Program Explanation (function by function):

Generate Cards:
This function, when given an integer, generates two lists, x, y and in range of the integer, adds to the list a random number between 1 and the integer. It then returns x and y

Check Int:
This function will take n and try to convert it into an integer and make sure it is greater than 0 (1+), then will return True if so. Will return False otherwise

Main:
This function will get the user to input a number, then while check int is called and equalled to false an input is asked again and again making sure that a number is indeed given. Defence and attack cards are then generated using generateCards() followed by greedy() being called with the generated lists and being equalled to the lists. A counter int is defined and in range length of the list, the list is printed with the match displayed each iteration and the worth of the card is added to the counter int if it won. The final total worth of the winning cards is displayed

Greedy:
Return lists are defined at the very start, now bear with me because this is confusing for me too. For range of the list of defence cards given as an input a current highest value is created and so is a match value, both lists, one nested. For J in range of the defence card list if the current card is higher than the current_highest variable set earlier it will be set to that. This finds the currently highest item in the list. For k in range will then find a match for that item via first looking for an item equal too then greater than, once found will add to match. At the end of I's iteration if a match has not been found this means all cards are dead and all the losing cards are added to the end of the return lists, otherwise else, the match is append to the return lists. Finally, at the end of I's iterations the return lists are return

Optimality:
This is not the most optimal solution as there are many iterations for than likely needed, furthermore, for k in range is not as good as it could be. Occasionally the perfect match will not be found. A match needed to be selected in the same way to current_highest otherwise occasionally a better match will be overwritten. This is a rather simple change that should've been made with hindsight.

There also could be a few better strategies that a more efficient or effective to complete the given task that I have not found.

My solution is not the most Optimal