Definizione di una funzione che data una location, l'API di twitter, una collection delle TT e i tempi da una request all'altra all'API di Twitter e di reset delle rate limit, inserisce le trending topics in tale collezione

```
def getAndAddTrendingTopics(location,
                            api_di_twitter,
                            collezione_delle_trending_topics,
                            seconds_between_each_app_request_arg,
                            seconds_to_reset_arg):
    rate_status_trends = api_di_twitter.application.rate_limit_status()['resources']['trends']['/trends/place']
    time.sleep(seconds_between_each_app_request_arg)  # wait for trends request time
    # print("seconds_between_each_app_request   ", seconds_between_each_app_request_arg)
    limit_trends = rate_status_trends['limit']
    remaining_limit_trends = rate_status_trends['remaining']
    seconds_between_each_trends_request = seconds_to_reset_arg / limit_trends
    # print("remaining_limit_trends             ", remaining_limit_trends)
    # print("seconds_between_each_trends_request", seconds_between_each_trends_request)

    while remaining_limit_trends <= 1:
        remaining_limit_trends = \
            api_di_twitter.application.rate_limit_status()['resources']['trends']['/trends/place']['remaining']
        time_to_wait = 0 if remaining_limit_trends > 1 else max(seconds_between_each_trends_request,
                                                                seconds_between_each_app_request_arg)
        time.sleep(time_to_wait)  # wait for trends or app request time
        # print("seconds_between_each_app_trend_req ", time_to_wait)
        # print("remaining_limit_trends             ", remaining_limit_trends)

    trendsDataJsonText = api_di_twitter.trends.place(_id=location['woeid'])
    time.sleep(max(seconds_between_each_trends_request - seconds_between_each_app_request_arg,
                   0))  # wait for trends request time - app request time
    trends = trendsDataJsonText[0]['trends']
    dateTimeNow = datetime.today().strftime('%Y-%m-%dT%H:%M:%SZ')
    metadata = {
        "as_of": trendsDataJsonText[0]['as_of'],
        "created_at": str(dateTimeNow),
        "locations_name": location['name'],
        "location_type": location['placeType']['name'],
        "locations_woeid": location['woeid'],
        "parent_id": location['parentid'],
        "parent_name": "Worldwide" if location['placeType']['name'] == "Country" else location[
            'country']
    }
    for trend in trends:
        record = {**metadata, **trend}
        if not checkIfPresent(collezione_delle_trending_topics, record):  # controlla che tale trending topic non sia già in db
            collezione_delle_trending_topics.insert_one(record)
```