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

```
//2 a (1p)
                                        //2 b (0.5p)
#include <iostream>
                                        #include <iostream>
using namespace std;
                                        using namespace std;
int except(bool thrEx) {
                                        class A {
     if (thrEx) {
                                        public:
           throw 2;
                                          A() {cout << "A" << endl;}
                                          ~A() {cout << "~A" << endl;}
     return 3;
                                          void print() {
                                             cout << "print" << endl;</pre>
                                          }
int main() {
                                        };
     try {
           cout << except(1 < 1); 3
                                        void f() {
           cout << except(true); 2</pre>
                                              A a[2];
           cout << except(false);</pre>
                                              a[1].print();
     } catch (int ex) {
           cout << ex;
                                        int main() {
                                              f();
     cout << 4;
                                             return 0;
     return 0;
                                        }
```

Page 2 of 13

```
1) Specificați și testați funcția: (1.5p)
```

```
std::pair<int, int> f(std::vector<int> 1) {
    if (l.size()<2) throw std::exception{};
    std::pair<int, int> rez{-1,-1};
    for (auto el:1) {
        if (el < rez.second) continue;
        if (rez.first < el) {
            rez.second = rez.first;
            rez.first = el;
        }else{
            rez.second=el;
        }
    }
    return rez;</pre>
```

2 Indicați rezultatul execuției pentru următoarele programe c++. Daca sunt erori indicați

```
//2 a (1p)
                                       //2 b (0.5p)
#include <iostream>
                                       #include <iostream>
#include <vector>
                                       using namespace std;
struct A {
                                       class A {
     A() {std::cout << "A";}
                                             int x;
     virtual void print() {
                                       public:
           std::cout << "A";
                                             A(int x) : x\{x\} \{\}
                                             void print(){cout<< x <<endl;}</pre>
};
                                       };
struct B : public A{
                                       A f(A a) {
     B() { std::cout << "B"; }
                                             a.print();
     void print() override {
                                             a = A\{ 10 \};
           std::cout << "B";
                                             a.print();
     }
                                             return a;
};
int main() {
                                       int main() {
     std::vector<A> v;
                                             A a{ 4 };
     v.push back(A{});
                                             a.print();
     v.push_back(B{});
                                             f(a);
     for (auto& el : v) el.print();
                                             a.print();
                                       }
     return 0;
}
```

Page 3 of 13

```
1 Specificați si testați funcția: (1.5p)
bool f(int a) {
   if (a <= 0)
        throw std::exception("Illegal argument");
   int d = 2;
   while (d<a && a % d>0) d++;
   return d>=a;
}
```

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

```
//2 a (1p)
                                        //2 b (0.5p)
#include <iostream>
                                        #include <iostream>
using namespace std;
                                        #include <vector>
class A {
                                        using namespace std;
public:
                                        int main() {
A(){cout << "A()" << endl;}
                                             vector<int> v;
void print() {cout << "printA" <<</pre>
                                             v.push_back(5);
endl;}
                                             v.push back(7);
};
                                             v[0] = 6;
class B: public A {
                                             v.push_back(8);
                                             auto it = v.begin();
public:
B(){cout << "B()" << endl;}
                                             it = it + 1;
void print() {cout << "printB" <<</pre>
                                             while (it != v.end()) {
endl;}
                                                   cout << *it << endl;
};
                                                   it++;
int main() {
                                             }
     A* o1 = new A();
                                          return 0;
     B* o2 = new B();
     o1->print();
     o2->print();
     delete o1;delete o2;
  return 0;
```

Page 4 of 13

```
1) Specificați si testați funcția: (1.5p)
bool f(int a) {
    if (a <= 1)
        throw "Illegal argument";
    int aux = 0;
    for (int i = 2; i < a; i++) {
        if (a % i == 0) {
            aux++;
        }
    }
    return aux == 0;
}</pre>
```

2 Indicați rezultatul execuției pentru următoarele programe c++. Daca sunt erori indicați

```
//2 a (1p)
                                        //2 b (0.5p)
#include <vector>
                                        #include <iostream>
#include <iostream>
                                        using namespace std;
using namespace std;
                                        class A {
class A {
                                        public:
                                         A() {cout << "A" << endl;}
public:
     virtual void f() = 0;
                                         ~A() {cout << "~A" << endl;
                                         void print() {cout << "print" <<</pre>
class B:public A{
                                        endl;}
public:
                                        };
                                        void f() {
     void f() override {
           cout << "f din B";</pre>
                                              A a[2];
                                              a[0].print();
};
class C :public B {
                                        int main() {
public:
                                             f();
                                             return 0;
     void f() override {
           cout << "f din C";</pre>
                                        }
     }
};
int main() {
     vector<A> v;
     B b;
     v.push_back(b);
     C c;
     v.push_back(c);
     for (auto e : v) { e.f(); }
     return 0;
}
```

```
1 Specificați și testați funcția: (1.5p)
using namespace std;
#include <vector>
#include <string>
#include <algorithm>
#include <map>
vector<int> f(vector<int> 1) {
      if (l.size() == 0)
            throw exception("Illegal argument");
      map<int, int> c;
      for (auto e : 1) {
            c[e]++;
      sort(1.begin(), 1.end(), [&](int a, int b) {
            return c[a] > c[b]; });
      return 1;
}
```

locul unde apare eroarea si motivul.

```
//2 a (1p)
                                          //2 b (0.5p)
#include <vector>
                                          #include <iostream>
#include <iostream>
                                          using namespace std;
class A {
                                          class A {
public:
                                          public:
     A() {
                                            A() {cout << "A" << endl;}
            std::cout << "A";
                                            ~A() {cout << "~A" << endl;}
                                            void print() {
      virtual void print() {
                                              cout << "print" << endl;</pre>
           std::cout << "printA";</pre>
                                            }
                                          };
                                          void f() {
class B : public A {
public:
                                                A a[2];
     B() {
                                                a[1].print();
            std::cout << "B";
                                          int main() {
      virtual void print() {
                                                f();
           std::cout << "printB";
                                               return 0;
      }
                                          }
int main() {
     std::vector<A> v;
     A a;
     B b;
     v.push_back(a);
     v.push_back(b);
     for (auto e : v) {e.print();}
     return 0;
```

3 Scrieti codul C++ ce corespunde diagramei de clase UML. (4p)

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

vector<int> f(int a) {
    if (a < 0)
        throw std::exception("Illegal argument");
    vector<int> rez;
```

for (int i = 1; i <= a; i++) {
 if (a % i == 0) {
 rez.push\_back(i);
 }
}
return rez;
}</pre>

2 Indicați rezultatul execuției pentru următoarele programe c++. Daca sunt erori indicați

```
//2 a (1p)
                                         //2 b (0.5p)
#include <iostream>
                                         #include <iostream>
using namespace std;
                                         using namespace std;
int except(int v) {
                                         class A {
     if (v < 0) {throw 1; }
                                               int x;
     else if (v > 0){
                                         public:
                                               A(int \times) : x\{ \times \} \{ \}
        throw std::exception ("A");
                                                void print(){cout<< x <<",";}</pre>
     return 0;
                                         };
                                         A f(A a) {
int main(){
                                               a.print();
     try {
                                               a = A\{ 7 \};
         cout << except(1 < 1);</pre>
                                               a.print();
         cout << except(-5);</pre>
                                               return a;
         cout << except(5);</pre>
     }catch (std::exception& e) {
                                         int main() {
           cout << "A";
                                               A a{ 5 };
     }catch (int x) {
                                               a.print();
           cout << "B";
                                               f(a);
                                               a.print();
     cout << "C";
                                         }
    return 0;
```

Page 7 of 13

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

2 Indicați rezultatul execuției pentru următoarele programe c++. Daca sunt erori indicați

```
//2 a (1p)
                                         //2 b (0.5p)
#include <iostream>
                                         #include <iostream>
using namespace std;
                                         using namespace std;
int except(bool thrEx) {
                                         class A {
     if (thrEx) {
                                         public:
                                           A() {cout << "A" << endl;}
           throw 2;
                                           ~A() {cout << "~A" << endl;}
     return 3;
                                           void print() {
                                              cout << "print" << endl;</pre>
                                           }
int main() {
                                         };
     try {
           cout << except(1 < 1);</pre>
                                         void f() {
           cout << except(true);</pre>
                                              A a[2];
           cout << except(false);</pre>
                                              a[1].print();
     } catch (int ex) {
           cout << ex;
                                         int main() {
                                              f();
     cout << 4;
                                              return 0;
     return 0;
                                         }
```

Page 8 of 13

1 Specificați si testați funcția: (1.5p)

```
std::pair<int, int> f(std::vector<int> 1) {
    if (l.size()<2) throw std::exception{};
    std::pair<int, int> rez{-1,-1};
    for (auto el:1) {
        if (el < rez.second) continue;
        if (rez.first < el) {
            rez.second = rez.first;
            rez.first = el;
        }else{
            rez.second=el;
        }
    }
    return rez;
}</pre>
```

2 Indicați rezultatul execuției pentru următoarele programe c++. Daca sunt erori indicați

```
//2 a (1p)
                                        //2 b (0.5p)
#include <iostream>
                                        #include <iostream>
#include <vector>
                                        using namespace std;
struct A {
                                        class A {
     A() {std::cout << "A";}
                                             int x;
     virtual void print() {
                                        public:
           std::cout << "A";
                                             A(int x) : x\{x\} \{\}
                                             void print(){cout<< x <<endl;}</pre>
};
                                        };
struct B : public A{
                                        A f(A a) {
     B() { std::cout << "B"; }
                                             a.print();
                                             a = A\{ 10 \};
     void print() override {
           std::cout << "B";
                                             a.print();
     }
                                             return a;
};
int main() {
                                       int main() {
     std::vector<A> v;
                                             A a{ 4 };
     v.push back(A{});
                                             a.print();
     v.push_back(B{});
                                             f(a);
     for (auto& el : v) el.print();
                                             a.print();
                                        }
     return 0;
}
```

Page 9 of 13

```
1 Specificați si testați funcția: (1.5p)
bool f(int a) {
    if (a <= 0)
        throw std::exception("Illegal argument");
    int d = 2;
    while (d<a && a % d>0) d++;
    return d>=a;
}
```

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

```
//2 a (1p)
                                        //2 b (0.5p)
#include <iostream>
                                        #include <iostream>
using namespace std;
                                        #include <vector>
class A {
                                        using namespace std;
public:
                                        int main() {
A(){cout << "A()" << endl;}
                                             vector<int> v;
void print() {cout << "printA" <<</pre>
                                             v.push_back(5);
endl;}
                                             v.push back(7);
};
                                             v[0] = 6;
class B: public A {
                                             v.push_back(8);
                                             auto it = v.begin();
public:
B(){cout << "B()" << endl;}
                                             it = it + 1;
void print() {cout << "printB" <<</pre>
                                             while (it != v.end()) {
endl;}
                                                   cout << *it << endl;
};
                                                   it++;
int main() {
                                             }
     A* o1 = new A();
                                          return 0;
     B* o2 = new B();
     o1->print();
     o2->print();
     delete o1;delete o2;
  return 0;
```

```
//2 a (1p)
                                        //2 b (0.5p)
#include <vector>
                                        #include <iostream>
#include <iostream>
                                        using namespace std;
using namespace std;
                                        class A {
class A {
                                        public:
                                         A() {cout << "A" << endl;}
public:
     virtual void f() = 0;
                                         ~A() {cout << "~A" << endl;
                                         void print() {cout << "print" <<</pre>
class B:public A{
                                        endl;}
public:
                                        };
     void f() override {
                                        void f() {
           cout << "f din B";</pre>
                                              A a[2];
                                              a[0].print();
};
class C :public B {
                                        int main() {
public:
                                             f();
                                             return 0;
     void f() override {
           cout << "f din C";</pre>
                                        }
     }
};
int main() {
     vector<A> v;
     B b;
     v.push_back(b);
     C c;
     v.push_back(c);
     for (auto e : v) { e.f(); }
     return 0;
}
```

```
1 Specificați și testați funcția: (1.5p)
using namespace std;
#include <vector>
#include <string>
#include <algorithm>
#include <map>
vector<int> f(vector<int> 1) {
      if (1.size() == 0)
             throw exception("Illegal argument");
      map<int, int> c;
      for (auto e : 1) {
            c[e]++;
       sort(1.begin(), 1.end(), [&](int a, int b) {
            return c[a] > c[b]; });
      return 1;
}
```

locul unde apare eroarea si motivul.

```
//2 a (1p)
                                          //2 b (0.5p)
#include <vector>
                                          #include <iostream>
#include <iostream>
                                          using namespace std;
class A {
                                          class A {
public:
                                          public:
     A() {
                                            A() {cout << "A" << endl;}
            std::cout << "A";
                                            ~A() {cout << "~A" << endl;}
                                            void print() {
      virtual void print() {
                                              cout << "print" << endl;</pre>
           std::cout << "printA";</pre>
                                            }
                                          };
                                          void f() {
class B : public A {
public:
                                                A a[2];
     B() {
                                                a[1].print();
            std::cout << "B";
                                          int main() {
      virtual void print() {
                                                f();
           std::cout << "printB";
                                               return 0;
      }
int main() {
     std::vector<A> v;
     A a;
     B b;
     v.push_back(a);
     v.push_back(b);
     for (auto e : v) {e.print();}
     return 0;
```

3 Scrieți codul C++ ce corespunde diagramei de clase UML. (4p)

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

```
vector<int> f(int a) {
    if (a < 0)
        throw std::exception("Illegal argument");
    vector<int> rez;
    for (int i = 1; i <= a; i++) {
        if (a % i == 0) {
            rez.push_back(i);
        }
    }
    return rez;
}</pre>
```

```
//2 a (1p)
                                         //2 b (0.5p)
#include <iostream>
                                         #include <iostream>
using namespace std;
                                         using namespace std;
int except(int v) {
                                         class A {
     if (v < 0) {throw 1; }
                                               int x;
     else if (v > 0){
                                         public:
                                               A(int \times) : x\{ \times \} \{ \}
        throw std::exception ("A");
                                                void print(){cout<< x <<",";}</pre>
     return 0;
                                         };
                                         A f(A a) {
int main(){
                                               a.print();
                                               a = A\{ 7 \};
     try {
         cout << except(1 < 1);</pre>
                                               a.print();
         cout << except(-5);</pre>
                                               return a;
         cout << except(5);</pre>
     }catch (std::exception& e) {
                                         int main() {
           cout << "A";
                                               A a{ 5 };
     }catch (int x) {
                                               a.print();
           cout << "B";
                                               f(a);
                                               a.print();
     cout << "C";
                                         }
    return 0;
```

```
1 Specificați si testați funcția: (1.5p)
 int f(int x) {
     if (x <= 0)
           throw std::exception("Invalid argument!");
     int rez = 0;
     while (x)
      {
           rez = rez * 10 + x % 10;
           \times /= 10;
     return rez;
2 Indicați rezultatul execuției pentru următoarele programe c++. Daca sunt erori indicați
locul unde apare eroarea si motivul.
//2 a (1p)
                                          //2 b (0.5p)
#include <iostream>
                                          #include <iostream>
using namespace std;
                                          using namespace std;
int except(bool thrEx) {
                                          class A {
                                          public:
      if (thrEx) {
                                            A() {cout << "A" << endl;}
            throw 2;
                                            ~A() {cout << "~A" << endl;}
      return 3;
                                            void print() {
                                                cout << "print" << endl;</pre>
                                            }
int main() {
                                          };
      try {
            cout << except(1 < 1);</pre>
                                          void f() {
            cout << except(true);</pre>
                                                A a[2];
            cout << except(false);</pre>
                                                a[1].print();
      } catch (int ex) {
            cout << ex;
                                          int main() {
                                                f();
      cout << 4;
                                                return 0;
      return 0;
                                          }
```