```
=== 演習1 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex11-1.cpp
$ ./a.out
plan A: 3270
plan B: -50


--- ex11-1.cpp ---
// std::accumulate
#include <functional>
#include <iostream>
#include <numeric>
#include <vector>
#include <map>

int main()
{
   int total {10000};
   std::map<std::string,std::vector<int>> plans
   {{"plan A", {1500, 2000, 1800, 780, 650}},
    {"plan B", {5500, 1000, 320, 1580, 1650}}};
   for (auto& [n,p]: plans) {
      std::cout << n <<": "
      << std::accumulate(p.begin(), p.end(), total, std::minus<int>{})
      << "\n";
   }
}
```

```
=== 演習2 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex11-2.cpp
$ ./a.out
5
1


--- ex11-2.cpp ---
// lambda expression
#include <algorithm>
#include <iostream>
#include <vector>
int main()
{
   std::vector a{3,2,4,6,8,1,2,3,4}, s{1,2,3};
   auto it { std::search(a.begin(), a.end(),
                         s.begin(), s.end()) };
   if (it != a.end())
      std::cout << it-a.begin() <<"\n"; // 5

   it = std::search(a.begin(), a.end(),
                    s.begin(), s.end(),
            [](int x, int y) { return x % y == 0; });
   if (it != a.end())
      std::cout << it-a.begin() <<"\n"; // 1
}


--- ex11-2org.cpp ---
// ex2: original program
#include <algorithm>
#include <iostream>
#include <vector>
bool pred(int x, int y) { return x % y == 0; }
int main()
{
   std::vector a{3,2,4,6,8,1,2,3,4}, s{1,2,3};
   auto it { std::search(a.begin(), a.end(),
                         s.begin(), s.end()) };
   if (it != a.end())
      std::cout << it-a.begin() <<"\n"; // 5

   it = std::search(a.begin(), a.end(),
                    s.begin(), s.end(), pred);
   if (it != a.end())
      std::cout << it-a.begin() <<"\n"; // 1
}
```

```
=== 演習3 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex11-3.cpp
$ ./a.out
25333 25333
$ ./a.out 20
722666 722666
$ ./a.out 30
5273999 5273999


--- ex11-3.cpp ---
// sum of series of n^4
#include <algorithm>
#include <iostream>
#include <numeric>
int main(int argc, char *argv[])
{
   int n { argc>1 ? std::stoi(argv[1]):10 };
   std::vector<int> x(n);
   auto b{x.begin()}, e{x.end()};
   std::iota(b, e, 1);
   std::cout <<
   std::accumulate(b, e, 0,
      [](int i, int e){ return i + e*e*e*e; })
   <<" "<< n*(n+1)*(2*n+1)*(3*n*n+3*n-1)/30<<"\n";
}
```

```
=== 演習4 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex11-4.cpp
$ ./a.out
[abc:2] [ijk:1] [opq:4] [xyz:3]
[ijk:1] [abc:2] [xyz:3] [opq:4]


--- print.hpp ---
#include <iostream>
template<typename T>
void print(const T& c)
{
   for (auto& e: c)
      std::cout << e <<" ";
   std::cout <<"\n";
}

template<typename Itr>
void print(Itr b, Itr e)
{
   for ( ; b != e; ++b)
      std::cout << *b <<" ";
   std::cout <<"\n";
}


--- sales.hpp ---
#include <iostream>
class Sales {
   std::string item;
   int num;
public:
   Sales() = default;
   Sales(std::string i, int n):item(i),num(n){}
   auto geti() const { return item; }
   auto getn() const { return num; }
   friend auto&
   operator<<(std::ostream& o, const Sales& s) {
      return o <<"["<< s.item <<":"<< s.num <<"]";
   }
};


--- ex11-4.cpp ---
// std::sort
#include <algorithm>
#include <vector>
#include "print.hpp"
#include "sales.hpp"
int main()
{
   std::vector<Sales> v {{"xyz",3},{"abc",2},
                         {"ijk",1},{"opq",4}};
   auto b{v.begin()}, e{v.end()};
   std::sort(b, e, [](const auto& a, const auto& b) { return a.geti() < b.geti(); });
   print(v);
   std::sort(b, e, [](const auto& a, const auto& b) { return a.getn()< b.getn(); });
   print(v);
}
```

```
=== 演習5 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex11-5.cpp
$ ./a.out
total = 10


--- sales.hpp ---
#include <iostream>
class Sales {
   std::string item;
   int num;
public:
   Sales() = default;
   Sales(std::string i, int n):item(i),num(n){}
   auto geti() const { return item; }
   auto getn() const { return num; }
   friend auto&
   operator<<(std::ostream& o, const Sales& s) {
      return o <<"["<< s.item <<":"<< s.num <<"]";
   }
};


--- ex11-5.cpp ---
// std::accumulate
#include <numeric>
#include <vector>
#include "sales.hpp"
int main()
{
   std::vector<Sales> v {{"xyz",3},{"abc",2},
                         {"ijk",1},{"opq",4}};
   auto b{v.begin()}, e{v.end()};
   std::cout <<"total = "
   << std::accumulate(b, e, 0,
      [](int i, const Sales& s) { return i+s.getn(); })
   <<"\n"; // total = 10
}
```

```
=== 演習6 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex11-6.cpp
$ ./a.out
 : not found
$ ./a.out abc
[abc:2]
$ ./a.out lmn
lmn : not found
$ ./a.out xyz
[xyz:3]


--- sales.hpp ---
#include <iostream>
class Sales {
   std::string item;
   int num;
public:
   Sales() = default;
   Sales(std::string i, int n):item(i),num(n){}
   auto geti() const { return item; }
   auto getn() const { return num; }
   friend auto&
   operator<<(std::ostream& o, const Sales& s) {
      return o <<"["<< s.item <<":"<< s.num <<"]";
   }
};


--- ex11-6.cpp ---
// std::find_if
#include <algorithm>
#include <vector>
#include "sales.hpp"
int main(int argc, char *argv[])
{
   std::string item { argc>1 ? argv[1]:"" };
   std::vector<Sales> v {{"xyz",3},{"abc",2},
                         {"ijk",1},{"opq",4}};
   auto it { std::find_if(v.begin(), v.end(),
             [item](const Sales& a)
             { return item == a.geti(); }) };
   if (it != v.end())
      std::cout << *it <<"\n";
   else
      std::cout << item << " : not found\n";
}
```