Part 3 - Practice

2 OF 2 QUESTIONS REMAINING

Test Content

Question 1

10 Points

Given the following code snippet, complete the tasks listed below:

```
1.#include<iostream>
2.#include<string>
3.#include<algorithm>
4.#include<numeric>
5.#include<vector>
6.usingnamespace std;
7.
8.structComputerParts{
9.string partType;
10.string model;
11.double price;
12.double powerConsumption;
13.bool used;
14.};
15.
16.int main(){
17.
18.  vector<ComputerParts> wishList;
19.
20.}
```

Task 1. Add to the wishLish some number of desired Computer Parts
Eg. GPU, RTX 3080, 1000, 300, false
Task 2. Sort the wishList by price in ascending order
Task 3. Print to the screen the part in the wishList the part that has the most and least powerConsumption
Task 4. Remove from the wishList all the parts that are 'used'
Task 5. Find the average cost of all the parts in the wishList
Task 6. Change all the items in the wishList to 'used' and reduce their price by 20% and save this result to a new vector
Task 7. Figure out how many of a given part type is in the wishList eg. how many CPUs and print to the screen
Task 8. Create new wishList of computer parts and add parts to it. Combine it with the original wishList

Use the editor to format your answer

Question 2

10 Points

Provide the output from the following program. The lines commented with "Output" shows where the lines are generated.

```
#include <iostream>
#include <string>
#include <vector>
#include <list>
#include <numeric>
#include <algorithm>
using namespace std;

struct Headphone {
  string description;
  double price{};
  bool wireless;
  bool noise_cancelling;
  bool used;
};

Headphone shopping(std::vector<Headphone> phones) {
  // Task 1
  {
    sort(phones.begin(), phones.end(), [](const Headphone& h1, const Headphone& h2) {
return h1.price > h2.price; });
    cout << phones.front().description << endl; // Output 1
    cout << phones.back().description << endl; // Output 2
  }
  // Task 2
  {
    for_each(phones.begin() + 1, phones.end() - 1, [](const Headphone& phone) {if
(phone.wireless && phone.noise_cancelling) {
      cout << phone.price << endl; // Output 3
    }
    });
  }
  vector<Headphone> phones2;
  // Task 3
  {
    auto num = count_if(phones.begin(), phones.end() - 2, [](const Headphone& phone) {
return phone.wireless; });
    phones2.resize(num);
    copy_if(phones.begin(), phones.end() - 2, phones2.begin(), [](const Headphone&
phone) { return phone.wireless; });
    cout << "Num: " << num << endl; // Output 4
    cout << phones2.front().description << endl; // Output 5
  }
  // Task 4
  {
```

```
    transform(phones2.begin(), phones2.end(), phones2.begin(), [](Headphone& phone)
    {
        if (phone.used) { phone.price -= phone.price * 0.2; }return phone;
    });
    cout << phones2.back().price << endl; // Output 6
  }
  Headphone ret;
  // Task 5
  {
    auto itr = find_if(phones2.begin(), phones2.end(), [](const Headphone& phone) {
return phone.noise_cancelling && phone.used; });
    ret = *itr;
    if (itr != phones2.end()) // Output 7
      cout << itr->description << endl;
    else
      cout << "Nothing found" << endl;
  }
  return ret;
}

int main() {
  std::vector<Headphone> phones({ {"Denon D5000", 350.99, false, false, true},
                                  {"Air Pods", 200.24, true, false, true},
                                  {"Sony 1000XM3", 300.25, true, true, true},
                                  {"Senhesier HD 650", 500, false, false, false},
                                  {"Sony 1000XM4", 400.50, true, true, false},
                                  {"Shure SE215", 99.99, true, true, true} });
  auto phone = shopping(phones);
}
```

Use the editor to format your answer

Questions Filter (2) ▼

Save and Close

Submit