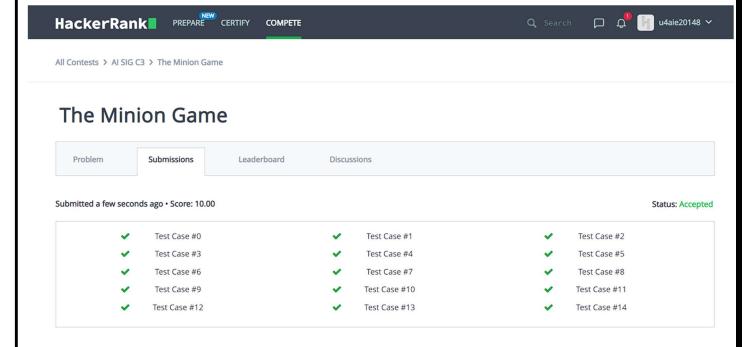
### AI-ACM SIG TASKS

NAME: KARTHIKEYA MOTURI

ROLL NUMBER: AM.EN. U4AIE20148

BATCH: AIE & B-BATCH

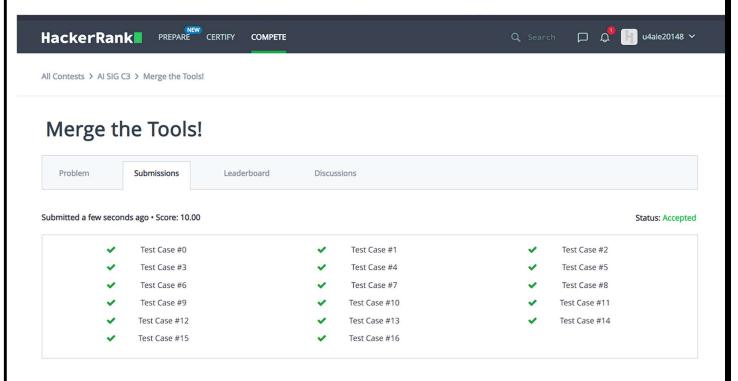
## The Minion Game



#### **Submitted Code**

```
Language: PyPy3
                                                                                                                     P Open in editor
 1 def minion_game(s):
      p1 = 0
       p2 = 0
      vowels = ['A','E','I','O','U']
      for i in range(len(s)):
          if s[i] in vowels:
              p1 = p1 + (len(s) - i)
8
9
           else:
              p2 = p2 + (len(s) - i)
10
11
12
      if (p1 > p2):
13
           print('Kevin', p1)
14
15
16
      elif (p1 < p2):
           print('Stuart', p2)
18
19
      elif(p1==p2):
21
22
           print('Draw')
```

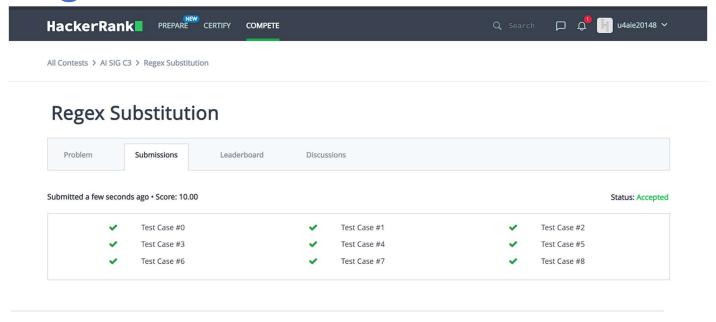
# Merge the Tools!



#### **Submitted Code**

```
Language: PyPy3
                                                                                                                    & Open in editor
1 def merge_the_tools(string, k):
      n = len(string)
      lst1 = []
      for i in range(0,n,k):
5
           s = ""
6
7
           for j in range(k):
              s += string[i+j]
9
          lst1.append(s)
10
11
      for wd in lst1:
12
13
           for j in range(k):
14
               if wd[j] not in m:
15
                  m += wd[j]
          print(m)
16
```

# Regex Substitution



#### **Submitted Code**

```
Language: PyPy3

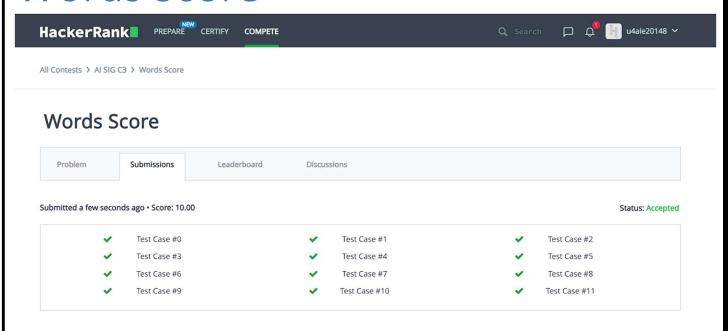
import re

def function(match):
    if match.group(1) == '&&':
        return 'and'
    else:
        return 'or'

x = int(input())
for _ in range(x):
    print(re.sub(r"(?<= )(\|\||&&)(?= )", function,input()))</pre>
```

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |

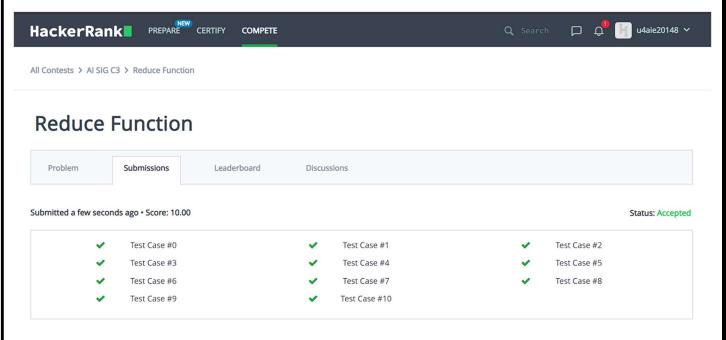
## Words Score



#### **Submitted Code**

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |

## Reduce Function



#### **Submitted Code**

```
Language: PyPy3

3 def product(fracs):
    t = Fraction(reduce(lambda x, y: x * y, fracs))
    return t.numerator, t.denominator
```