```
=== 演習1 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex04-1.cpp
$ ./a.out
3, 3: 1 2 3
2, 3: 1 2
1, 3: 1
0, 3:


--- ex04-1.cpp ---
#include <iostream>
#include <vector>
using std::cout;

void print(const std::vector<int>& a)
{
   cout << a.size() <<", "<< a.capacity() <<": ";
   for (auto e : a)
      cout << e <<" ";
   cout <<"\n";
}

int main()
{
   std::vector<int> a {1,2,3};
   print(a);    // 3, 3: 1 2 3
   a.pop_back();
   print(a);    // 2, 3: 1 2
   a.pop_back();
   print(a);    // 1, 3: 1
   a.clear();
   print(a);    // 0, 3:
}
```

```
=== 演習2 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex04-2.cpp
$ ./a.out
1 5
1 2 3 4 5


--- ex04-2.cpp ---
#include <iostream>
#include <vector>
using std::cout;

int main() {
   std::vector a {3,2,3,4,8};
   a.front() = 1;
   a.back() = 5;
   cout << a.front() <<" "<< a.back() <<"\n"; // 1 5
   for (auto e : a)
      cout << e <<" ";
   cout <<"\n"; // 1 2 3 4 5
}
```

```
=== 演習3 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex04-3.cpp
$ ./a.out
1 2
2
4


--- ex04-3.cpp ---
#include <iostream>
#include <vector>
using std::cout;
int main()
{
  std::vector a {1,2,3,4};
  auto it { a.begin() };   // イテレータ
  cout << *it <<" "<< *(it + 1)<<"\n"; // 1 2
  ++it;
  cout << *it <<"\n";                  // 2
  cout << a.end() - a.begin() <<"\n";  // 4
}
```

```
=== 演習4 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex04-4.cpp
$ ./a.out
9 8 6 5 3 2 1
9 8 7 6 5 3 2 1
9 8 7 5 3 2 1


--- ex04-4.cpp ---
#include <iostream>
#include <vector>
using std::cout;
void print(const std::vector<int>& a) {
   for (auto e : a) cout << e <<" ";
   cout <<"\n";
}

int main() {
   std::vector a {8,6,5,3,2,1};
   a.insert( a.begin(), 9 );   // push\_front() に相当
   print(a); // 9 8 6 5 3 2 1
   a.insert( a.begin()+2, 7 ); // a[2]に7を挿入
   print(a); // 9 8 7 6 5 3 2 1
   a.erase( a.begin()+3 );     // a[3]を削除
   print(a); // 9 8 7 5 3 2 1
}
```

```
=== 演習5 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex04-5.cpp
$ ./a.out
1
2
3
4
5
1
2
3
4
5


--- ex04-5.cpp ---
#include <iostream>
#include <vector>
int main() {
  std::vector a {1,2,3,4,5};

  for (auto it=a.begin(); it!=a.end(); ++it) {
     auto e { *it };
     std::cout << e <<"\n";
  }

  for (auto e : a)
     std::cout << e <<"\n";
}
```

```
=== 演習6 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex04-6.cpp
$ ./a.out
1 [0]
2 [0 1]
4 [0 1 2]
4 [0 1 2 3]
8 [0 1 2 3 4]
8 [0 1 2 10 3 4]
8 [0 1 10 3 4]


--- myvec.hpp ---
#include <vector>
template<typename T>
class MyVec {
    size_t capsize{}; // 割り当てサイズ
    size_t cursize{}; // 使用サイズ
    std::vector<T> ar;
public:
    bool   empty()    const       { return cursize == 0; }
    size_t size()     const       { return cursize; }
    void   clear()                { cursize = 0; }
    size_t capacity() const       { return capsize; }
    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
         // 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;
         std::vector<T> n(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);
    }

    // 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;
          std::vector<T> n(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;
    }

};


--- ex04-6.cpp ---
// myvecライブラリのテスト
#include <iostream>
#include "myvec.hpp"
void print(const MyVec<int>& a)
{
   std::cout <<a.capacity()<<" [";
   for (size_t i = 0; i < a.size(); i++)
      std::cout <<(i==0?"":" ")<< a[i];
   std::cout <<"]\n";
}

int main()
{
   MyVec<int> x;
   for (int i = 0; i <5; i++) {
      x.push_back(i);
      print(x);
   }
   x.insert(x.begin()+3, 10);
   print(x); // 8 [0 1 2 10 3 4]
   x.erase(x.begin()+2);
   print(x); // 8 [0 1 10 3 4]
}
```

```
=== 演習7 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex04-7.cpp
$ ./a.out
1 [0]
2 [1 0]
4 [2 1 0]
4 [3 2 1 0]
8 [4 3 2 1 0]
8 [4 3 2 10 1 0]
8 [4 3 10 1 0]

1 [0]
2 [1 0]
4 [2 1 0]
4 [3 2 1 0]
8 [4 3 2 1 0]
8 [4 3 2 10 1 0]
8 [4 3 10 1 0]


--- myvec.hpp ---
#include <vector>
template<typename T>
class MyVec {
    size_t capsize{}; // 割り当てサイズ
    size_t cursize{}; // 使用サイズ
    std::vector<T> ar;
public:
    bool   empty()    const       { return cursize == 0; }
    size_t size()     const       { return cursize; }
    void   clear()                { cursize = 0; }
    size_t capacity() const       { return capsize; }
    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
         // 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;
         std::vector<T> n(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);
    }

    // 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;
           std::vector<T> n(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;
    }

};


--- ex04-7.cpp ---
#include <iostream>
#include "myvec.hpp"

template<typename T>
void print(const T& a) {
   std::cout <<a.capacity()<<" [";
   for (size_t i = 0; i < a.size(); i++)
      std::cout <<(i==0?"":" ")<< a[i];
   std::cout <<"]\n";
}

template<typename T>
void test(T& x)
{
   for (int i = 0; i < 5; i++) {
      x.insert(x.begin(), i);
      print(x);
   }
   x.insert(x.begin()+3, 10);
   print(x);
   x.erase(x.begin()+2);
   print(x);
}

int main()
{
   MyVec<int> x;
   test(x);
   std::cout <<"\n";
   std::vector<int> y;
   test(y);
}
```

```
=== 演習8 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex04-8.cpp
$ ./a.out
8 [0 1 2 3 4]
8 [2 3 4]
8 [0 1]


--- myvec.hpp ---
#include <vector>
template<typename T>
class MyVec {
    size_t capsize{}; // 割り当てサイズ
    size_t cursize{}; // 使用サイズ
    std::vector<T> ar;
public:
    bool   empty()    const       { return cursize == 0; }
    size_t size()     const       { return cursize; }
    void   clear()                { cursize = 0; }
    size_t capacity() const       { return capsize; }
    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
         // 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;
         std::vector<T> n(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);
    }

    // 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;
          std::vector<T> n(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;
    }

    // 演習8の解答例
    void erase(size_t first, size_t last) {
        if (last >= cursize) last = cursize;
        if (first < last) {
            for (size_t i = first, j = last; j < cursize; i++, j++)
                ar[i] = ar[j];
            cursize -= (last-first);
        }
    }

};


--- ex04-8.cpp ---
// myvecライブラリのテスト
#include <iostream>
#include "myvec.hpp"
template<typename T>
void print(const MyVec<T>& a)
{
    std::cout <<a.capacity()<<" [";
    for (size_t i = 0; i < a.size(); i++)
        std::cout <<(i==0?"":" ")<< a[i];
    std::cout <<"]\n";
}

int main()
{
    MyVec<int> x, y;
    for (int i = 0; i < 5; i++)
        x.push_back(i);
    print(x); // 8 [0 1 2 3 4]
    y = x;
    y.erase(y.begin(), y.begin()+2);
    print(y); // 8 [2 3 4]
    y = x;
    y.erase(y.begin()+2, y.end());
    print(y); // 8 [0 1]
}
```

```
=== 演習9 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex04-9.cpp
$ ./a.out
5 [1 1 1 1 1]
10 [1 1 1 2 2 2 2 2 1 1]
14 [1 1 1 2 2 3 3 3 3 2 2 2 1 1]


--- myvec.hpp ---
#include <vector>
template<typename T>
class MyVec {
    size_t capsize{}; // 割り当てサイズ
    size_t cursize{}; // 使用サイズ
    std::vector<T> ar;
public:
    bool   empty()    const       { return cursize == 0; }
    size_t size()     const       { return cursize; }
    void   clear()                { cursize = 0; }
    size_t capacity() const       { return capsize; }
    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
         // 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;
         std::vector<T> n(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);
    }

    // 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;
          std::vector<T> n(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;
    }

    // 演習8の解答例
    void erase(size_t first, size_t last) {
        if (last >= cursize) last = cursize;
        if (first < last) {
            for (size_t i = first, j = last; j < cursize; i++, j++)
                ar[i] = ar[j];
            cursize -= (last-first);
        }
    }

    // 演習9の解答例
    void insert(size_t pos, size_t count, const T& v) {
        if (pos > cursize) pos = cursize;
        if (cursize+count > capsize) { // 足りない場合
            capsize = cursize+count;
            std::vector<T> n(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+count] = ar[i];
            ar.swap(n); // 入れ替え
        } else {
            for (size_t i = cursize+count-1; i > pos+count; i--) // 後半を後ろシフト
                ar[i] = ar[i-(cursize-pos)];
        }
        for (size_t i = 0; i < count; i++)
            ar[pos+i] = v;
        cursize += count;
    }
};


--- ex04-9.cpp ---
#include <iostream>
#include "myvec.hpp"
template<typename T>
void print(const MyVec<T>& a)
{
    std::cout <<a.capacity()<<" [";
    for (size_t i = 0; i < a.size(); i++)
        std::cout <<(i==0?"":" ")<< a[i];
    std::cout <<"]\n";
}

int main()
{
    MyVec<int> x;
    x.insert(x.begin(), 5, 1);
    print(x); // 5 [1 1 1 1 1]
    x.insert(x.begin()+3, 5, 2);
    print(x); // 10 [1 1 1 2 2 2 2 2 1 1]
    x.insert(x.begin()+5, 4, 3);
    print(x); // 14 [1 1 1 2 2 3 3 3 3 2 2 2 1 1]
}
```

```
=== 演習10 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex04-10.cpp
$ ./a.out
8 [1 6 7 8 9 2 3 4]


--- ex04-10.cpp ---
#include <iostream>
#include <vector>
template<typename T>
void print(const std::vector<T>& a)
{
   std::cout <<a.capacity()<<" [";
   for (size_t i = 0; i < a.size(); i++)
      std::cout <<(i==0?"":" ")<< a[i];
   std::cout <<"]\n";
}


int main()
{
   std::vector a{1,2,3,4}, b{5,6,7,8,9};
   a.insert(a.begin()+1, b.begin()+1, b.end());
   print(a); // 8 [1 6 7 8 9 2 3 4]
}

/*
 * MyVecのイテレータはsize_tで別のオブジェクトbの情報が引数から得られないため
 */
```