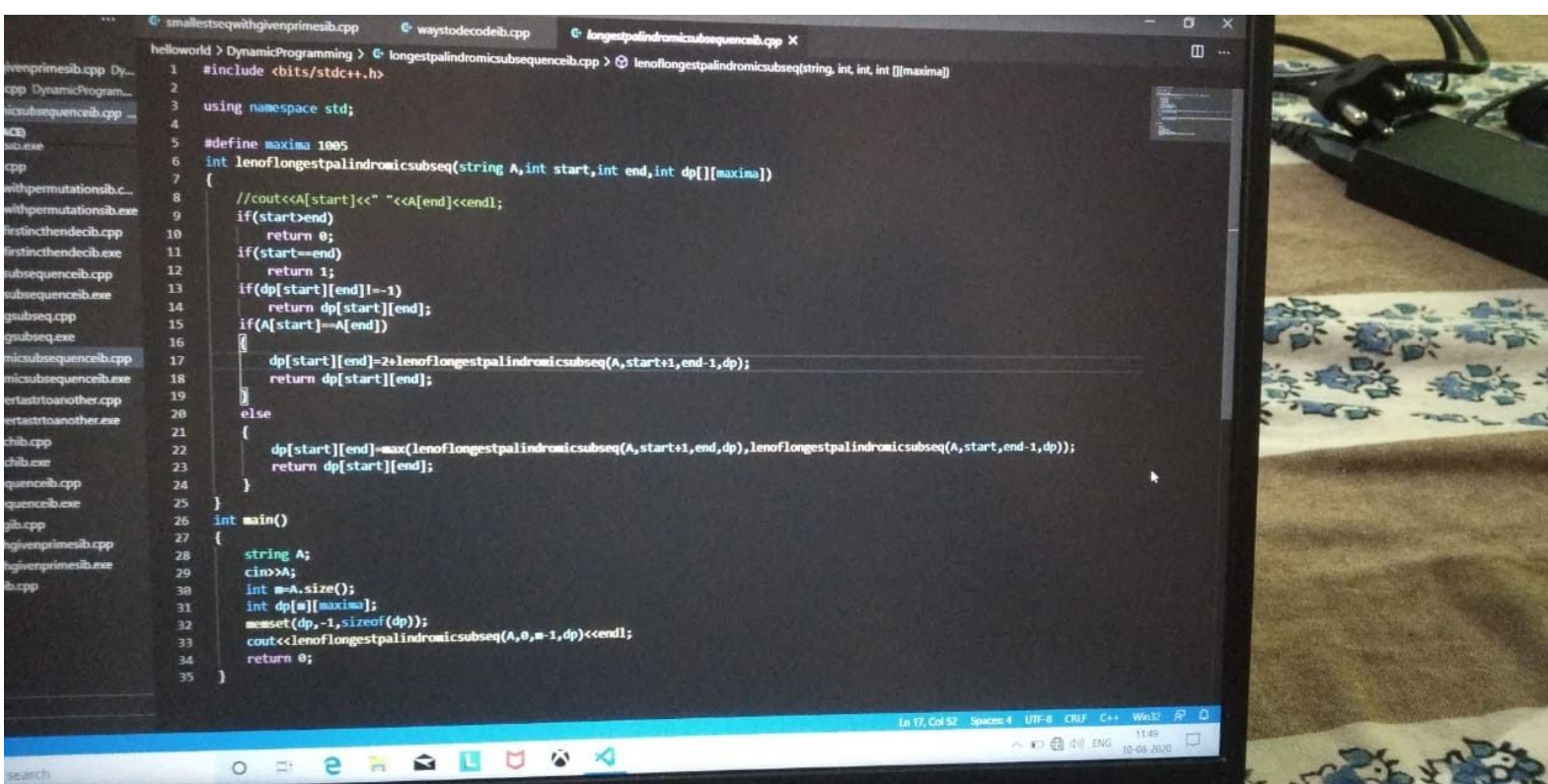
LCS with Vowels

```
str1 = input()
str2 = input ()
```

CHANGE ALL THE CODES TO AVOID PLAGIARISM :-

```
vowels = 'aeiou'  
c1 = c2 = 0  
for ch in str1:  
    if ch in vowels:  
        c1 += 1  
for ch in str2:  
    if ch in vowels:  
        c2 += 1  
print(min(c1,c2))
```

Longest palindromic subsequence



```
1 #include <bits/stdc++.h>  
2  
3 using namespace std;  
4  
5 #define maxima 1005  
6 int lenoflongestpalindromicsubseq(string A,int start,int end,int dp[][maxima])  
7 {  
8     //cout<<A[start]<<" "<<A[end]<<endl;  
9     if(start>end)  
10        return 0;  
11     if(start==end)  
12        return 1;  
13     if(dp[start][end]!=-1)  
14        return dp[start][end];  
15     if(A[start]==A[end])  
16     {  
17         dp[start][end]=2+lenoflongestpalindromicsubseq(A,start+1,end-1,dp);  
18         return dp[start][end];  
19     }  
20     else  
21     {  
22         dp[start][end]=max(lenoflongestpalindromicsubseq(A,start+1,end,dp),lenoflongestpalindromicsubseq(A,start,end-1,dp));  
23         return dp[start][end];  
24     }  
25 }  
26  
27 int main()  
28 {  
29     string A;  
30     cin>>A;  
31     int n=A.size();  
32     int dp[n][maxima];  
33     memset(dp,-1,sizeof(dp));  
34     cout<<lenoflongestpalindromicsubseq(A,0,n-1,dp)<<endl;  
35     return 0;  
}
```

Bob and Numbers

CHANGE ALL THE CODES TO AVOID PLAGIARISM :-

```
n = int(input())
a = list(map(int,input().split()))
c = 0
for i in range(n-1):
    for j in range(i+1:n):
        s1 = sum(list(map(int,bin(i)[2:].split()))))
        s2 = sum(list(map(int,bin(j)[2:].split()))))
        if s1 == s2:
            c += 1
print(c)
```

apple orchard solution

Sort the array .
Initialize a max = 0
iterate a loop on sorted array.
suppose sorted arr[] = {48,80,82}
now iterate
for (int i = 0 ; i < n ; i++)
{
int temp = arr[i] * (n-i+1);
if (temp > max)
{
max = temp;
}
} // end of for loop
Print(max)

New----

```
n = int(input())
l = list(map(int,input().split()))
res = min(l)*n
print(res)
```

CHANGE ALL THE CODES TO AVOID PLAGIARISM :-

Evaluate a given infix expression

Use eval() in python

```
Python >>>
>>> eval("2 ** 8")
256
>>> eval("1024 + 1024")
2048
>>> eval("sum([8, 16, 32])")
56
>>> x = 100
>>> eval("x * 2")
200
```

Maximum subarray

def maxSubArray(self, nums: List[int]) -> int:

```
    max_sub_sum = 0
    prev_max = 0
    for num in nums:
        prev_max = max(prev_max + num, num)
        max_sub_sum = max(prev_max, max_sub_sum)
```

#if max is 0 and 0 not in list it means that there is no num in subarray, so we choose max num from list

if max_sub_sum is 0 and 0 not in nums:

```
    max_sub_sum = max(nums)
```

return max_sub_sum

CHANGE ALL THE CODES TO AVOID PLAGIARISM :-

MOVING APPLES

```
Language: PYTHON3 ▾ Compiler: Python 3.6
6      ...
7      input1 : int
8      input2 : int[]
9
10     Expected return type : int
11     ...
12     # Read only region end
13     # Write code here
14     avg=sum(input2)//input1
15     s=0
16     for i in input2:
17         if i>avg:
18             s=s+(i-avg)
19     return s
20     pass
21
```

CHANGE ALL THE CODES TO AVOID PLAGIARISM :-

NEXT GENERATOR NUMBER

```
class Solution {
public int[] nextGreaterElements(int[] nums) {
    int[] output=new int[nums.length];
    int n=nums.length;
    Arrays.fill(output,-1);
    Stack<Integer> stk=new Stack();

    for(int i=0;i<n*2;i++){

        while(!stk.isEmpty() && nums[stk.peek()]<nums[i%n]){

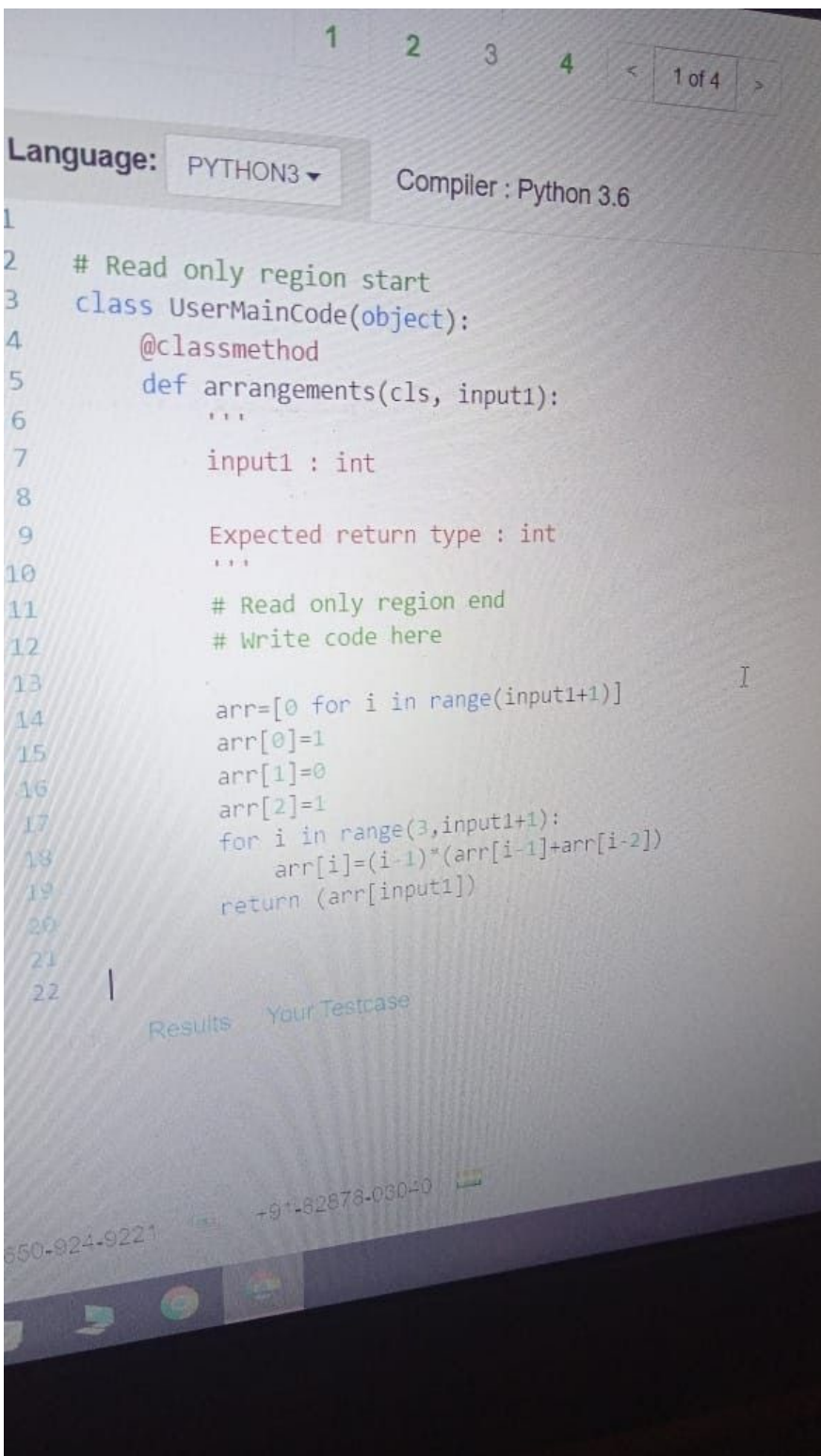
            output[stk.pop()]=nums[i%n];
        }

        if(i<n) stk.push(i);

    }
    return output;
}
}
```

Selective arrangements

CHANGE ALL THE CODES TO AVOID PLAGIARISM :-



The image shows a screenshot of a Python IDE interface. At the top, there are navigation tabs numbered 1, 2, 3, and 4, with a '1 of 4' indicator. Below the tabs, the 'Language' is set to 'PYTHON3' and the 'Compiler' is 'Python 3.6'. The main area contains Python code for a class named 'UserMainCode' with a method 'arrangements'. The code includes comments for 'Read only region start' and 'Read only region end', and a section for 'Write code here'. The implementation uses a list 'arr' to store Fibonacci numbers, with initial values [0, 1, 0, 1] and a loop that calculates the next number as the sum of the two previous numbers. The method returns the value at the index 'input1'.

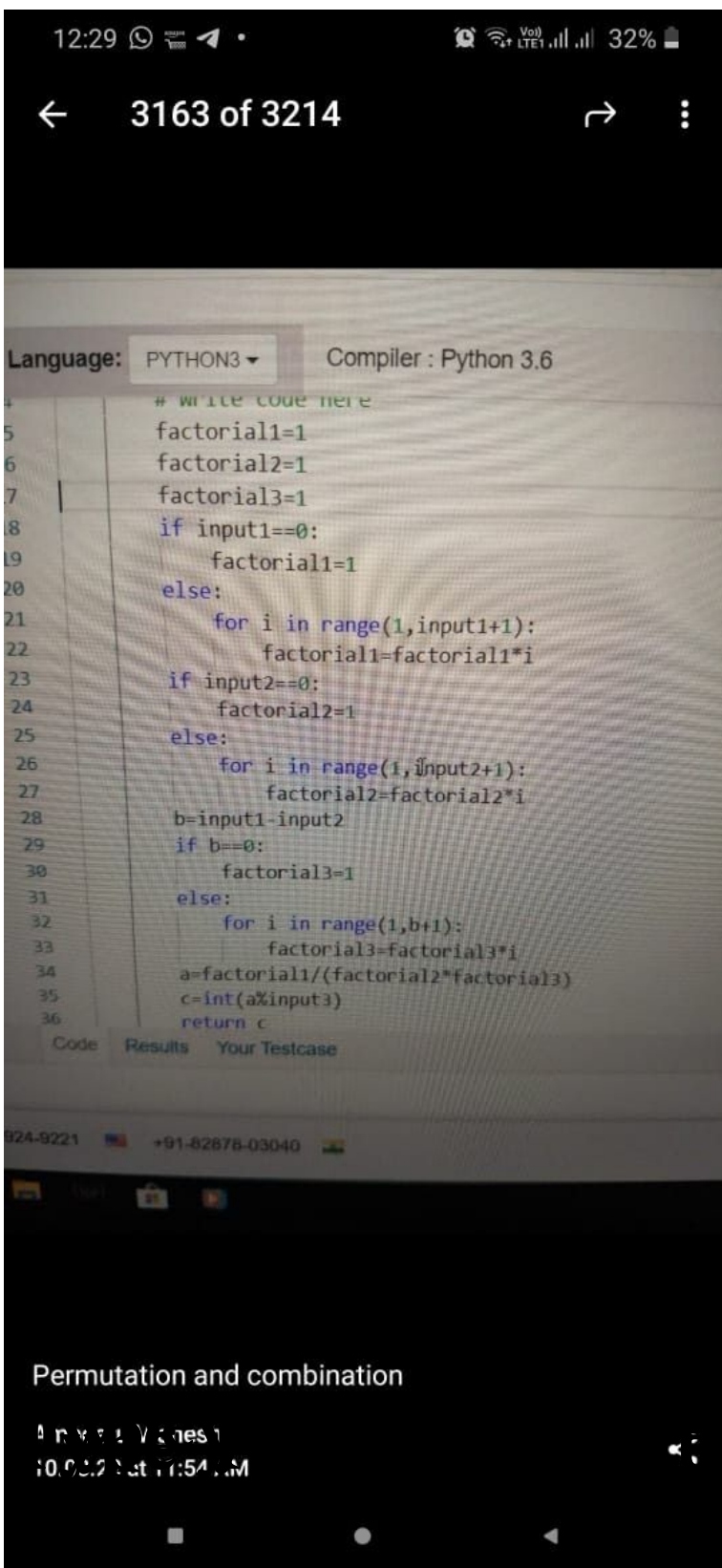
```
1
2 # Read only region start
3 class UserMainCode(object):
4     @classmethod
5     def arrangements(cls, input1):
6         ...
7         input1 : int
8
9         Expected return type : int
10        ...
11        # Read only region end
12        # Write code here
13
14        arr=[0 for i in range(input1+1)]
15        arr[0]=1
16        arr[1]=0
17        arr[2]=1
18        for i in range(3,input1+1):
19            arr[i]=(i-1)*(arr[i-1]+arr[i-2])
20        return (arr[input1])
21
22
```

Results Your Testcase

650-924-9221 +91-82878-03040

CHANGE ALL THE CODES TO AVOID PLAGIARISM :-

Permutation combination



CHANGE ALL THE CODES TO AVOID PLAGIARISM :-

Abraham:

```
n=int(input())
a= {j:i for i in range(n)}
#print(a)
while(len(a)>1):
    l=[]
    for i in range(0,len(a),2):
        l.append(i)
    #print('l=',l)
    for i in l:
        a.pop(i)
        #print(i)
    #print(a)
    counter=0
    d1={}
    for i in a:
        key=i
        val=a[i]
        #a.pop(key)
        d1[counter]=val
        counter+=1
    #print(a)
a=dict(d1)
#print(a)
#input()
print(a[0])
```

CHANGE ALL THE CODES TO AVOID PLAGIARISM :-

Think a number:

