第3章「たくさんのデータを扱う」

主なトピック

- 要素数可変の配列　ベクトル型（std::vector）
- ライブラリのアルゴリズム（std::sort）の利用

個数がわからない大量のデータを扱う

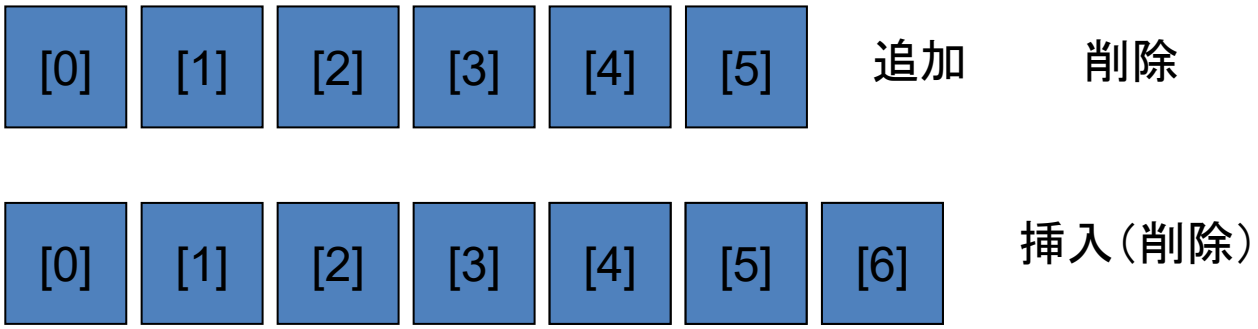
- 例えば，ファイルの読み込み
  - 何行あるか解らない
- 今回の課題だと，演習の提出回数
  - 個人により提出回数が異なる
- このようなデータをプログラムの中で保存したい
  - 配列だと最大個数をあらかじめ設定
  - 大きな配列をとったとしても，使わなかった無駄
  - 足りなかったら全然ダメ
  - どうする？

メモリを使う分だけ用意する

- メモリの動的確保
  - 使うときに使う分だけ，システムからメモリをもらう
  - 使い終わったらメモリをシステムに返す
  - C++なら，new と delete
  - C言語なら，malloc と free
  - でもC++には，簡単に使えるライブラリが用意
  - コンテナ・クラス
    - vector, list, map など多数

vector

- 配列に要素の追加と削除を可能
- データの順序を保持
- 様々な型のデータ（int, double, string など任意のクラス）を保管可能
  - テンプレート



vectorのサンプルソース

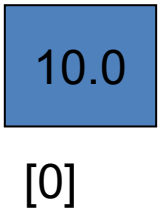
```
#include <vector>
#include <iostream>

int main()
{
	std::vector<double>  homework;
	homework.clear();
	homework.push_back(10.0);
	homework.push_back(5.3);
	homework.push_back(21.8);
	std::cout << homework[1] << std:: endl;
}
```

vectorのサンプルソースのイメージ

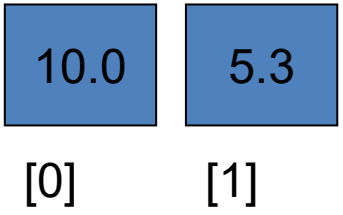
homework.clear();

データなし

homework.push_back(10.0);



homework.push_back(5.3);



homework.push_back(21.8);

サンプルソースの解説

- #include <vector>
  - vectorを使う時に必要なインクルードファイル
- std::vector<double> homework;
  - double型の（配列のような）vector の homework という名前の変数を宣言
- homework.clear();
  - homeworkという名前の変数に保存されているデータを消去する
- homework.push_back(10.0);
  - homeworkの最後に10.0というデータを追加
- homework[1];
  - homeworkの先頭から2番目のデータにアクセス

その他のvectorのメンバ関数(1/2)

- homework.pop_back();
  - 末尾の要素を削除
- homework.size();
  - homeworkに格納されている要素の数を返す
- std::vector<double>::size_type
  - vectorに格納されている要素の数を表現するための型
  - std::vector<double>::size_type size = homework.size();
- 他にもたくさんの関数がある
  - 例えば，insert(), erase()
  - これらはイテレータを学んでから、改めて学習

ソート

- データを大きさ順に並び替え
- C++では sort という関数が提供
- 対象はコンテナ，もちろんvectorも含まれる
- サンプルソース

```
#include <algorithm>
int main()
{
    std::vector<double> homework;
    //      すでにhomeworkにデータが格納されているとして
    std::sort( homework.begin(), homework.end() );
}
```

ソートのサンプルソースの解説

- #include <algorithm>
  - sort 関数が含まれるヘッダファイル
- std::sort(homework.begin(), homework.end());
  - sort(a, b);
    - aからbまでの範囲を昇順（非降順）に，つまりだんだん大きくなるようにソート
  - homework.begin()
    - homeworkの先頭の要素を表す
  - homework.end()
    - homeworkの末尾の要素の一つ後を表す

標準出力ストリームの書式

- 数字を出力するときの書式設定
  - 表示桁数，小数点以下の桁数
  - 固定小数点，指数表現
  - 右寄せ，左寄せ，など
- cout に書式の設定を出力，または書式設定の関数を適用
  - cout << setprecision(3) << a;
  - cout.precision(3); cout << a;
  - 上は両者とも小数点以下3桁で出力

オペレータと関数の違い

- オペレータ（操作子）
  - cout << setprecision(3) << a;
  - cout << b;
    - setprecision(3) はaの出力にのみ有効
    - b の出力はデフォルトの設定に戻る
- 関数
  - cout.precision(3);
  - cout << a;
  - cout << b;
    - この関数の適用以降ずっと（a も b も）有効

標準出力ストリームのサンプルソース

```
#include <iostream> //  標準ストリームを利用するときに必要
#include <iomanip>  //  書式設定を利用するときに必要

int main()
{
    //  129.65と出力される
    std::cout << std::setw(8) << std::setprecision(2) << 129.653 << endl;
    //  129.7と出力される
    std:: cout.precision(1);
    std::cout.width(8);
    std:: cout << 129.65395 << std::endl;
}
```

出力ストリームの書式設定

- std:: cout << std:: setprecision(n);
- std:: cout.precision(n);
  - 有効数字，小数点以下の桁数を指定
- std:: cout << std:: setw(n);
- std:: cout.width(n);
  - 出力するときの幅，文字数を指定
- std:: cout << std:: fixed;
- std:: cout.setf(std:: ios_base::fixed, std:: ios_base::floatfield);
  - 固定小数点での出力
- std::cout.setf(std:: ios::showpoint);
  - 小数点の強制表示
- 他にもたくさんの書式設定が可能

ストリームの評価

- std::cin >> x
  - 標準入力ストリームから x に代入
- (std:: cin >> x)
  - 上の代入文を「評価」
  - 正しく代入できたら，真(true)
  - 代入が失敗だったら（入力がない），偽(false)
  - while(std:: cin >> x){}
  - if(std:: cin >> x)　　{}

小技

- typedef
  - 変数の型の別名を定義
  - std:: vector<double>::size_type size = homework.size();
  - typedef std::vector<double>::size_type vec_sz;
  - 長い型名を vec_sz と短い別名
  - vec_sz size = homework.size();

演習課題3のポイント(1/2)

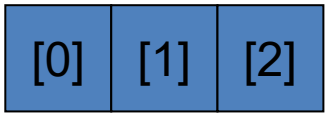
- データの読み取り方は2重ループ

```
//  外側のループは各行（学生一人ひとり）の読み取り
// 学生番号，姓，名，中間試験，期末試験などを読み込む
while()  {
      // 内側のループは演習の読み取り
      // -1が表れるまで読み続ける
      while()        {
            // vectorの末尾に追加;
      }
}
```

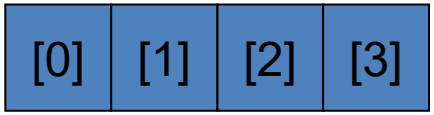
演習課題3のポイント(2/2)

- メジアン（中央値）
  - データを小さいものから順に並べたときの，中央の順位の値
  - 9件あるときは第5位
  - 10件あるときは，第5位と第6位の平均
  - vector, sort

a: ソートされているとして



メジアンはa[1]



メジアンは(a[1]+a[2])/2

クイズ解説

- 標準入力ストリームから実数値を
  EOFになるまで読み込み，それぞれ
  をstd::vector<double>型の変数bに
  格納し，それをソートした結果を出力
  するプログラムを書け．
  注(1)必要なインクルード文を書くこと，
  (2)main()関数も書くこと，(3)必要な
  変数宣言を書くこと