Subject 1

1 Specificati si testati functia: (1.5p)

```
int f(vector<int> l) {
     if (l.size() < 1)
          throw std::exception("Illegal argument");
     map<int, int> m;
     for (auto e : l) {
          m[e]++;
     }
     int rez{ 0 };
     for (auto e : m) {
          if (e.second > 1) rez++;
     }
     return rez;
}
```

2 Indicați rezultatul execuției pentru următoarele programe c++. Daca sunt erori indicați locul unde apare eroarea si motivul.

```
//2 a  (1p)
#include <iostream>
using namespace std;
class A {
public:
 virtual void print() {
   cout << "printA" <<endl;
 }
 ~A() {cout << "~A" << endl;}
};

class B: public A {
public:
 void print() {
   cout << "printB" << endl;
 }
 virtual ~B() {
    cout << "~B" << endl;
 }
};
int main() {
      A* o1 = new A();
      A* o2 = new B();
      o1->print();
      o2->print();
      delete o1;
      delete o2;
      return 0;
}
```

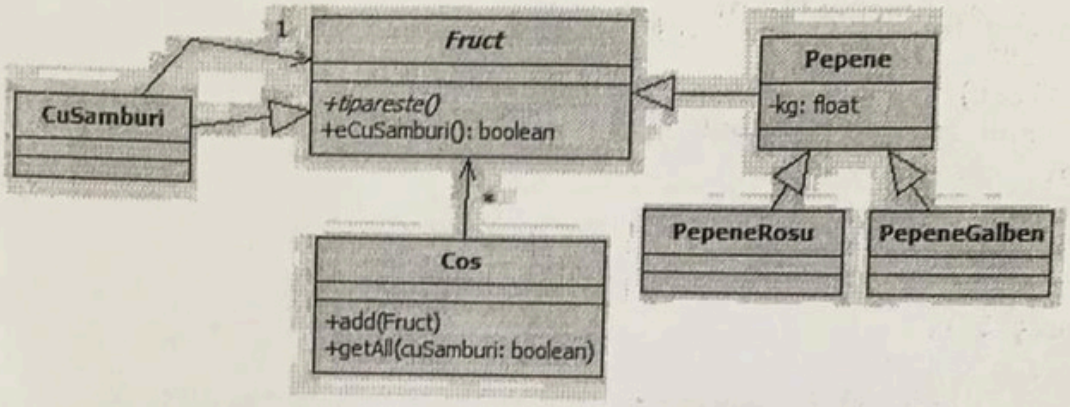
printA
printB
~A
~A

```
//2 b  (0.5p)
#include <iostream>
#include <deque>
#include <string>
using namespace std;

int main() {
     deque<string> s;
     s.push_back("1");
     s.push_back("2");
     s.push_back("3");
     s.push_front("1");
     s.push_front("4");
     auto it = s.begin();
     auto end = s.end();
     end = end - 1; it++;
     while (it != end) {
          cout << *it;
          it++;
     }
     return 0;
}
```

112

3 Scrieți codul C++ ce corespunde diagramei de clase UML. **(2p)**



- Clasa abstracta **Fruct** are o metoda pur virtuala *tipărește*.
- Metoda *tipărește* din clasa **Pepene** tipărește kilogramele apoi cuvântul "pepene"
- Clasa **PepeneRosu** si **PepeneGalben** pe lângă ce tipărește clasa de baza mai tipărește cuvântul "rosu" respectiv "galben" .
- Clasa **CuSamburi** tipărește pe lângă ce tipărește fructul agregat de el si textul "cu samburi". Metoda eCuSamburi returnează adevărat.
- Metoda *add* din clasa **Cos** permite adăugarea de orice fruct, iar metoda *getAll* returnează doar fructele cu sâmburi sau doar fructele fără sâmburi (in funcție de parametru). Implicit toate fructele sunt fără sâmburi daca nu sunt decorate cu **CuSamburi**.
- Scrieți o funcție C++ care creează un obiect **Cos** si adaugă următoarele fructe (alegeți voi kilogramele pentru fructe): un pepene roșu fără sâmburi, un pepene galben cu sâmburi, un pepene galben fără sâmburi si un pepene roșu cu sâmburi. Tipăriți separat fructele cu sâmburi si fructele fără sâmburi din cos. Creați doar metode si atribute care rezulta din diagrama UML(adăugați doar lucruri specifice C++ ex: constructori). Nu adăugați câmpuri, metode, nu schimbați vizibilitatea. Implementați corect gestiunea memoriei. Folosiți STL unde exista posibilitatea **(2p)**

4 Definiți clasa Adder astfel încât următoarea secvența C++ sa fie corecta sintactic si sa efectueze ceea ce indica comentariile. (2p)

```
void adder() {
      Adder<int> add{ 1 };//construim un adder,pornim cu valoarea 1
      //se adauga valorile 7 si 3
      add = add + 7 + 3;
      add + 8;
      //tipareste suma (in acest caz:19 rezultat din 1+7+3+8)
      cout << add.suma() << "\n";
      //elimina ultimul termen adaugat
      add--;
      //tipareste suma (in acest caz:11 rezultat din 1+7+3)
      cout << add.suma() << "\n";
      --add--;
      //tipareste suma (in acest caz:1)
      cout << add.suma() << "\n";
}
```