EXERCISES UNIT 2

1. Implement a *copy constructor* and a *move constructor* for the following classes:

```
class Pila
{
  int * v;
  int cima;
  unsigned max_tam;
public:
  Pila(unsigned tam)
  {
    v = new int[tam];
    max_tam = tam;
    cima = -1;
  }
};

class Vector2
{
  int * v[MAX];
  unsigned tam;
public:
  Vector2(unsigned t)
  {
    tam = t;
    for(unsigned i = 0; i < MAX; i++)
      v[i] = new int[tam];
  }
};
```

2. The following loop is not working correctly; why?

```
vector<int> v = {1,2,3,4};

for(auto i: v)
  cout << v[i] << endl;
```

3. Display the following vector using a range-based for loop.

```
vector<list<double>> v;
```

4. Write a class constructor with an initializer list parameter for the class *Pila* of exercise **1** (ex. Pila p({2,3,4,5})).

5. Given the class *Pila* of exercise **1**, add the appropriate code to avoid the possibility of calling the default copy constructor.

6. Create a strict enumerated type (enum class) for representing days of the week. The underlying type should be an unsigned char. Monday will be the first day of the week and it should have value 1. Write an example of use of the newly created type. There should be at least one *if* statement.

7. Define the necessary *user defined literals* for the following expression to function correctly.

```
// time contains number of seconds.
unsigned time = 1_h + 15_m + 30_s
```

8. Complete the following exercises with lists:

```
list<int> l1 = {1,2,3};
list<int> l2 = {-2,-1};
```

a) Insert the value 0 at the beginning of list l1.

b) Insert the value 4 at the end of list l1.

c) Insert the list l2 at the beginning of list l1.

```
forward_list<int> l3 = {1,2,3};
forward_list<int> l4 = {-2,-1};
```

d) Insert the value 0 at the beginning of list l3.

e) Insert the value 4 at the end of list l3.

f) Insert the list l4 at the beginning of list l3.

9. Write the function *suma* for the following program. The function *suma* must sum all the elements of the array.

```
int main()
{
    array<int,5> a = {2,4,6,8,10};
    int s = suma(a);
    cout << s << endl;
}
```

10. Write a function that as input parameter has a *vector* (of the STL) of strings and as output parameter has an *unordered_map* that contains how many times appears each string in the vector.

11. Complete the following code that multiplies by 3 all the elements of the container.

```
vector<int> c = {1,2,3};
auto mult3 = ??
for_each(c.begin(), c.end(), mult3);
```