第13回演習

演習 1 変数のサイズ ex13-1.cpp

以下のプログラムを実行して、それぞれの型に対する単一の変数と配列のサイズを比較せよ。得られた数字から分かることを 1,2 行で答えよ。

```
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サイズ比較 ex13-2.cpp

前の演習の size() テンプレート関数を利用して、配列、配列を含むクラス、vector, vector を含むクラスのサイズを比較せよ。この方法では vector 要素のサイズ情報が得られないことも確認せよ。

```
template<typename T>
void size(std::string n); // 前の演習と同じ

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 未初期化 ex13-3[ab].cpp

組み込み型の変数とその配列の初期化を行わないと結果がどうなるかを確認せよ (a)。また、関数 `f(){ double a{0.0}; }`とすると結果がどう変わるかも試せ (b)。環境によっては変化がない場合もある。提出プログラムは二つとする

```
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();
}
```

`ex13-3a.cpp` と `ex13-3b.cpp` 双方を作り、
それぞれのコンパイルと実行を行う。

演習 4 未初期化 ex13-4[ab].cpp

構造体の初期化を行わないと結果がどうなるかを確認せよ (a)。また、構造体 T の定義のみを変更して、すべてのメンバを 0 に初期化するように変更したプログラムも作成せよ (b)。提出プログラムは二つ。

```
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(); }
```

ex13-4a.cpp と ex13-4b.cpp 双方を作り、
それぞれのコンパイルと実行を行う。

演習 5 配列引数 ex13-5.cpp

print 関数の引数 a の宣言を指定し、その引数のサイズを確認せよ。なお、コンパイル時に警告が出ても、sizeof 関連ならばそのままで良い。

```
#include <iostream>
#include <cmath>
void print( /*  a の宣言  */  , 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);
}
```

```
$ ./a.out
1
... 中略
sizeof(a)=8
```

演習 6 配列引数 ex13-6.cpp

2x4 の 2 次元配列を行列のように出力する print() 関数と転置行列のように出力する printT() を作成せよ。

```
void print(int a[][4], int n){ /* write here */ }
void printT /* write here */ }

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

```
1, 2, 3, 4
5, 6, 7, 8
---
1, 5
2, 6
3, 7
4, 8
```

演習 7 配列引数 ex13-7.cpp

ポインタ配列の指す内容を行列のように出力する print() 関数を作成せよ。ヒント：資料 p.22 と p.28 を見比べよ。

```
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);
}
```

```
5, 6, 7, 8
1, 2, 3, 4
```