```
=== 演習1 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex14-1a.cpp
$ ./a.out
ctor(default): 0x16e4eb0
ctor(default): 0x16e4eb4
ctor(default): 0x16e4eb8
ctor(default): 0x16e4ebc
ctor(default): 0x16e4ec0
ctor(default): 0x16e4ec4
ctor(default): 0x16e4ec8
ctor(default): 0x16e4ecc
ctor(default): 0x16e4ed0
ctor(default): 0x16e4ed4
sum = ctor(copy):    0x16e5ef0
ctor(copy):    0x16e5ef4
ctor(copy):    0x16e5ef8
ctor(copy):    0x16e5efc
ctor(copy):    0x16e5f00
ctor(copy):    0x16e5f04
ctor(copy):    0x16e5f08
ctor(copy):    0x16e5f0c
ctor(copy):    0x16e5f10
ctor(copy):    0x16e5f14
ctor(copy):    0x7fff3344f59c
dtor:          0x7fff3344f59c
ctor(copy):    0x7fff3344f59c
dtor:          0x7fff3344f59c
ctor(copy):    0x7fff3344f59c
dtor:          0x7fff3344f59c
ctor(copy):    0x7fff3344f59c
dtor:          0x7fff3344f59c
ctor(copy):    0x7fff3344f59c
dtor:          0x7fff3344f59c
ctor(copy):    0x7fff3344f59c
dtor:          0x7fff3344f59c
ctor(copy):    0x7fff3344f59c
dtor:          0x7fff3344f59c
ctor(copy):    0x7fff3344f59c
dtor:          0x7fff3344f59c
ctor(copy):    0x7fff3344f59c
dtor:          0x7fff3344f59c
ctor(copy):    0x7fff3344f59c
dtor:          0x7fff3344f59c
10
dtor:          0x16e5ef0
dtor:          0x16e5ef4
dtor:          0x16e5ef8
dtor:          0x16e5efc
dtor:          0x16e5f00
dtor:          0x16e5f04
dtor:          0x16e5f08
dtor:          0x16e5f0c
dtor:          0x16e5f10
dtor:          0x16e5f14
dtor:          0x16e4eb0
dtor:          0x16e4eb4
dtor:          0x16e4eb8
dtor:          0x16e4ebc
dtor:          0x16e4ec0
dtor:          0x16e4ec4
dtor:          0x16e4ec8
dtor:          0x16e4ecc
dtor:          0x16e4ed0
dtor:          0x16e4ed4
$ ./a.out | grep ctor
ctor(default): 0x116aeb0
ctor(default): 0x116aeb4
ctor(default): 0x116aeb8
ctor(default): 0x116aebc
ctor(default): 0x116aec0
ctor(default): 0x116aec4
ctor(default): 0x116aec8
ctor(default): 0x116aecc
ctor(default): 0x116aed0
ctor(default): 0x116aed4
sum = ctor(copy):    0x116bef0
ctor(copy):    0x116bef4
ctor(copy):    0x116bef8
ctor(copy):    0x116befc
ctor(copy):    0x116bf00
ctor(copy):    0x116bf04
ctor(copy):    0x116bf08
ctor(copy):    0x116bf0c
ctor(copy):    0x116bf10
ctor(copy):    0x116bf14
ctor(copy):    0x7ffe305c1fac
ctor(copy):    0x7ffe305c1fac
ctor(copy):    0x7ffe305c1fac
ctor(copy):    0x7ffe305c1fac
ctor(copy):    0x7ffe305c1fac
ctor(copy):    0x7ffe305c1fac
ctor(copy):    0x7ffe305c1fac
ctor(copy):    0x7ffe305c1fac
ctor(copy):    0x7ffe305c1fac
ctor(copy):    0x7ffe305c1fac
$ ./a.out | grep ctor | wc -l
30
$ ./a.out | grep dtor | wc -l
30
$ g++ -std=c++17 ex14-1b.cpp
ctor(default): 0xca8eb0
ctor(default): 0xca8eb4
ctor(default): 0xca8eb8
ctor(default): 0xca8ebc
ctor(default): 0xca8ec0
ctor(default): 0xca8ec4
ctor(default): 0xca8ec8
ctor(default): 0xca8ecc
ctor(default): 0xca8ed0
ctor(default): 0xca8ed4
sum = 10
dtor:          0xca8eb0
dtor:          0xca8eb4
dtor:          0xca8eb8
dtor:          0xca8ebc
dtor:          0xca8ec0
dtor:          0xca8ec4
dtor:          0xca8ec8
dtor:          0xca8ecc
dtor:          0xca8ed0
dtor:          0xca8ed4
$ ./a.out | grep ctor | wc -l
10
$ ./a.out | grep dtor | wc -l
10
```

```
--- ex14.hpp ---
// ex14.hpp
#include <iostream>
class T {
   int a;
   void log(std::string m) {
      std::cout << m <<" "<< &a <<"\n";
   }
public:
   T(int b)      :a{b}  { log("ctor(int):     "); }
   T()           :a{1}  { log("ctor(default):"); }
   T(const T& x):a{x.a}{ log("ctor(copy):    "); }
   ~T()                 { log("dtor:          "); }
   int get()     const { return a; }
   void set(int x)      { a = x;     }
};


--- ex14-1a.cpp ---
// ex14-1a.cpp
#include <vector>
#include "ex14.hpp"
using std::vector;
int sum(vector<T> x) { // あえて値渡し
   int s {0};
   for (auto e:x) s += e.get(); // コピー
   return s;
}

int main(){
   vector<T> a(10);
   std::cout <<"sum = "<< sum(a) <<"\n";
}



--- ex14-1b.cpp ---
// ex14-1b.cpp
#include <vector>
#include "ex14.hpp"
using std::vector;
int sum(vector<T>& x) { // リファレンス
   int s {0};
   for (auto& e:x) s += e.get(); // リファレンス
   return s;
}

int main() {
   vector<T> a(10);
   std::cout <<"sum = "<< sum(a) <<"\n";
}
```

```
=== 演習2 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex14-2a.cpp
$ ./a.out
ctor(int):     0x18eaeb0
1
ctor(int):     0x18eb2e8
ctor(int):     0x18eb2ec
ctor(default): 0x18eb2f0
ctor(default): 0x18eb2f4
ctor(default): 0x18eb2f8
3 2 1 1 1
$ g++ -std=c++17 ex14-2b.cpp
$ ./a.out
ctor(int):     0x22c2eb0
1
ctor(int):     0x22c32e8
ctor(int):     0x22c32ec
ctor(default): 0x22c32f0
ctor(default): 0x22c32f4
ctor(default): 0x22c32f8
3 2 1 1 1
dtor:          0x22c2eb0
dtor:          0x22c32f8
dtor:          0x22c32f4
dtor:          0x22c32f0
dtor:          0x22c32ec
dtor:          0x22c32e8


--- ex14.hpp ---
// ex14.hpp
#include <iostream>
class T {
   int a;
   void log(std::string m) {
      std::cout << m <<" "<< &a <<"\n";
   }
public:
   T(int b)      :a{b}  { log("ctor(int):     "); }
   T()           :a{1}  { log("ctor(default):"); }
   T(const T& x):a{x.a}{ log("ctor(copy):    "); }
   ~T()                 { log("dtor:          "); }
   int get()      const { return a; }
   void set(int x)      { a = x;    }
};


--- ex14-2a.cpp ---
// ex14-2.cpp
#include "ex14.hpp"
using std::cout;
int main() {
   T* p { new T{1} };
   cout << p->get() <<"\n";

   T* a { new T[5]{3,2} }; // 初期値指定
   for (int i = 0; i<5; i++)
      cout << a[i].get() <<" ";
   cout <<"\n";
   // どちらも解放を忘れている。
}



--- ex14-2b.cpp ---
// ex14-2.cpp
#include "ex14.hpp"
using std::cout;
int main() {
   T* p { new T{1} };
   cout << p->get() <<"\n";

   T* a { new T[5]{3,2} }; // 初期値指定
   for (int i = 0; i<5; i++)
      cout << a[i].get() <<" ";
   cout <<"\n";

   delete p;
   delete[] a;
}
```

```
=== 演習3 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex14-3.cpp
$ ./a.out
ctor(int):      0x2142eb0
1
ctor(default): 0x21432e8
ctor(default): 0x21432ec
ctor(default): 0x21432f0
ctor(default): 0x21432f4
ctor(default): 0x21432f8
3 2 1 1 1
dtor:           0x21432f8
dtor:           0x21432f4
dtor:           0x21432f0
dtor:           0x21432ec
dtor:           0x21432e8
dtor:           0x2142eb0


--- ex14.hpp ---
// ex14.hpp
#include <iostream>
class T {
   int a;
   void log(std::string m) {
      std::cout << m <<" "<< &a <<"\n";
   }
public:
   T(int b)      :a{b}  { log("ctor(int):     "); }
   T()           :a{1}  { log("ctor(default):"); }
   T(const T& x):a{x.a}{ log("ctor(copy):    "); }
   ~T()                 { log("dtor:          "); }
   int get()     const { return a; }
   void set(int x)      { a = x;    }
};


--- ex14-3.cpp ---
// ex14-3.cpp
#include <memory>
#include "ex14.hpp"
using std::cout;
int main() {
   auto p { std::make_unique<T>(1) };
   cout << p->get() <<"\n";

   auto a { std::make_unique<T[]>(5) };
   a[0].set(3);
   a[1].set(2);
   for (int i = 0; i<5; i++)
      cout << a[i].get() <<" ";
   cout <<"\n";
}
```

```
=== 演習4 解答例 ===========================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex14-4a.cpp
$ ./a.out
ctor(int):     0x7fff5ac9b238
ctor(copy):    0xf262c0
dtor:          0x7fff5ac9b238
ctor(int):     0x7fff5ac9b238
ctor(copy):    0xf262e4
ctor(copy):    0xf262e0
dtor:          0xf262c0
dtor:          0x7fff5ac9b238
ctor(int):     0x7fff5ac9b238
ctor(copy):    0xf262c8
ctor(copy):    0xf262c0
ctor(copy):    0xf262c4
dtor:          0xf262e0
dtor:          0xf262e4
dtor:          0x7fff5ac9b238
ctor(int):     0x7fff5ac9b238
ctor(copy):    0xf262cc
dtor:          0x7fff5ac9b238
ctor(int):     0x7fff5ac9b238
ctor(copy):    0xf26310
ctor(copy):    0xf26300
ctor(copy):    0xf26304
ctor(copy):    0xf26308
ctor(copy):    0xf2630c
dtor:          0xf262c0
dtor:          0xf262c4
dtor:          0xf262c8
dtor:          0xf262cc
dtor:          0x7fff5ac9b238
dtor:          0xf26300
dtor:          0xf26304
dtor:          0xf26308
dtor:          0xf2630c
dtor:          0xf26310

$ g++ -std=c++17 ex14-4b.cpp
$ ./a.out
ctor(default): 0x821eb0
ctor(default): 0x821eb4
ctor(default): 0x821eb8
ctor(default): 0x821ebc
ctor(default): 0x821ec0
dtor:          0x821eb0
dtor:          0x821eb4
dtor:          0x821eb8
dtor:          0x821ebc
dtor:          0x821ec0


--- ex14.hpp ---
// ex14.hpp
#include <iostream>
class T {
   int a;
   void log(std::string m) {
      std::cout << m <<" "<< &a <<"\n";
   }
public:
   T(int b)      :a{b}  { log("ctor(int):     "); }
   T()           :a{1}  { log("ctor(default):"); }
   T(const T& x):a{x.a}{ log("ctor(copy):    "); }
   ~T()                 { log("dtor:          "); }
   int get()     const { return a; }
   void set(int x)      { a = x;    }
};
```

```
--- ex14-4a.cpp ---
#include <vector>
#include "ex14.hpp"
int main() {
   std::vector<T> a;
   for (int i = 0; i < 5; i++)
      a.push_back(T(i));
}
```

```
--- ex14-4b.cpp ---
#include <vector>
#include "ex14.hpp"
int main() {
   std::vector<T> a(5);
   for (int i = 0; i < 5; i++)
      a[i].set(5);
}
```

```
=== 演習5 解答例 ===========================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex14-5.cpp
$ ./a.out
0 1 2 4


--- myvec-u.hpp ---
// vectorの内部を知るためのクラス
// 以下の機能だけを使い他のメンバ関数を作る
// vector::vector(int);  サイズ指定で作成
// vector::operator[](); 要素へのアクセス
// vector::swap(); 中身の入れ替え
// これをunique_ptrで置き換えてみる
#include <memory>
template<typename T> void print(const T& a);

template<typename T>
class MyVec {
    size_t capsize{}; // 割り当てサイズ
    size_t cursize{}; // 使用サイズ
    std::unique_ptr<T[]> ar;
public:
    bool   empty()    const       { return cursize == 0; }
    size_t size()     const       { return cursize; }
    size_t capacity() const       { return capsize; }
    //const T&  operator[](size_t i) const { return ar[i]; }
    T  operator[](size_t i) const { return ar[i]; }
    T& operator[](size_t i)       { return ar[i]; }
    void push_back(const T& x)
    {
      if (cursize == capsize) { // 空or満杯の場合は
         // 新しいサイズは空なら1それ以外は容量2倍
         capsize = (capsize==0) ? 1 : 2*capsize;
         auto n { std::make_unique<T[]>(capsize) }; // 新しい割り当て
         for (size_t i = 0; i < cursize; i++) // コピー
            n[i] = ar[i];
         ar.swap(n); // 入れ替え
      }
      ar[cursize] = x;
      ++cursize;
    }
    void pop_back()               { -- cursize; }
    void swap(MyVec<T>& x){
      ar.swap(x);
      std::swap(capsize, x.capsize);
      std::swap(cursize, x.cursize);
    }
    void clear() { cursize = 0; }

    // 汎用性がない
    size_t begin() { return 0; }
    size_t end()   { return cursize; }

    size_t erase(size_t pos) {
       if (pos >= cursize) return cursize;
       for (size_t i = pos; i < cursize-1; i++) // 後半を前シフト
          ar[i] = ar[i+1];
       -- cursize;
       return pos;
    }

    size_t insert(size_t pos, const T& v) {
       if (pos > cursize) pos = cursize;
       if (cursize == capsize) { // 空or満杯の場合は
          capsize = (capsize==0) ? 1 : 2*capsize;
          auto n { std::make_unique<T[]>(capsize) };
          for (size_t i = 0; i < pos; i++) // 前半コピー
             n[i] = ar[i];
          for (size_t i = pos; i < cursize; i++) // 後半コピー
```

```
            n[i+1] = ar[i];
         ar.swap(n); // 入れ替え
      } else {
         for (size_t i = cursize; i > pos; i--) // 後半を後ろシフト
            ar[i] = ar[i-1];
      }
      ar[pos] = v;
      ++cursize;
      return pos;
   }

};
/////////////////
```

```
--- ex14-5.cpp ---
// myvecライブラリのテスト
#include <iostream>
#include "myvec-u.hpp"
int main() {
   MyVec<int> myvec;
   for (size_t i = 0; i < 5; i++)
      myvec.push_back(i);
   myvec.erase(3);
   for (size_t i = 0; i < myvec.size(); i++)
      std::cout << myvec[i] <<" ";
   std::cout <<"\n";
}
```

```
=== 演習6 解答例 ===========================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex14-6.cpp
$ ./a.out
3
dtor: 3
2
dtor: 2
dtor: 1


--- ex14-6.cpp ---
// singly-linked list with unique_ptr
#include <iostream>
#include <memory>

using Ptr = std::unique_ptr<struct Node>;

struct Node {
   int value;
   Ptr nextp;
   Node(int a, Ptr& p)
      :value{a},nextp{std::move(p)}{}
   ~Node() { std::cout <<"dtor: "<< value <<"\n"; }
};

class Stack {
   Ptr ptr;
public:
   void push(int v) {
      ptr = std::make_unique<Node>(v, ptr); }
   void pop() {
      ptr = std::move(ptr->nextp); }
   int top() const { return ptr->value; }
};

int main()
{
   Stack s;
   s.push(1);
   s.push(2);
   s.push(3);
   std::cout << s.top() <<"\n";
   s.pop();
   std::cout << s.top() <<"\n";
}
```