Inversion count:

Positive test cases:

1:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

/opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/invcount-positive-tc1.py"
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/invcount-positive-tc1.py"
Total inversion count (Brute Force) across all students: 244
Total inversion count (Divide and Conquer) across all students: 244

Categorized Inversion Counts (Brute Force):
Inversion Count 0: Students [12, 18, 27, 29, 38, 41, 45, 49, 55, 61, 90, 94, 97, 98, 99]
Inversion Count 1: Students [2, 5, 13, 21, 22, 33, 37, 46, 58, 78, 85, 87, 88, 95]
Inversion Count 2: Students [1, 4, 15, 19, 23, 25, 35, 40, 50, 54, 56, 60, 66, 69, 73, 77, 79, 81, 83, 84, 86, 92, 100]
Inversion Count 3: Students [6, 8, 11, 14, 34, 36, 39, 43, 44, 48, 57, 62, 63, 64, 65, 70, 71, 72, 76, 80, 93]
Inversion Count 4: Students [7, 9, 10, 17, 24, 30, 32, 47, 51, 59, 67, 74, 75, 89, 91, 96]
Inversion Count 5: Students [16, 20, 26, 28, 31, 52, 53, 68, 82]
Inversion Count 6: Students [3, 42]

Categorized Inversion Counts (Divide and Conquer):
Inversion Count 0: Students [12, 18, 27, 29, 38, 41, 45, 49, 55, 61, 90, 94, 97, 98, 99]
Inversion Count 1: Students [2, 5, 13, 21, 22, 33, 37, 46, 58, 78, 85, 87, 88, 95]
Inversion Count 2: Students [1, 4, 15, 19, 23, 25, 35, 40, 50, 54, 56, 60, 66, 69, 73, 77, 79, 81, 83, 84, 86, 92, 100]
Inversion Count 3: Students [6, 8, 11, 14, 34, 36, 39, 43, 44, 48, 57, 62, 63, 64, 65, 70, 71, 72, 76, 80, 93]
Inversion Count 4: Students [7, 9, 10, 17, 24, 30, 32, 47, 51, 59, 67, 74, 75, 89, 91, 96]
Inversion Count 5: Students [16, 20, 26, 28, 31, 52, 53, 68, 82]
Inversion Count 6: Students [3, 42]
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

2:

```
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/invcount-positive-tc2.py"
Total inversion count (Brute Force) across all students: 251
Total inversion count (Divide and Conquer) across all students: 251

Categorized Inversion Counts (Brute Force):
Inversion Count 0: Students [12, 18, 27, 38, 41, 45, 49, 55, 61, 90, 94, 97, 98, 99]
Inversion Count 1: Students [2, 5, 13, 21, 22, 33, 37, 46, 58, 65, 78, 85, 87, 88, 95]
Inversion Count 2: Students [1, 4, 15, 19, 23, 25, 35, 40, 50, 54, 56, 60, 66, 69, 73, 79, 81, 82, 83, 84, 86, 92, 100]
Inversion Count 3: Students [6, 8, 11, 14, 34, 36, 39, 44, 48, 57, 62, 63, 64, 71, 72, 76, 80, 93]
Inversion Count 4: Students [7, 9, 10, 17, 24, 30, 32, 43, 47, 51, 59, 67, 70, 74, 75, 89, 91, 96]
Inversion Count 5: Students [16, 20, 26, 28, 31, 52, 53, 68]
Inversion Count 6: Students [3, 29, 42, 77]

Categorized Inversion Counts (Divide and Conquer):
Inversion Count 0: Students [12, 18, 27, 38, 41, 45, 49, 55, 61, 90, 94, 97, 98, 99]
Inversion Count 1: Students [2, 5, 13, 21, 22, 33, 37, 46, 58, 65, 78, 85, 87, 88, 95]
Inversion Count 2: Students [1, 4, 15, 19, 23, 25, 35, 40, 50, 54, 56, 60, 66, 69, 73, 79, 81, 82, 83, 84, 86, 92, 100]
Inversion Count 3: Students [6, 8, 11, 14, 34, 36, 39, 44, 48, 57, 62, 63, 64, 71, 72, 76, 80, 93]
Inversion Count 4: Students [7, 9, 10, 17, 24, 30, 32, 43, 47, 51, 59, 67, 70, 74, 75, 89, 91, 96]
Inversion Count 5: Students [16, 20, 26, 28, 31, 52, 53, 68]
Inversion Count 6: Students [3, 29, 42, 77]
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

3:

```
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/invcount-positive-tc3.py"
Total inversion count (Brute Force) across all students: 503
Total inversion count (Divide and Conquer) across all students: 503

Categorized Inversion Counts (Brute Force):
Inversion Count 1: Students [30, 67, 74, 80]
Inversion Count 2: Students [7, 18, 57]
Inversion Count 3: Students [15, 34, 38, 47, 50, 51, 58, 69, 70, 71, 72, 73, 79, 97]
Inversion Count 4: Students [2, 4, 11, 13, 17, 19, 20, 21, 35, 36, 40, 44, 46, 53, 61, 62, 65, 68, 77, 82, 84, 94, 99]
Inversion Count 5: Students [1, 5, 12, 23, 31, 52, 75, 85, 86, 87, 88, 92, 100]
Inversion Count 6: Students [6, 8, 9, 24, 27, 29, 33, 37, 39, 43, 49, 54, 56, 59, 63, 78, 81, 83, 96]
Inversion Count 7: Students [3, 10, 14, 16, 25, 26, 28, 41, 42, 45, 48, 66, 89, 93, 98]
Inversion Count 8: Students [22, 60, 64, 76, 90, 91]
Inversion Count 9: Students [32, 55, 95]

Categorized Inversion Counts (Divide and Conquer):
Inversion Count 1: Students [30, 67, 74, 80]
Inversion Count 2: Students [7, 18, 57]
Inversion Count 3: Students [15, 34, 38, 47, 50, 51, 58, 69, 70, 71, 72, 73, 79, 97]
Inversion Count 4: Students [2, 4, 11, 13, 17, 19, 20, 21, 35, 36, 40, 44, 46, 53, 61, 62, 65, 68, 77, 82, 84, 94, 99]
Inversion Count 5: Students [1, 5, 12, 23, 31, 52, 75, 85, 86, 87, 88, 92, 100]
Inversion Count 6: Students [6, 8, 9, 24, 27, 29, 33, 37, 39, 43, 49, 54, 56, 59, 63, 78, 81, 83, 96]
Inversion Count 7: Students [3, 10, 14, 16, 25, 26, 28, 41, 42, 45, 48, 66, 89, 93, 98]
Inversion Count 8: Students [22, 60, 64, 76, 90, 91]
Inversion Count 9: Students [32, 55, 95]
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

4:

```
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/invcount-positive-tc4.py"
Total inversion count (Brute Force) across all students: 651
Total inversion count (Divide and Conquer) across all students: 651

Categorized Inversion Counts (Brute Force):
Inversion Count 2: Students [66, 67]
Inversion Count 4: Students [11, 34, 54, 65, 74, 80]
Inversion Count 5: Students [2, 19, 20, 76]
Inversion Count 6: Students [7, 15, 17, 41, 45, 62, 64, 69, 70, 73]
Inversion Count 7: Students [1, 18, 40, 44, 49, 53, 56, 59, 72]
Inversion Count 8: Students [21, 22, 38, 50, 58, 61, 68, 71, 77, 79]
Inversion Count 9: Students [4, 5, 9, 13, 29, 39, 47, 48, 60, 75]
Inversion Count 10: Students [3, 12, 23, 27, 30, 31, 32, 35, 36, 37, 52, 63]
Inversion Count 11: Students [6, 8, 10, 14, 24, 25, 26, 33, 42, 46, 51, 55, 57, 78]
Inversion Count 12: Students [16, 28, 43]

Categorized Inversion Counts (Divide and Conquer):
Inversion Count 2: Students [66, 67]
Inversion Count 4: Students [11, 34, 54, 65, 74, 80]
Inversion Count 5: Students [2, 19, 20, 76]
Inversion Count 6: Students [7, 15, 17, 41, 45, 62, 64, 69, 70, 73]
Inversion Count 7: Students [1, 18, 40, 44, 49, 53, 56, 59, 72]
Inversion Count 8: Students [21, 22, 38, 50, 58, 61, 68, 71, 77, 79]
Inversion Count 9: Students [4, 5, 9, 13, 29, 39, 47, 48, 60, 75]
Inversion Count 10: Students [3, 12, 23, 27, 30, 31, 32, 35, 36, 37, 52, 63]
Inversion Count 11: Students [6, 8, 10, 14, 24, 25, 26, 33, 42, 46, 51, 55, 57, 78]
Inversion Count 12: Students [16, 28, 43]
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

Negative test cases:

1:

```
students_random_numbers = []
```

```
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/invcount-negative-tc1.py"
ERROR: The list of course code is empty, so the inversion count cannot be found by neither brute force nor divide and conquer approach.
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

2:

```
students_random_numbers = [
    [5, 2, 3, 6], ['a', 1, 5, 2], [7, 6, 4, 1], [6, 2, 'b', 7], [2, 3, 8, 4], [5, 5, 5, 4]
]
```

```
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/invcount-negative-tc2.py"

Categorized Inversion Counts (Valid Entries):
Student 1: Brute Force Inversion Count = 2, Divide and Conquer Inversion Count = 2
Student 3: Brute Force Inversion Count = 6, Divide and Conquer Inversion Count = 6
Student 5: Brute Force Inversion Count = 1, Divide and Conquer Inversion Count = 1
Student 6: Brute Force Inversion Count = 3, Divide and Conquer Inversion Count = 3

Error Messages for Invalid Entries:
Student 2: Error: Array contains non-integer values, inversion count can't be performed.
Student 4: Error: Array contains non-integer values, inversion count can't be performed.
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

3:

```
students_random_numbers = [
    [5, 2, 3, 6], ['a', 1, 5, 2], [-7, -6, -4, -1], [-6, -2, -5, -7], [2, 3, 8, 4], [5, 5, 5, 4]
]
```

```
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/invcount-negative-tc3.py"

Categorized Inversion Counts (Valid Entries):
Student 1: Brute Force Inversion Count = 2, Divide and Conquer Inversion Count = 2
Student 5: Brute Force Inversion Count = 1, Divide and Conquer Inversion Count = 1
Student 6: Brute Force Inversion Count = 3, Divide and Conquer Inversion Count = 3

Negative Integer Entries:
Student 3: Inversion count can be found since course code cant be negative.
Student 4: Inversion count can be found since course code cant be negative.

Error Messages for Invalid Entries:
Student 2: Error: Array contains letters instead of integer values, inversion count can't be performed.
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

4:

```
students_random_numbers = [
    [5, 2, 3, 6], [-3, -1, -5, -2], [-7, -6, -4, -1], [-6, -2, -5, -7], [2, 3, 8, 4], [5, 5, 5, 4
]
```

```
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/invcount-negative-tc4.py"

Categorized Inversion Counts (Valid Entries):
Student 1: Brute Force Inversion Count = 2, Divide and Conquer Inversion Count = 2
Student 5: Brute Force Inversion Count = 1, Divide and Conquer Inversion Count = 1
Student 6: Brute Force Inversion Count = 3, Divide and Conquer Inversion Count = 3

Negative Integer Entries:
Student 2: Inversion count can be found since course code cant be negative.
Student 3: Inversion count can be found since course code cant be negative.
Student 4: Inversion count can be found since course code cant be negative.
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

5:

```
students_random_numbers = [
    [5, 2, 3, 6], [-2, -1, -5, -2], [7, False, 4, 1], [6, 2, -5, -7],
    [2, 3, 8, 4], [5, 5, 5, 4], [True, False, 1, 0]
]
```

```
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/invcount-negative-tc5.py"

Categorized Inversion Counts (Valid Entries):
Student 1: Brute Force Inversion Count = 2, Divide and Conquer Inversion Count = 2
Student 4: Brute Force Inversion Count = 6, Divide and Conquer Inversion Count = 6
Student 5: Brute Force Inversion Count = 1, Divide and Conquer Inversion Count = 1
Student 6: Brute Force Inversion Count = 3, Divide and Conquer Inversion Count = 3

Negative Integer Entries:
Student 2: ERROR: Inversion count can't be found since course code can't be negative.

Error Messages for Invalid Entries:
Student 3: ERROR: Inversion count can't be found due to the presence of boolean values.
Student 7: ERROR: Inversion count can't be found due to the presence of boolean values.
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

Karatsuba:

positive test cases:

1:

```
/opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/karatsuba-positive-tc1.py"
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/karatsuba-positive-tc1.py"
Enter a 10-digit number: 1234567890
Enter a 10-digit number: 9876543210
The product of 1234567890 and 9876543210 is: 12193263111263526900
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

2:

```
/opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/karatsuba-positive-tc2.py"
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/karatsuba-positive-tc2.py"
Enter a 50-digit number: 98467262402804811356329967086984724476555713969185
Enter a 50-digit number: 98467262402804811356329967086984724476555713969185
The product of 98467262402804811356329967086984724476555713969185 and 98467262402804811356329967086984724476555713969185 is: 9695801765102817951620352507019
57452925089913954541224763196945493457086136035197655966734712956422 5
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

3:

```
/opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/karatsuba-positive-tc3.py"
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/karatsuba-positive-tc3.py"
Enter a 100-digit number: 1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
Enter a 100-digit number: 9876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210
The product of 1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890 and 9876543210987654321098765432109876543
21098765432109876543210987654321098765432109876543210987654321098765432109876543210 is: 12193263113702179522618503273386678859451150739156363359236761164455788599298790108215
20013565005212360920580111263525898643499378616064616736777929561194939744871208653362292333223746380111126352690 0
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

4:

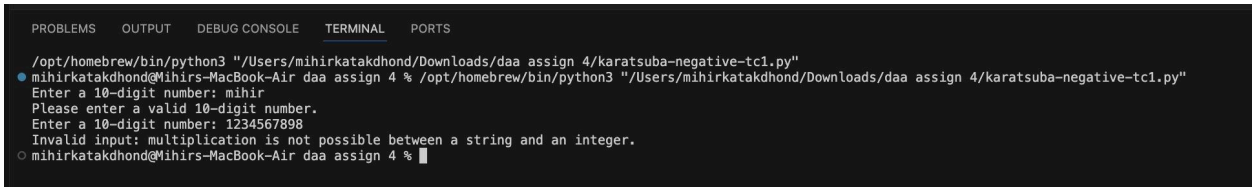
```
/opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/karatsuba-positive-tc4.py"
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/karatsuba-positive-tc4.py"
Enter a 500-digit number: 513845047929725806080113515607756886916218042115686510365161976093847477892565196603293337341908390169285578397845676128463761778040081138932370122520768341861029958753073247416574474237911238210842114464266477803482530949859920320099210223407968514822658433328825579032956701118215021725095472329295688047682438009007013426957648451209110301135566325380835731686100101602992950508267795490079760828657585419566965238058159757641576472398244908501373734568000624751088085267690220139197730409887498 97
Enter a 500-digit number: 15963517391266864325809744125351989023728953144277333735057821877150770429125486381446735252841973264155819278792428017345948441461612705266097513930133950793611093295148779194585137265528880388154898207900581491372659573055236664773357493713275017174681266902330572708392770691634910968377150823079780739947551751143419574551148785871844969110531186539649603341197410287293927141790341935477053464507753363720489010853272781594626704361368876546135097328121138708634650385382277872121047908115636933
The product of the two 500-digit numbers is: 8202774359042533363507827579235534089156110602748944344960305166716781101910588823971661161670848640770962284051229407526093654418503346650330344070485294584489430758572310887730877116807139287368340519378672469354825942609240641975502599751168410132346853291858338301120075774093326459773018517174225856525706405996814417720587245840608441552168262985979382763299730450806021891170760837863270355042865516924225324251939736706453937259752048564989627885618358410471468967553475432663007869509495927689057738010649436629395957653932794236638022276082697678666314183172082123023517486333191320679766392903862871922216414804651936426808098391734213245249113461767757824272352679498099115125402259560685091607109937202001548714481474294065772888284281517713513507082157626232413559218480351224711094946853514384033296225647288067983478530917038076041251289190422151962835858281730315268274579042221802977747164155551934102388818612928130399870745024361150075004172768942661943111031473083087801593145901
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

5:

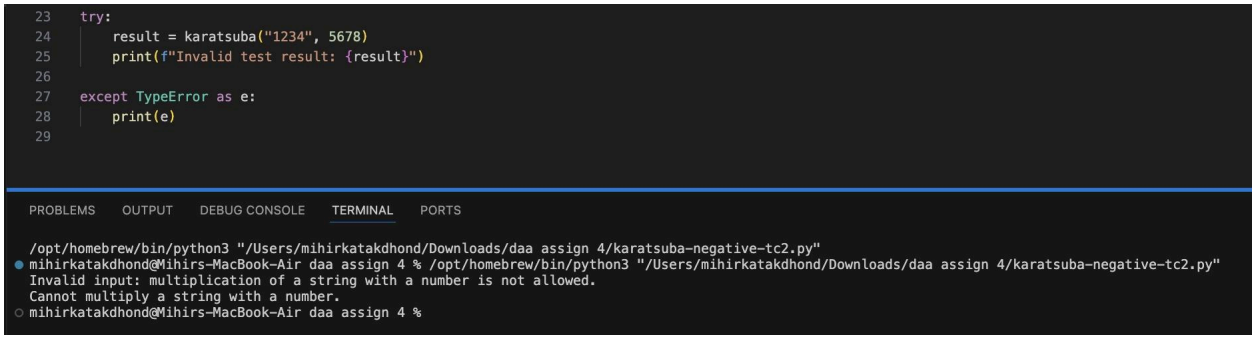
```
/opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/karatsuba-positive-tc5.py"
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/karatsuba-positive-tc5.py"
Enter a 1000-digit number: 5191668756721169362737692194543949924928520587584229019318067243287971270541036150175518110560777318058882556555350917285223602415367673458220447427124007564202562301891140173316689974810874619758977797405425276670294736838790710062791113787813420905441221530120693608149388409799352800623321562953829157226739186410040434526480105062980056059309321865643839392821903648186065418770836908641832038903714340800467105681870407620589387021086133645846345340762367199156513472251053257125647488371156695149052222760443204812727617993754975187218444750202474915202535566493357907690625074187191498654297542744680098697606852816711491139528537805955571498692085677654352958740124018076810659669391696191542198650871768184700199160047337420754655680653879536267596103656564548619903058797035246965552152547810470642549578288383236326461670038661364172920472322265584911586931366956276755646949607083358482606759872242218029777471641555519341023881861292813039987074502436115007500417276894266194311103147308308780159314590 1
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

Negative test cases:

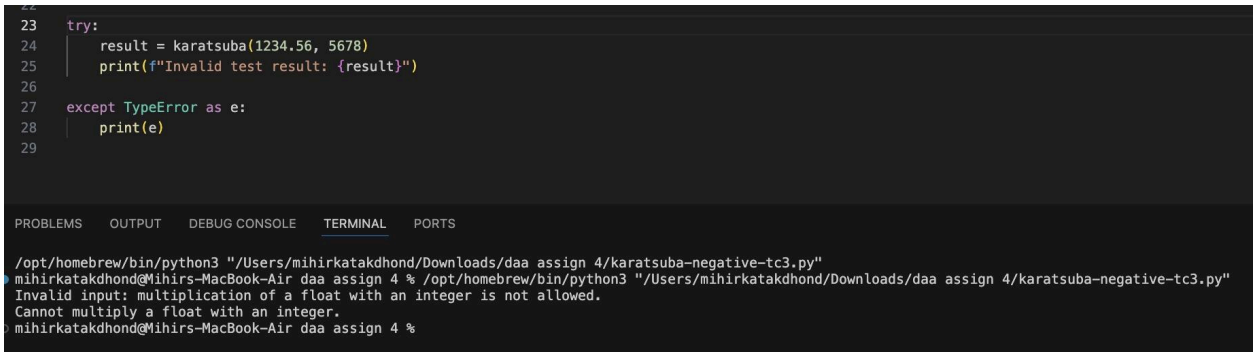
1:



2:



3:



4:

5:

```
try:
    result = karatsuba(1+2j, 5678)
    print(f"Test result: {result}")

except TypeError as e:
    print(e)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
/opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/karatsuba-negative-tc5.py"
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/karatsuba-negative-tc5.py"
Invalid input: multiplication of a complex number is not allowed.
Cannot multiply a complex number with a number.
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

Brute force:

Positive test cases:

```
/opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/bruteforce-positive.py"
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/bruteforce-positive.py"
Multiplication of two 10-digit numbers:
Number 1: 4653751900
Number 2: 4764032143
Product: 22170623637147321700

Multiplication of two 50-digit numbers:
Number 1: 77757962536886034087007993281692404590968833985652
Number 2: 67250716438210385128915093465122890157947247241256
Product: 5229278689381108912880274552408808741722980874123473186079542494931559144209434792711254791208645891 2

Multiplication of two 100-digit numbers:
Number 1: 476201900217851837489386659161460758663411659132761721043230332734520950510689225973783848932303179 4
Number 2: 403319580065645241355814770946613673979343440726350946802155499726654383198390200600453141765819276042
Product: 1920615504223263002723461882552442593249764282090395435004887652901656676187408202761682829309831815502303033682537452797924640398387895105175587210234215497802065833205498276158902273474449162 8479348

Multiplication of two 500-digit numbers:
Number 1: 36848094323869475159776279112722973008674451942257406812849936489758759329211795582779347226791536012089722382292956774926333897834343260087178319994734621278432409041446784759035996106256695 4
Number 2: 90914806991547592469762865525510810886260541400242301919031202477616371849675160091689530516488348498601356023257731099453967672231217699447274149162929598774892222200791533034590642503451356058860122 8752199176633660685716534344354964364700328615207030421403628865594159705136802536012111311162135624368606 64
Number 2: 19653998887962737570988525774567298889311486623821315729254855473898267975667526193833487669159698477994795141393277370470249738457194146322681641564622719898716294195895005960401370891053730265 1472890012401210632948009250623165567750392384799462921566589723816858132427712709206007383796312729455239604816603939394354261180660762691296532470100891288108449504146627176139164567108099928355838874 3156722690410831546608656425403516488354957906722256613880455683588325094998490246054907072157012844933437
Product: 7242124048648767271287813204578262607679225330725317750036623855154508247137917214426346730388252543528863960748138270377746735994434085779133004631582286727871405543571755493252594997724467418 2174480172600254054591445217935845700887294712414055517893458014085131289439480224099560440464438843053230052459951754870282245766968224220540980708557320104062983711667401526829009742130538340810511512 8804763033115095421944438526163696748247665583470250339429792188395842515492117005807232897537457536924559111728365534915831009575269987313530882258098174888036148566447466093376291412589514958262662303002 8880241782239255605057230034681308116075526012549598775819565222929618480730523718433964131105007746066974213004934187724745111792708163377098317852457604888690439645198589417064662721644079447518226086 6197703292694163815515897089850804967101149428064488027440295820200165171537122259979919811917139888280302032040677125966014182254594557196278581744075937409982308228496250445620781791036674432362168

Multiplication of two 1000-digit numbers:
Number 1: 9490786300883575909842520201766383139423476599330824554994224711288404889021755978508497293037762738510144151875942498339301922831246832207369277698201159515345539172957721292909551150341348 5
Number 2: 2963404400268998099352467245510944800519716703969067583158422446878757939869883033025473614838044609297188740276444857602555297129679282136146223014241672288827408397272686398580440565375420831 2506239626012872265334197455306414271485169227014174726490888158013514721637173172297481439549556930733709911748908605534598835822349650113647442155726595711548888145790119411447906550945150047103882 8225050305247275148361992484189376004134528590922640785133545951181769178059038700017302423373155035556353361683096716503057801097438585736956427845234255950342906150721138360773741674180430580331691 5840512265444399753470775768319804457520565584129654212654544043021007500654519069772930917775201538140321913320798516107856414474544594056069997291743884806629909364897113346519277406989471403546070541 2025913503465795034877297315433133873191749106788017487983379034362813792407696250451184116790589719186565072181043413116659054630862483567568201375050090187077159437260726554883399536586198726828675 4 Product: 2812503788685111647929877498224639452337320926967248854573408135865529097049872951081525463151557137563008938905241397732169276311831657562633748882845767051362271517824855715285728934078572851 6454828400923483197950498819256269730638126348424669734197589857058608507452643391233373127599180875484897305198622511313440713258751428114098531713040648101728892003303053097992240931629337448039622 0 9095393025425558446726124297890954132173266993205825492634796299303384312465400006597966988182538272452433386708294678989246788244875607156626199738537383902208606350085364547753165523477358440518445145 6433807525001194187393487213377090813770584302719393972959964215831612728877153951899355265378953166517545868199810199953530963991351253435058802434774326529725878680037882605008419056073148578859301349 2 7994720097656168713177830870166550961776246706051857891900156828792970331387749333905465616732007078600748033588033305561423565311619418206233398429523494141846142646618891403379102213368557545963491 6 3487542497724651463847217509629333996735650743819457835451034562719366643798057070838984365314948713617431907629026124343987386211642265338166760938900714544265679755453437418141035848077379750991762 9 3800408921279047776158395949738286537809283626162006650136755871216099229648068255839507339949213426078283815995045333718314299785866728164229135823641091450385315244955242899083130392144677993139986561 4 0154493626600728945205065983557899308988208051939699813203842844542500378848395731981071759866776295815732413903779853950358741999285075950150210329576356069745936722238686232059829972969875343998913 1 6379776840456465234085891351820072669878198917015616490888918760827255590028191729728421010421295638849754284416094417728291656555100675277933261294791583478755069381587912146306935060272375866855402239 5 8969984683442123693092156635107353444997455045718508237491457022028464017171718801168479420347563087660270891825853171159572071256806405838663684674421309225823458308418054353686945729974775 1
```

Negative test cases:

1:

```
try:
    result = brute_force_multiplication("1234", 5678)
    print(f"Multiplication result: {result}")
except TypeError as e:
    print(f"Error: {e}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
/opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/bruteforce-negative-tc1.py"
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/bruteforce-negative-tc1.py"
Error: Both inputs must be integers for multiplication, here one number is string which is not allowed.
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

2:

```
try:
    result = brute_force_multiplication("mihir", 5678)
    print(f"Multiplication result: {result}")
except TypeError as e:
    print(f"Error: {e}")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

/opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/bruteforce-negative-tc2.py"
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/bruteforce-negative-tc2.py"
Error: Invalid input: One of the inputs is text. Multiplication requires numbers.
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

3:

```
try:
    result = brute_force_multiplication(True, 5678)
    print(f"Multiplication result: {result}")
except TypeError as e:
    print(f"Error: {e}")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

/opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/bruteforce-negative-tc3.py"
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/bruteforce-negative-tc3.py"
Error: Invalid input: One of the inputs is a boolean. Multiplication requires integers.
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

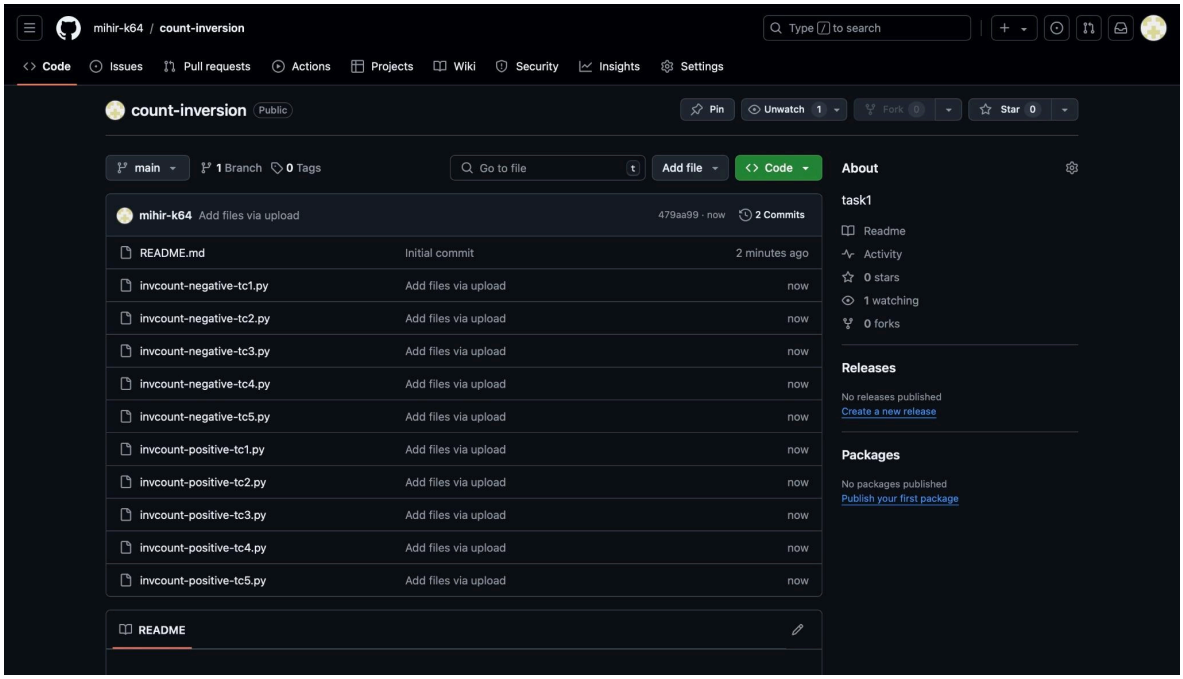
4:

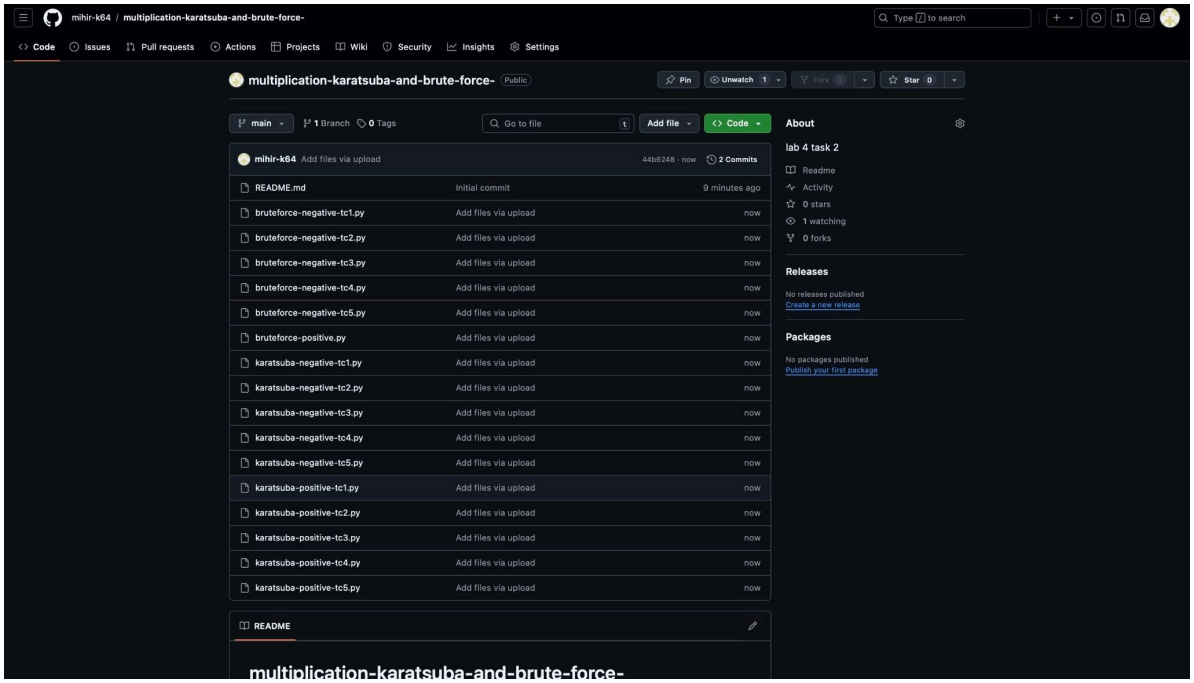
```
try:
    result = brute_force_multiplication(3 + 4j, 5678)
    print(f"Multiplication result: {result}")
except TypeError as e:
    print(f"Error: {e}")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

/opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/bruteforce-negative-tc4.py"
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/bruteforce-negative-tc4.py"
Error: Invalid input: One of the inputs is a complex number. Multiplication requires integers.
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

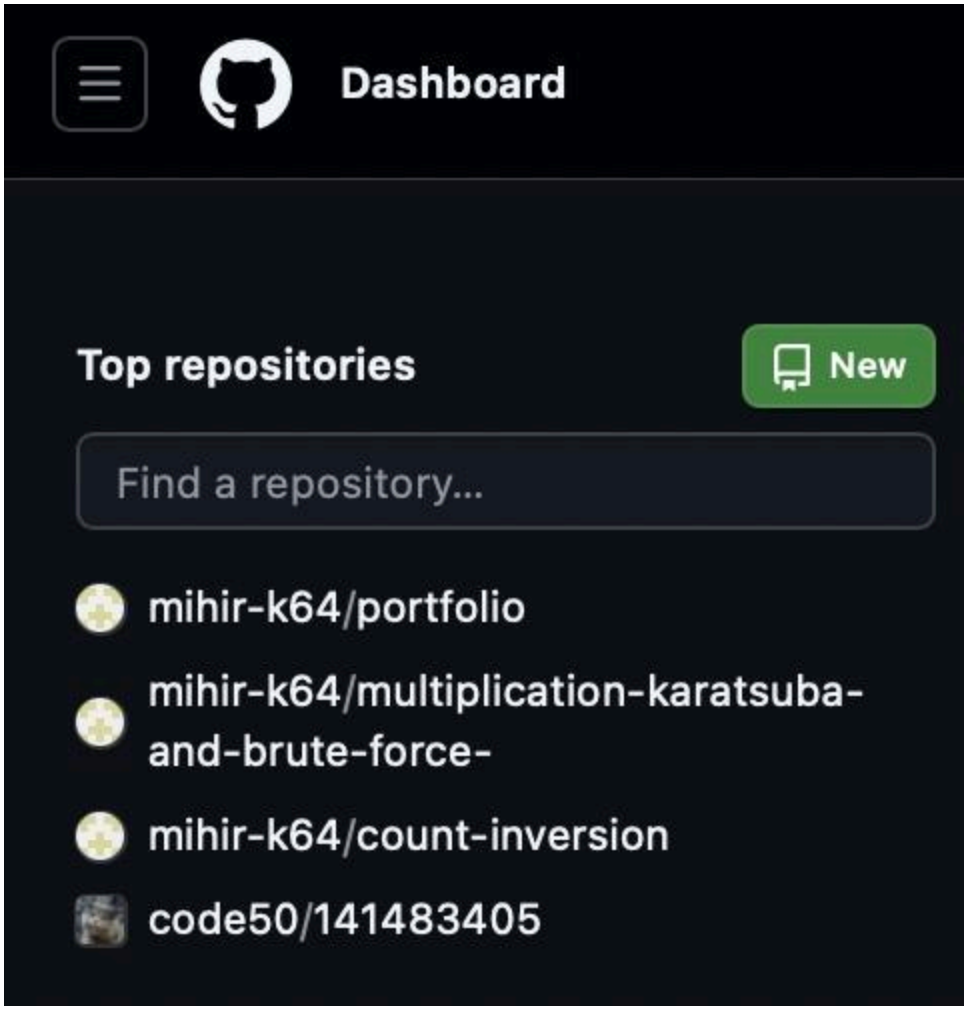
5:

```
try:
    result = brute_force_multiplication(1234.56, 5678)
    print(f"Multiplication result: {result}")
except TypeError as e:
    print(f"Error: {e}")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

/opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/bruteforce-negative-tc5.py"
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 % /opt/homebrew/bin/python3 "/Users/mihirkatakdhond/Downloads/daa assign 4/bruteforce-negative-tc5.py"
Error: Invalid input: One of the inputs is a floating-point number. Multiplication requires integers.
mihirkatakdhond@Mihirs-MacBook-Air daa assign 4 %
```

Github repo:


mihir-k64 / count-inversion
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
count-inversion
Public
Pin
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
Add file
Code
About
task1
Readme
Activity
0 stars
1 watching
0 forks
mihir-k64 Add files via upload
479aa99 · now
2 Commits
README.md Initial commit 2 minutes ago
invcount-negative-tc1.py Add files via upload now
invcount-negative-tc2.py Add files via upload now
invcount-negative-tc3.py Add files via upload now
invcount-negative-tc4.py Add files via upload now
invcount-negative-tc5.py Add files via upload now
invcount-positive-tc1.py Add files via upload now
invcount-positive-tc2.py Add files via upload now
invcount-positive-tc3.py Add files via upload now
invcount-positive-tc4.py Add files via upload now
invcount-positive-tc5.py Add files via upload now
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
README


mihir-k64 / multiplication-karatsuba-and-brute-force-
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
multiplication-karatsuba-and-brute-force-
Public
Pin
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
Add file
Code
About
lab 4 task 2
Readme
Activity
0 stars
1 watching
0 forks
mihir-k64 Add files via upload
44b6248 · now
2 Commits
README.md Initial commit 9 minutes ago
bruteforce-negative-tc1.py Add files via upload now
bruteforce-negative-tc2.py Add files via upload now
bruteforce-negative-tc3.py Add files via upload now
bruteforce-negative-tc4.py Add files via upload now
bruteforce-negative-tc5.py Add files via upload now
bruteforce-positive.py Add files via upload now
karatsuba-negative-tc1.py Add files via upload now
karatsuba-negative-tc2.py Add files via upload now
karatsuba-negative-tc3.py Add files via upload now
karatsuba-negative-tc4.py Add files via upload now
karatsuba-negative-tc5.py Add files via upload now
karatsuba-positive-tc1.py Add files via upload now
karatsuba-positive-tc2.py Add files via upload now
karatsuba-positive-tc3.py Add files via upload now
karatsuba-positive-tc4.py Add files via upload now
karatsuba-positive-tc5.py Add files via upload now
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
README
multiplication-karatsuba-and-brute-force-


Dashboard
Top repositories
New
Find a repository...
mihir-k64/portfolio
mihir-k64/multiplication-karatsuba-and-brute-force-
mihir-k64/count-inversion
code50/141483405