Subiect 3

1 Specificați si testați funcția: **(1.5p)**

```
using namespace std;
#include <vector>
#include <algorithm>
#include <stack>
vector<string> g(vector<string> l) {
	if (l.size() == 0)	throw exception("Illegal argument");
	std::stack<string> st;
	for (auto& s : l) {
		st.push(s);
	}
	vector<string> r;
	while (!st.empty()) {
		r.push_back(st.top());
		st.pop();
	}
	return r;
}
```

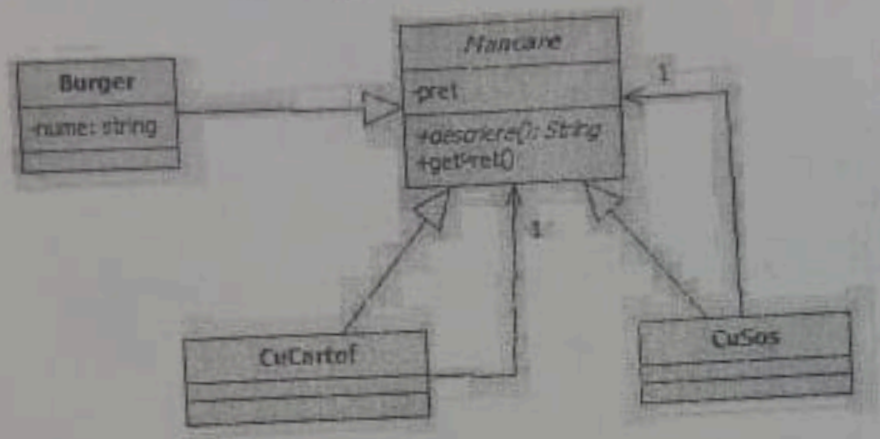
2 Indicați rezultatul execuției pentru următoarele programe c++. Daca sunt erori indicați locul unde apare eroarea si motivul.

```
//2 a  (1p)
#include <vector>
#include <iostream>
class A {
public:
	virtual void print() = 0;
};
class B : public A {
public:
	virtual void print() {
		std::cout << "printB";
	}
};
class C : public B {
public:
	virtual void print() {
		std::cout << "printC";
	}
};
int main() {
	std::vector<A> v;
	B b; C c;
	v.push_back(b);
	v.push_back(c);
	for (auto e : v) { e.print(); }
	return 0;
}
```

```
//2 b  (0.5p)
void f(bool b) {
	std::cout << "1";
	if (b) {
		throw
std::exception("Error");
	}
	std::cout << "3";
}
int main() {
	try {
		f(false);
		f(true);
		f(false);
	}
	catch (std::exception& ex) {
		std::cout << "4";
	}
	return 0;
}
```

1 3 1 4

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



- Clasa abstracta **Mancare** are o metoda pur virtuala descriere()
- **CuCartof** si **CuSos** conțin o mâncare si metoda descriere() adaugă textul "cu cartof" respectiv "cu sos" la descrierea mâncării conținute. Prețul creste cu 3 RON pentru cartofi, mâncarea cu sos costa in plus 2 RON.
- Clasa **Burger** reprezintă o un hamburger fără cartof si fără sos, metoda descriere() returnează denumirea hamburgerului.
- Scrieți o funcție C++ care returnează o lista de mâncăruri: un burger BigMac, un burger BigMac cu cartof si sos, un burger Zinger cu cartof si un burger Zinger cu sos (alegeți voi prețul de baza pentru fiecare mâncare).

în programul principal se creează o lista de mâncăruri (folosind funcția descrisa mai sus), apoi tripartiți descrierea si prețul pentru fiecare in ordinea descrescătoare a preturilor. 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 Carnet generala astfel încât următoarea secvență C++ sa fie corecta sintactic si sa efectueze ceea ce indica comentariile. (2p)

```
void anscolar() {
	Carnet<int> cat;
	cat.add("SDA", 9);//adauga nota pentru o materie
	cat.add("OOP", 7).add("FP", 10);
	cout << cat["OOP"]; //tipareste nota de la materia data (7 la OOP);
	//removeLast() sterge ultima nota adaugata in carnet
	cat.removeLast().removeLast();//sterge nota de la FP si OOP
	try{
		//se arunca exceptie daca nu exista nota pentru materia ceruta
		cout << cat["OOP"];
	}catch (std::exception& ex) {
		cout << "Nu exista nota pentru OOP";
	}
}
```