

```

1: #include <iostream>
2: #include <vector>
3: using namespace std;
4:
5: // Topic 10 >> Containers, e.g., vector
6: class Point
7: {
8: private:
9:     int x, y;
10:
11: public:
12:     Point(int _x = 0, int _y = 0) : x(_x), y(_y) {}
13:     void set(int _x, int _y)
14:     {
15:         x = _x;
16:         y = _y;
17:     }
18:     int getX() const { return x; }
19:     int getY() const { return y; }
20: };
21:
22: int menu()
23: {
24:     cout << endl;
25:     cout << "1. Add point" << endl;
26:     cout << "2. Remove last point" << endl;
27:     cout << "3. Edit point" << endl;
28:     cout << "4. Print list" << endl;
29:     cout << "5. Exit" << endl;
30:     cout << endl;
31:     cout << "Choose an operation from 1 to 5 => ";
32:
33:     int choice;
34:     cin >> choice;
35:
36:     cout << endl;
37:
38:     return choice;
39: }
40:
41: int main()
42: {
43:     vector<Point> list;
44:     list.push_back(Point(1, 2));
45:     list.push_back(Point(31, 2));
46:     list.push_back(Point(13, 32));
47:
48:     int c = menu();
49:     int x, y;
50:     int index;
51:
52:     while (c != 5)
53:     {
54:
55:         switch (c)
56:         {
57:             case 1: // add a new item at the back of the list
58:                 cout << "Enter x and y => ";
59:                 cin >> x >> y;
60:

```

```

61:         list.push_back(Point(x, y));
62:         cout << "Number of items in the list now is " << list.size() << endl;
63:         break;
64:
65:     case 2: // Remove the Last item from the List
66:         list.pop_back();
67:         cout << "Number of items in the list now is " << list.size() << endl;
68:         break;
69:
70:     case 3: // Edit the item at index
71:         cout << "Enter item's index to edit => ";
72:         cin >> index;
73:         cout << endl;
74:
75:         cout << "Current x and y: " << list[index].getX() << " , " <<
list[index].getY() << endl;
76:         cout << "Enter new x and y => ";
77:         cin >> x >> y;
78:         list[index].set(x, y);
79:         break;
80:
81:     case 4: // Print all items in the list
82:         // for (int i=0; i<list.size(); i++)
83:         //     cout << (i+1) << ".\t x=" << list[i].getX() << ", y=" <<
list[i].getY() << endl;
84:
85:         vector<Point>::iterator i;
86:
87:         for (i = list.begin(); i != list.end(); i++)
88:             cout << ".\t x=" << i->getX() << ", y=" << i->getY() << endl;
89:
90:         cout << endl;
91:
92:         break;
93:
94:     } // switch
95:
96:     c = menu();
97: } // while
98:
99: system("pause");
100: return 0;
101: }

```

```

1: #include <iostream>
2: #include <string>
3: #include <map>
4: using namespace std;
5:
6: // Topic 10 >> Containers, e.g., map
7:
8: string dayNumToName(int d)
9: {
10:     const string DAYS[7] = {"Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"};
11:     return DAYS[d - 1];
12: }
13:
14: int dayNameToNum(string n)
15: {
16:     map<string, int> DAYS = {{"Mon",1}, {"Tue",2}, {"Wed",3}, {"Thu",4}, {"Fri",5},
17:     {"Sat",6}, {"Sun",7}};
18:     DAYS["Monday"] = 1;
19:
20:     return DAYS[n];
21: }
22:
23: int main()
24: {
25:
26:     cout << dayNumToName(1) << endl;
27:     cout << dayNameToNum("Mon") << endl;
28:     cout << dayNameToNum("Tue") << endl;
29:     cout << dayNameToNum("Monday") << endl;
30:     cout << dayNameToNum("monday") << endl;
31:
32:     system("pause");
33:     return 0;
34: }

```

```

1: #include <iostream>
2: #include <vector>
3: #include <map>
4: using namespace std;
5:
6: // Topic 10 >> Iterators: Sample code
7:
8: class Point
9: {
10: private:
11:     int x, y;
12:
13: public:
14:     Point(int _x = 0, int _y = 0) : x(_x), y(_y) {}
15:     int getX() const { return x; }
16:     int getY() const { return y; }
17: };
18:
19: int main()
20: {
21:     vector<int> numbers;
22:     numbers.push_back(1);
23:     numbers.push_back(2);
24:     numbers.push_back(3);
25:
26:     //without iterator
27:     for (int i = 0; i < numbers.size(); i++)
28:         cout << numbers[i] << "\t";
29:     cout << endl;
30:
31:     //with iterator
32:     vector<int>::iterator i;
33:     for (i = numbers.begin(); i != numbers.end(); i++)
34:         cout << *i << "\t";
35:     cout << endl;
36:
37:     vector<Point> points;
38:     points.push_back(Point(1, 2));
39:     points.push_back(Point(11, 22));
40:     points.push_back(Point(41, 32));
41:
42:     // work with objects without iterator
43:     for (int i = 0; i < points.size(); i++)
44:         cout << "x=" << points[i].getX() << "\ty=" << points[i].getY() << endl;
45:
46:     // work with objects with iterator
47:     vector<Point>::iterator p;
48:     for (p = points.begin(); p != points.end(); p++)
49:         cout << "x=" << (*p).getX() << "\ty=" << p->getY() << endl;
50:
51:     map<string, int> days;
52:     days["mon"] = 1;
53:     days["tue"] = 2;
54:
55:     map<int, string> hari;
56:     hari[1] = "isnin";
57:     hari[5] = "jumaat";
58:
59:     map<string, int>::iterator d;
60:     for (d = days.begin(); d != days.end(); d++)

```

```
61:         cout << d->first << "\t" << d->second << endl;
62:
63:     map<int, string>::iterator h;
64:     for (h = hari.begin(); h != hari.end(); h++)
65:         cout << h->first << "\t" << h->second << endl;
66:
67:     system("pause");
68:     return 0;
69: }
```