Complete ALL questions. Download the give main.cpp file and complete your work on the file.

1. Use <tuple> to create a function q1() that return 3 values (int, double, and string) as 1, 2.3, and "456", respectively.
2. Variadic templates can take any variable(zero or more) number of arguments. In C++, templates can have a fixed number of parameters only that have to be specified at the time of declaration. variadic templates help to overcome this issue.

Write a variadic template function printfq2 that takes a package of parameters; unpack the parameter package; and print all parameters. Use either approach discussed in classes.

1. Create a function get\_student(int id) and returns a student’s details in a <tuple> of <double, char, string>. The return type of the function should be set to auto.

Student details as following:

* if id = 0, the returned tuple is (3.8, ‘A’, “John”)
* if id = 1, the returned tuple is (2.9, ‘C’, “Jack”)
* if id = 2, the returned tuple is (1.7, ‘D’, “Ive”)
* Other ids, the returned tuple is (0.0, ‘U’, “Null”)

1. Follow the steps and use <concept> to modify the template:
   1. Use the same generic print method without the concept first.

template <typename T>  
void printQ4(const T& msg){  
 std::cout << msg;  
}

* 1. Create a class Foo with an int data member, say intData\_. Initialize the data member to a desired value.
  2. Instantiate an object Foo and try to print its value by passing it to the generic print method.
  3. Transform the generic print method to a concept based print method like in mentioned in lecture notes.
  4. Again observe the error when trying to print the variable of type Foo.
  5. Complete the interface of Foo by providing an overloaded operator<< to print the value of the int data member within Foo.

1. Copy the following code into [Wandbox](https://wandbox.org/) online compiler (<https://wandbox.org>[/](https://wandbox.org/)) .

*// rangesV3FilterTransform.cpp*

#include <iostream>

#include <range/v3/all.hpp>

#include <vector>

void Q5(auto& v){

// put your work below…

// put your work above

for (auto v: results) std::cout << v << " ";

}

**int** main() {

**std::vector<int> numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};**

**Q5(numbers);**

}

Develop void Q5() by applying *Ranges* library to create views of:

* 1. Use ranges::views::filter to pick all values that are not divisible by 3, then
  2. Use ranges::views::transform to multiply all elements by 5, then
  3. Use ranges::views::take(…) to select the first 4 values.

Display the final results.

Note that your compiler may not be able to compile this programme as <Range> is not ready for most of the compilers on the market. Use Wandbox with gcc HEAD 12.0.0 20210 or above version to compile. (Simply click the link provided above will take you to Wandbox.)

Copy your work on void Q5() to the main.cpp file you have worked on the previous question. Turn your code of this question (Q5) as comment so that the compiler will ignore this code.