**ĐÁNH GIÁ BÀI TẬP VỀ NHÀ CỦA CÁC NHÓM**

**đề bài : hãy tưởng tượng bạn là một nhà sinh học đang thực hiện các nghiên cứu về ARN . trong đó có các nucleotic ATGX  
.Trong công việc nghiên cứu của mình , bạn phải tìm và xác định những biến dị về gen của các ARN này , vậy nên bạn phải có danh sách của các biến dị có thể xảy ra . Hãy xây dựng 1 chương trình có thể tạo ra tất cả các biến dị có thể xảy ra khi các nucleotic này hoán vị với nhau .  
ví dụ :  
cho ATG là input .  
outout : ATG  
AGT  
TAG  
TGA  
GTA  
GAT                 lưu ý : GTA != ATG       và       không được in ra kết quả trùng nhau**

**nhóm 10 : nhóm nộp bài đầu tiên từ rất sớm**

code chạy hoàn thiện và thực hiện đúng yêu cầu của đề bài ,

(Phần link đính kèm trên mail bi lỗi nhẹ ko đáng kể )

**Nhóm 12 : nhóm về thứ 2**

Code chạy đúng yêu cầu và in ra kết quả đầy đủ , ( góp ý nhỏ : nhóm có thể bổ sung phần nhận diện trùng lặp)

**Nhóm 11 : hoàn thành tốt**

Code cũng hoàn thành tốt yêu cầu của đề bài và in ra đc kết quả phù hợp ,( có thể bổ sung thêm tính năng như nhóm 12 )

**Nhóm 13 : có đầy đủ tất cả các yêu cầu và có loại các trường hợp trùng nhau**

Hoàn thành tốt

**Nhóm 1 : bạn Bá Thi : nhóm bạn hoàn thành đầy đủ các yêu cầu của đề ra**

**Nhóm 6 : nhóm bạn hoàn thành đầy đủ các yêu cầu của đề ra**

**Nhóm 2 : nhón bạn Trí làm rất công phu ( 2 cách ) và in ra kết quả đúng**

**Lấy bài nhóm 2 làm minh họa :**

def per1(nucleotic\_list):

stack = list(nucleotic\_list)

results = [stack.pop()]

while len(stack) != 0:

c = stack.pop()

results\_2 = []

for w in results:

for i in range(len(w) + 1):

results\_2.append(w[:i] + c + w[i:])

results = results\_2

return sorted(results)

def per2(nucleotic\_list, l, r):

if l == r:

print(''.join(nucleotic\_list))

else:

for i in range(l, r + 1):

nucleotic\_list[l], nucleotic\_list[i] = nucleotic\_list[i], nucleotic\_list[l]

per2(nucleotic\_list, l + 1, r)

nucleotic\_list[l], nucleotic\_list[i] = nucleotic\_list[i], nucleotic\_list[l]

def per3(i, s):

if i == len(s) - 1:

print(s)

return s

prev = '\*'

for j in range(i, len(s)):

temp = s.copy();

if (j > i and temp[i] == temp[j]):

continue

if prev != '\*' and prev == s[j]:

continue

t = temp[i]

temp[i] = temp[j]

temp[j] = t

prev = s[j]

per3(i + 1, temp)

nucl\_list1 = 'ATGX'

nucl\_list2 = 'ATGT'

# unduplicated

print(per1(nucl\_list1))

per2(list(nucl\_list1), 0, len(nucl\_list1) - 1)

# duplicated

per3(0, sorted(nucl\_list2))