```
=== 演習1 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex13-1.cpp
$ ./a.out
int:    sizeof(x) = 4, sizeof(x[5]) = 20
double: sizeof(x) = 8, sizeof(x[5]) = 40
Single: sizeof(x) = 4, sizeof(x[5]) = 20
Pair:   sizeof(x) = 16, sizeof(x[5]) = 80


--- ex13-1.cpp ---
// サイズ確認
#include <iostream>
template<typename T>
void size(std::string n) {
   T x;
   T xa[5];
   std::cout << n <<":\t"
    <<"sizeof(x) = "   << sizeof(x) <<", "
    <<"sizeof(x[5]) = "<< sizeof(xa)<<"\n";
}

class Single { int a; };
class Pair   { int a; double b; };

int main()
{
   size<int>("int");
   size<double>("double");
   size<Single>("Single");
   size<Pair>("Pair");
}
```

```
=== 演習2 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex13-2.cpp
$ ./a.out
int[10]:        sizeof(x) = 40, sizeof(x[5]) = 200
CArray :        sizeof(x) = 40, sizeof(x[5]) = 200
vector<int>:    sizeof(x) = 24, sizeof(x[5]) = 120
vector<double>: sizeof(x) = 24, sizeof(x[5]) = 120
Vector :        sizeof(x) = 24, sizeof(x[5]) = 120


--- ex13-2.cpp ---
// サイズ確認
#include <iostream>
#include <vector>
template<typename T>
void size(std::string n) {
   T x;
   T xa[5];
   std::cout << n <<":\t"
    <<"sizeof(x) = "   << sizeof(x) <<", "
    <<"sizeof(x[5]) = "<< sizeof(xa)<<"\n";
}

class CArray { int a[10]; };
class Vector {
   std::vector<int> a{1,2,3,4,5,6,7,8,9,0};
};

int main()
{
   size<int[10]>("int[10]");
   size<CArray>("CArray ");
   size<std::vector<int>>("vector<int>");
   size<std::vector<double>>("vector<double>");
   size<Vector>("Vector ");
}
```

```
=== 演習3 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex13-3a.cpp
$ ./a.out
1074440874 0 -1431655765
$ g++ -std=c++17 ex13-3b.cpp
$ ./a.out
0 0 0


--- ex13-3a.cpp ---
// 未初期化
#include <iostream>
void f() { double a {10.0/3.0}; }

void g()
{
   int x, y[2]; // 悪い例
   std::cout << x <<" "
      << y[0] <<" "
      << y[1] <<"\n";
}

int main()
{
   f();
   g();
}


--- ex13-3b.cpp ---
// 未初期化
#include <iostream>
void f() { double a {0.0}; }

void g()
{
   int x, y[2]; // 悪い例
   std::cout << x <<" "
      << y[0] <<" "
      << y[1] <<"\n";
}

int main()
{
   f();
   g();
}
```

```
=== 演習4 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex13-4a.cpp
$ ./a.out
0 -1431655765 1074440874
$ g++ -std=c++17 ex13-4b.cpp
$ ./a.out
0 0 0


--- ex13-4a.cpp ---
// 未初期化
#include <iostream>
struct T {
   int x;
   int y[2];
};

void f() { double a {10.0/3.0}; }

void g()
{
   T a; // 悪いかもしれない例
   std::cout << a.x <<" "
      << a.y[0] <<" "
      << a.y[1] <<"\n";
}

int main()
{
   f();
   g();
}



--- ex13-4b.cpp ---
// 未初期化
#include <iostream>
struct T {
   int x {0};
   int y[2] {0};
};

void f() { double a {10.0/3.0}; }

void g()
{
   T a; // 悪いかもしれない例
   std::cout << a.x <<" "
      << a.y[0] <<" "
      << a.y[1] <<"\n";
}

int main()
{
   f();
   g();
}
```

```
=== 演習5 解答例 ============================
--- compile-and-run.txt ---
]$ g++ -std=c++17 ex13-5.cpp
ex13-5.cpp: 関数 ‘void print(double*, int)’ 内:
ex13-5.cpp:9:38: 警告: ‘sizeof’ on array function parameter ‘a’ will return size of
 ‘double*’ [-Wsizeof-array-argument]
    9 |     std::cout <<"sizeof(a)="<< sizeof(a) <<"\n";
      |                                      ^~
ex13-5.cpp:5:19: 備考: ここで宣言されています
    5 | void print(double a[], int n)
      |            ~~~~~~~^~~
$ ./a.out
1
1.41421
1.73205
2
2.23607
2.44949
2.64575
2.82843
3
3.16228
sizeof(a)=8


--- ex13-5.cpp ---
// 配列引数
#include <iostream>
#include <cmath>

void print(double a[], int n)
{
   for (int i = 0; i < n; i++)
      std::cout << a[i] <<"\n";
   std::cout <<"sizeof(a)="<< sizeof(a) <<"\n";
}

int main()
{
   double x[10];
   for (int i = 0; i < 10; i++)
      x[i] = std::sqrt(1.0+i);
   print(x, 10);
}
```

```
=== 演習6 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex13-6.cpp
$ ./a.out
1, 2, 3, 4
5, 6, 7, 8
---
1, 5
2, 6
3, 7
4, 8


--- ex13-6.cpp ---
// 2次元配列の引数
#include <iostream>

void print(int a[][4], int n)
{
   for (int i = 0; i < n; i++) {
      for (int j = 0; j < 4; j++) {
         if (j != 0) std::cout <<", ";
         std::cout << a[i][j];
      }
      std::cout <<"\n";
   }
}

void printT(int a[][4], int n)
{
   for (int j = 0; j < 4; j++) {
      for (int i = 0; i < n; i++) {
         if (i != 0) std::cout <<", ";
         std::cout << a[i][j];
      }
      std::cout <<"\n";
   }
}

int main()
{
   int x[2][4] {{1,2,3,4},{5,6,7,8}};
   print(x, 2);
   std::cout<<"---\n";
   printT(x, 2);
}
```

```
=== 演習7 解答例 ============================
--- compile-and-run.txt ---
$ g++ -std=c++17 ex13-7.cpp
$ ./a.out
5, 6, 7, 8
1, 2, 3, 4


--- ex13-7.cpp ---
// 2次元配列の引数
#include <iostream>

void print(int* a[], int n, int m)
{
   for (int i = 0; i < n; i++) {
      for (int j = 0; j < m; j++) {
         if (j != 0) std::cout <<", ";
         std::cout << a[i][j];
      }
      std::cout <<"\n";
   }
}

int main()
{
   int x[2][4] {{1,2,3,4},{5,6,7,8}};
   int* p[2] {x[1],x[0]};
   print(p, 2, 4);
}
```