

---

**Started on** Thursday, 3 March 2022, 3:09 PM

---

**State** Finished

---

**Completed on** Friday, 4 March 2022, 2:19 PM

---

**Time taken** 23 hours 10 mins

---

**Marks** 540.00/540.00

---

**Grade** **100.00** out of 100.00

Question **1**

Correct

Mark 80.00 out  
of 80.00

Time limit

1 s

Memory limit

64 MB

Diberikan sebuah file deklarasi sekaligus definisi [Vector2](#), yakni sebuah kelas yang merepresentasikan vektor, sama seperti vektor pada fisika atau matematika ( **bedakan dengan STL vector C++**, yang sebenarnya adalah array dinamis).

Tugas anda adalah mengubah kelas ini menjadi generic, yakni:

- mampu menerima tipe data apa saja
- mampu menerima panjang vector berapa saja

Contoh program utama:

```
int main() {
    Vector<int, 4> v1, v2;

    cout << "Masukkan vektor 4 elemen: ";
    cin >> v1;

    v2[0] = -1;
    v2[1] = -2;
    v2[2] = -3;
    v2[3] = -4;

    cout << v1 << " + " << v2 << " = " << v1 + v2 << endl;
    cout << v1 << " - " << v2 << " = " << v1 - v2 << endl;

    return 0;
}
```

Kumpulkan **Vector.hpp**

C++14

 [Vector.hpp](#)

Score: 80

Blackbox

Score: 80

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 3.02 MB
2	10	Accepted	0.00 sec, 2.97 MB
3	10	Accepted	0.00 sec, 2.89 MB
4	10	Accepted	0.00 sec, 2.99 MB
5	10	Accepted	0.00 sec, 3.02 MB
6	10	Accepted	0.00 sec, 2.98 MB
7	10	Accepted	0.00 sec, 3.03 MB
8	10	Accepted	0.00 sec, 3.14 MB

Time limit	1 s
Memory limit	64 MB

Di python, kita dapat membuat tuple, yakni gabungan dari beberapa nilai, misalnya `(123, "abc", bool)`. Di C++, kita juga dapat menggunakan STL `pair<>` untuk menggabungkan dua nilai. Tapi, kali ini Anda membutuhkan tuple untuk tiga nilai. Karena itu, buatlah sebuah kelas bernama `Triplet` yang dapat menampung tiga nilai berbeda.

Anda sudah membuat [Triplet.hpp](#) yang menampung tiga nilai integer, buatlah menjadi 3 tipe data berbeda

Contoh Kode	Contoh Output
<pre>Triplet&lt;int, int, int&gt; a(3, 2, -5);  cout &lt;&lt; a.getFirst() &lt;&lt; endl; cout &lt;&lt; a.getSecond() &lt;&lt; endl; cout &lt;&lt; a.getThird() &lt;&lt; endl;</pre>	<pre>3 2 -5</pre>
<pre>Triplet&lt;string, float, string&gt; a("abc", 4.5, "def"); Triplet&lt;string, float, string&gt; b("ghi", -1.0, "def"); Triplet&lt;string, float, string&gt; c("abc", 4.5, "def");  if (a == b) {     cout &lt;&lt; "a == b" &lt;&lt; endl; } if (a == c) {     cout &lt;&lt; "a == c" &lt;&lt; endl; }</pre>	<pre>a == c</pre>

Submit file **Triplet.hpp**

C++14

 [Triplet.hpp](#)

Score: 120

Blackbox

Score: 120

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.89 MB
2	10	Accepted	0.00 sec, 2.87 MB
3	10	Accepted	0.00 sec, 2.92 MB
4	10	Accepted	0.00 sec, 2.96 MB
5	10	Accepted	0.00 sec, 2.87 MB
6	10	Accepted	0.00 sec, 3.02 MB
7	10	Accepted	0.00 sec, 3.09 MB
8	10	Accepted	0.00 sec, 3.06 MB
9	10	Accepted	0.00 sec, 3.05 MB
10	10	Accepted	0.00 sec, 3.14 MB
11	10	Accepted	0.00 sec, 3.01 MB
12	10	Accepted	0.00 sec, 2.93 MB

Time limit	1 s
Memory limit	64 MB

Diberikan header kelas Car yang merupakan representasi dari objek mobil. Pada kelas Car ini terdapat berbagai fitur yang dapat dijalankan pada mobil pada umumnya. Kalian diminta untuk membuat implementasi kelas Car sesuai dengan header file yang diberikan dengan memperhatikan kaidah **Exception Handling**.

[Car.h](#)

Kumpulkan file **Car.cpp**.

C++14

 [Car.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	7	Accepted	0.00 sec, 2.91 MB
2	7	Accepted	0.00 sec, 2.91 MB
3	7	Accepted	0.00 sec, 2.83 MB
4	7	Accepted	0.00 sec, 2.82 MB
5	8	Accepted	0.00 sec, 2.96 MB
6	8	Accepted	0.00 sec, 2.88 MB
7	8	Accepted	0.00 sec, 2.89 MB
8	8	Accepted	0.00 sec, 2.89 MB
9	8	Accepted	0.00 sec, 3.00 MB
10	8	Accepted	0.00 sec, 2.83 MB
11	8	Accepted	0.00 sec, 2.97 MB
12	8	Accepted	0.00 sec, 2.84 MB
13	8	Accepted	0.00 sec, 2.88 MB

Time limit	1 s
Memory limit	64 MB

Anda sedang mengerjakan proyek sebuah bank. Ternyata, programmer sebelum Anda tidak belajar OOP dengan baik, dan tidak menggunakan Exception. Sebagai programmer berkualitas, Anda ingin membenahi program Bank. Lengkapilah program Bank dengan:

[Account.cpp](#), [Account.hpp](#), [Bank.hpp](#), [BaseException.hpp](#)

- Definisikan `TransactionFailedException` dalam file `Bank.hpp`
- Definisikan `AccountNotFoundException` dalam file `Bank.hpp`
- Tambahkan exception pada method `withdraw` dan `add` dalam file `Account.cpp`
- Tambahkan exception pada method `findAccountIdx` dalam file `Bank.hpp`
- Tambahkan exception handling pada method `transfer` dalam file `Bank.hpp`
- Lengkapi method `transfer` dalam file `Bank.hpp`

Sebagai catatan:

- Dalam soal ini, semua exception yang dilempar berupa pointer, bukan instance
- Pada penulisan exception, `"\n"` menandakan karakter newline / `endl`

Contoh Kode	Contoh Output
<pre>Bank b; b.getAccount(0).setNumber("10"); b.getAccount(0).add(100); b.getAccount(1).setNumber("20"); b.getAccount(1).add(120);  b.transfer("10", "20", 50);  cout &lt;&lt; b.getAccount(0).getBalance() &lt;&lt; endl; cout &lt;&lt; b.getAccount(1).getBalance() &lt;&lt; endl;</pre>	<pre>50 170</pre>
<pre>Bank b; b.getAccount(0).add(-20);</pre>	<pre>// Dilempar InvalidNumberException* // dengan printMessage menuliskan: Bilangan -20 tidak valid</pre>
<pre>Bank b; b.getAccount(0).setNumber("10"); b.getAccount(0).add(100); b.getAccount(1).setNumber("20"); b.getAccount(1).add(120);  b.transfer("10", "20", 150);</pre>	<pre>// dilempar TransactionFailedException* // dengan printMessage menuliskan: Transaksi gagal dengan pesan kesalahan: Membutuhkan 150 namun hanya tersedia 100</pre>

Kumpulkan **Bank.zip** berisikan `Account.cpp` dan `Bank.hpp`

C++14

 [Bank.zip](#)

Score: 140

Blackbox

Score: 140

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.96 MB
2	10	Accepted	0.00 sec, 2.96 MB
3	10	Accepted	0.01 sec, 3.00 MB
4	10	Accepted	0.00 sec, 2.96 MB

No	Score	Verdict	Description
5	10	Accepted	0.00 sec, 2.96 MB
6	10	Accepted	0.00 sec, 2.96 MB
7	10	Accepted	0.00 sec, 3.05 MB
8	10	Accepted	0.00 sec, 2.91 MB
9	10	Accepted	0.00 sec, 3.03 MB
10	10	Accepted	0.01 sec, 2.96 MB
11	10	Accepted	0.00 sec, 2.99 MB
12	10	Accepted	0.00 sec, 2.95 MB
13	10	Accepted	0.00 sec, 2.95 MB
14	10	Accepted	0.00 sec, 3.06 MB

Time limit	1 s
Memory limit	64 MB

Hojun dan Qila sedang bermain. Hojun memiliki N buah kata, dan Qila memiliki M buah kata. Baik kata-kata Hojun maupun Qila, mungkin ada kata yang terulang.

Mula-mula, Hojun akan menuliskan N kata tersebut di atas kertas. Lalu, Qila akan membacakan kata-katanya satu demi satu. Untuk tiap kata, Hojun akan mencoret kata tersebut bila ada di kertas. Jika tidak ada (atau ada, tapi sudah dicoret semua), maka Hojun akan diam saja.

Kini tugas Anda adalah membuat program yang menerima N, daftar kata Hojun, M, dan daftar kata Qila. Program akan menuliskan daftar kata yang **tidak** dicoret di atas kertas, dipisahkan oleh baris baru. Urutan dari kata yang dituliskan sembarang.

Contoh Input	Contoh Output
4 Budi Budi adalah Budi 4 Budi Bermain Bola Adalah	Budi Budi adalah
4 Saya Suka Pemrograman C++ 3 Tuan Putri Tidur	Saya Suka Pemrograman C++
2 Saya Mengantuk 5 Saya Ingin Tidur Karena Mengantuk	

Anda diharuskan menggunakan minimal kakas dari STL yakni [Map](#). Untuk memudahkan, lihat juga contoh penggunaan map [di sini](#).

Kumpulkan **main.cpp**

C++14

 [main.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.10 sec, 2.94 MB
2	10	Accepted	0.00 sec, 2.89 MB
3	10	Accepted	0.00 sec, 2.93 MB

No	Score	Verdict	Description
4	10	Accepted	0.00 sec, 2.94 MB
5	10	Accepted	0.01 sec, 3.01 MB
6	10	Accepted	0.00 sec, 2.92 MB
7	10	Accepted	0.00 sec, 2.94 MB
8	10	Accepted	0.00 sec, 2.93 MB
9	10	Accepted	0.02 sec, 2.91 MB
10	10	Accepted	0.00 sec, 3.02 MB

[◀ Vector2.hpp](#)

Jump to...