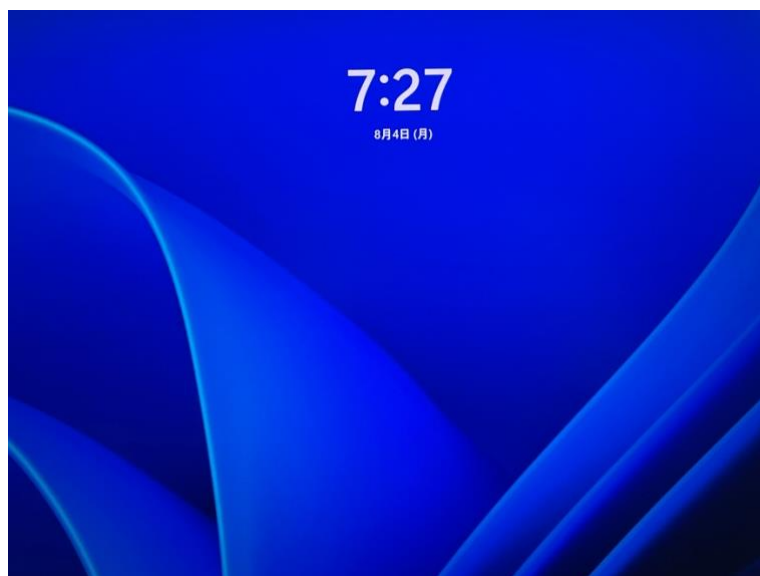


[オリエンテーション]

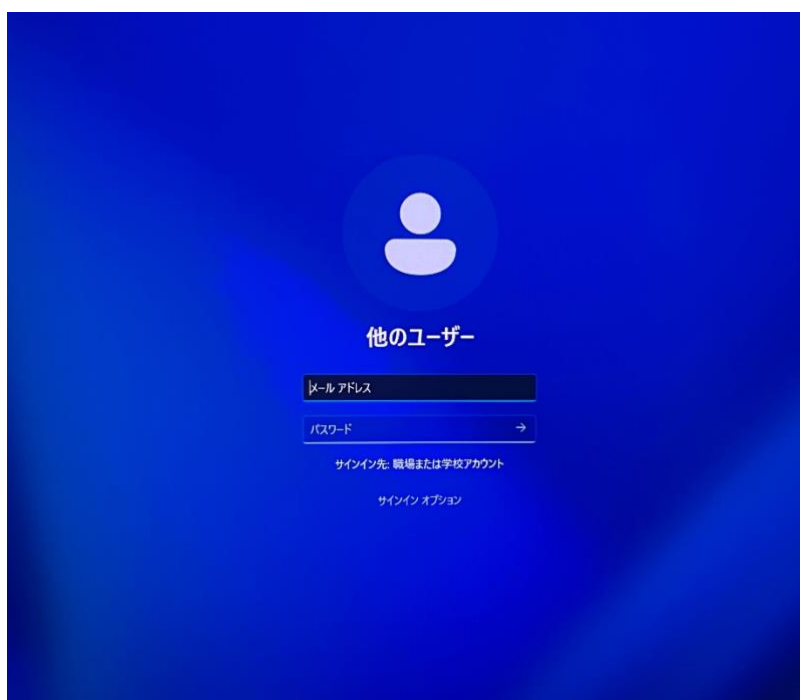
(1) 計算機室の PC へのログイン

本講習会では計算機室の PC を利用します。計算機室の PC を利用するには、静大 ID でログインする必要があります。以下の説明を確認し、配布された静大 ID でログインしてください、

(1 日目については、静大 ID の封を切る作業がありますので、切り取った紙は TA が回収します。また、お帰りの際に静大 ID は回収させていただきます。)



パソコンの電源を入れ、上図のような画面が出てきたら何かキーを押して下さい。そうするとアカウントとパスワードを入力する画面が出てくるはずです。



この画面が出てきたら配布された静大 ID とパスワードをそれぞれ入力してください。静大 ID とパスワードは配布した用紙を開き、見開きの右上部にあります。

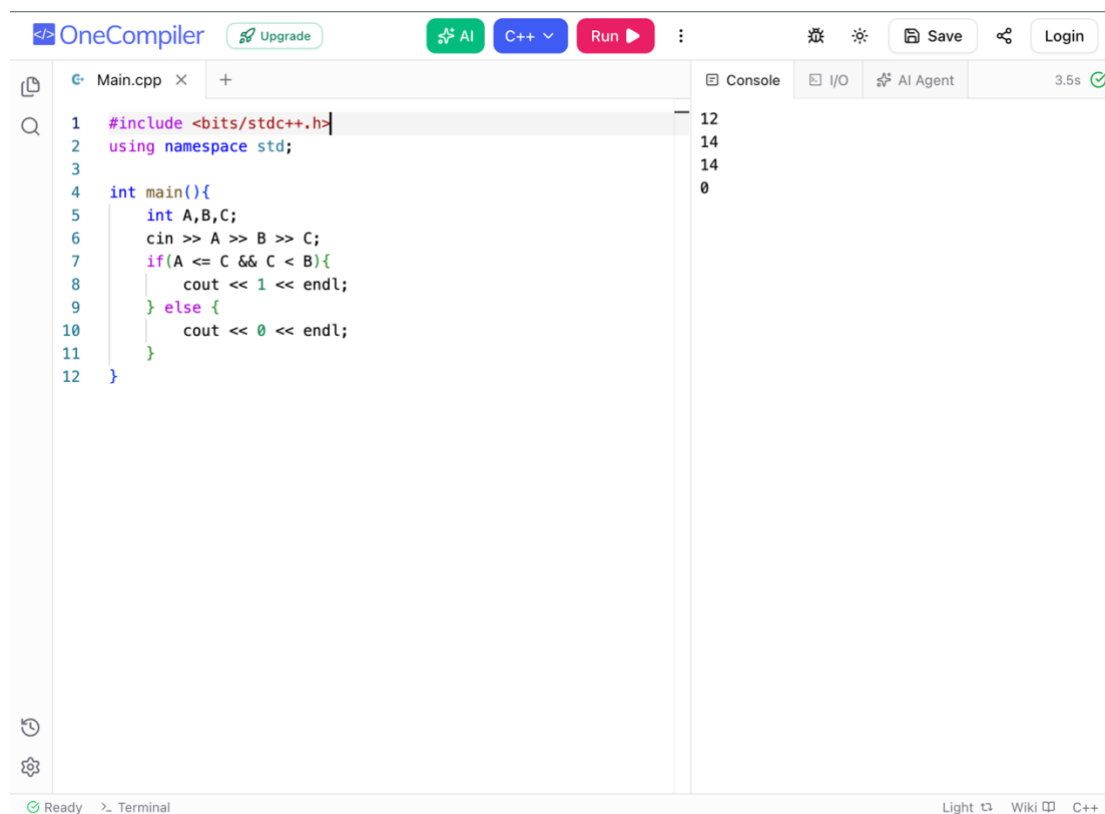
静岡大学の全学情報基盤サービスにログインする
ときに必要な認証情報を発行します。

静 大 I D :	
パスワード :	

(2) 本講習会におけるプログラミング環境

本講習会では、**OneCompiler** というサービスを利用します。OneCompiler は、ブラウザ上でコードの編集、コンパイル、実行ができます。複数の言語に対応しており、C、C++、Java、Python などのプログラミング言語だけでなく、HTML などでも利用できます。

C++を利用するには、<https://onecompiler.com/cpp> にアクセスしてください。OneCompiler では通常のコンソール実行と標準入力の事前入力の両方を行えます。AtCoder と同様の形式で入力をする場合は画面右側の **I/O** タブを選択し、利用してください。



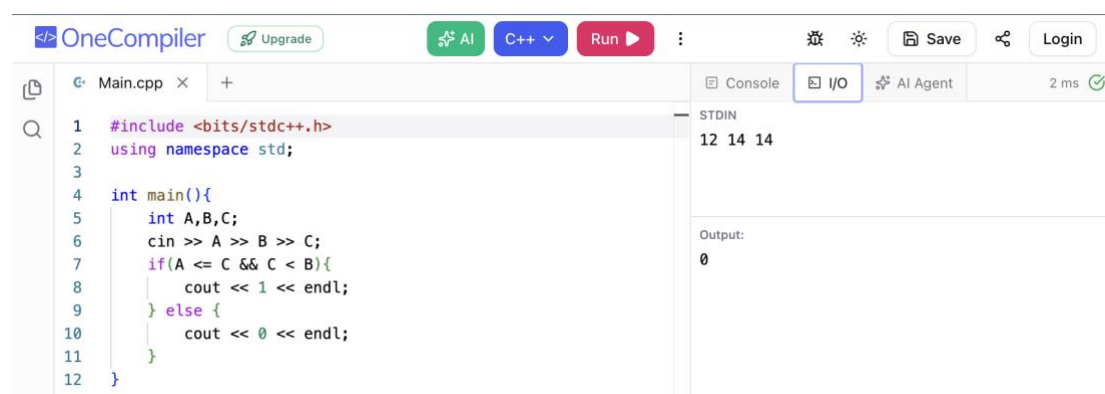
The screenshot shows the OneCompiler web interface. The code editor on the left contains the following C++ code:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main(){
5     int A,B,C;
6     cin >> A >> B >> C;
7     if(A <= C && C < B){
8         cout << 1 << endl;
9     } else {
10        cout << 0 << endl;
11    }
12 }
```

The right panel shows the 'Console' tab with the following output:

```
12
14
14
0
```

The status bar at the bottom indicates 'Ready' and 'Terminal'.



The screenshot shows the OneCompiler web interface with the 'I/O' tab selected. The code editor on the left contains the same C++ code as the previous screenshot:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main(){
5     int A,B,C;
6     cin >> A >> B >> C;
7     if(A <= C && C < B){
8         cout << 1 << endl;
9     } else {
10        cout << 0 << endl;
11    }
12 }
```

The right panel shows the 'I/O' tab with the following input and output:

```
STDIN
12 14 14

Output:
0
```

The status bar at the bottom indicates '2 ms' and 'C++'.

また、オンラインで C++ の編集、コンパイル、実行ができるサービスは他にもありますので紹介します。

Wandbox というサービスでも、同様にコード編集、コンパイル、実行をすることが出来ます。例えば、ブラウザ上で下記のようにすると事前入力した標準入力を用いて、コンパイル・実行ができます。Wandbox も複数言語に対応しています。

Wandbox online C++ IDE interface showing a C++ program and its execution.

```
#include <bits/stdc++.h>
using namespace std;

int main () {
    int A, B, C;
    cin >> A >> B >> C;
    if (A <= C && C < B) {
        cout << 1 << endl;
    } else {
        cout << 0 << endl;
    }
}
```

標準入力: 1 12 14 14

コンパイルコマンド: `$ g++ prog.cc -Wall -Wextra -I/opt/wandbox/boost-1.83.0-gcc-13.2.0/include -std=gnu++2b`

実行 共有

C++ shell (cpp.sh)も同様のサービスです。コードを記述して Run ボタンを押すと実行できます。しかしながら、OneCompiler や wandbox と違い、標準入力を AtCoder のような形式で事前入力しておくことができません。また、`<bits/stdc++.h>`が使えませんので、`<iostream>`や`<string>`を適切に include する必要があります。

The screenshot shows the cpp.sh online C++ compiler interface. At the top, it says "C++ shell" and "cpp.sh online C++ compiler". Below this, there is a text area containing a C++ program:

```
1 // Example program
2 #include <iostream>
3 #include <string>
4
5 int main()
6 {
7     std::string name;
8     std::cout << "What is your name? ";
9     getline (std::cin, name);
10    std::cout << "Hello, " << name << "!\n";
11 }
```

Below the code area, there is a "Run" button. Underneath the button, there are three tabs: "options", "compilation", and "execution". The "execution" tab is selected, showing the output of the program:

```
What is your name? john
Hello, john!
```

At the bottom, there is a status bar that says "Normal program termination. Exit status: 0".

① 10:15～11:00『基本的なプログラム，入出力の方法を確認しよう』

例題 ■ 1.00.はじめに (C++入門 AtCoder Programming Guide for beginners (APG4b) A)

https://atcoder.jp/contests/apg4b/tasks/APG4b_a

C++ 文法の確認

文字列等をコンソールに出力する場合は，`cout` を使用する．

使用例： `cout << “文字列” << endl; // endl は改行`

課題リンク先の指示にしたがって，AtCoder へのプログラムの提出方法を確認しよう．

言語は “C++ 23 (GCC 15.2.0)” を選ぼう．

結果が  になれば OK！

例題 ■ 身長(JOI 2021/2022 一次予選 (第 3 回) 問題 A)

https://atcoder.jp/contests/joi2022yo1c/tasks/joi2022_yo1c_a

C++ 文法の確認

整数の変数は `int` を使って宣言する

使用例： `int A, B;`

数字等をキーボードから入力するときは，`cin` を使用する

使用例： `cin >> A >> B;`

2 つの整数 A, B を入力し， $B-A$ の値を画面に出力するプログラムを作る．

Tips： AtCoder に課題を提出するときは，出力すべき情報を出力した最後に「改行」を出力することを忘れずに！

```
// 多分こんなプログラムだろう
#include <bits/stdc++.h>
using namespace std;

int main() {
    int A, B;
    cin >> A >> B;
    cout << B-A << endl;
}
```

それでは演習！ ひとまず下記 2 つにチャレンジ.

演習 ■ 立方体 (JOI 2021/2022 一次予選 (第 2 回) 問題 A)

https://atcoder.jp/contests/joi2022yo1b/tasks/joi2022_yo1b_a

演習 ■ box (AtCoder Beginner Contest 180 問題 A)

https://atcoder.jp/contests/abc180/tasks/abc180_a

上記問題を解く際に C++の文法等で困る状況になった場合は

C++入門 AtCoder Programming Guide for beginners (APG4b)

<https://atcoder.jp/contests/apg4b/tasks>

の, A: 1.00 ~ F: 1.05 の問題に記載の解説を確認しよう！

ここまで楽勝という方は次の 3 つも挑戦してみよう.

演習 ■ 合計時間 (JOI 2010/2011 予選 問題 A)

https://atcoder.jp/contests/joi2011yo/tasks/joi2011yo_a

演習 ■ Garden (AtCoder Beginner Contest 106 問題 A)

https://atcoder.jp/contests/abc106/tasks/abc106_a

演習 ■ Payment (AtCoder Beginner Contest 173 問題 A)

https://atcoder.jp/contests/abc173/tasks/abc173_a

ここまでも楽勝！という方は②以降も進めてみてください.

② 11:00～12:00『条件分岐の使い方，文字列の扱い方を確認しよう』

C++ 文法の確認

条件分岐は `if` を使う

使用例：

```
if (A == B) {  
    cout << "A=B" << endl;  
} else {  
    cout << "A!=B" << endl;  
}
```

こちらも確認！

G - 1.06.if 文・比較演算子・論理演算子 (https://atcoder.jp/contests/apg4b/tasks/APG4b_g)

H - 1.07.条件式の結果と `bool` 型 (https://atcoder.jp/contests/apg4b/tasks/APG4b_h)

例題 ■ 帰省 (JOI 2020/2021 一次予選 (第 2 回) 問題 A) [条件分岐]

https://atcoder.jp/contests/joi2021yo1b/tasks/joi2021_yo1b_a

ビ太郎：A 日後の午前に実家に行き，B 日後の午前に帰る

ビバ子：C 日後の午後にビ太郎の実家に行く

入力：A B C

出力：ビバ子がビ太郎に会えるかどうか（会える場合は 1，会えない場合は 0 を出力）

考え方：A<=C かつ C<B のときに会えるはず（C<=B ではないので注意）

```
// 多分こんな感じ  
#include <bits/stdc++.h>  
using namespace std;  
  
int main () {  
    int A, B, C;  
    cin >> A >> B >> C;  
    if (A <= C && C < B) {  
        cout << 1 << endl;  
    } else {  
        cout << 0 << endl;  
    }  
}
```

C++ 文法の確認

文字列を使う場合は `string` 型を使う

使用例： `string str = "sample";`

```
int len = str.size(); // 文字列の長さ
if (len > 3) {
    cout << str.at(0) << endl; // n 文字目を参照する：.at(n-1) or [n-1]
    // cout << str[0] << endl; // こちらでも良い
}
```

部分文字列は、`.substr()`を使う。

使用例： `string str = "sample";`

```
cout << str.substr(3); // 4 文字目から最後まで：ple が出力される
cout << str.substr(0,3); // 1 文字目から 3 文字：sam が出力される
cout << str.substr(2,4); // 3 文字目から 4 文字：mple が出力される
```

逆順にしたければ、`reverse()`を使う

使用例： `string str = "sample";`

```
reverse(str.begin(), str.end());
cout << str; // elpmas が出力される
```

こちらも確認！

M - 1.12.文字列と文字 (https://atcoder.jp/contests/apg4b/tasks/APG4b_m)

例題 ■ Rotate (AtCoder Beginner Contest 197 問題 A) [文字列]

https://atcoder.jp/contests/abc197/tasks/abc197_a

入力：3 英小文字の文字列

出力：2 文字目，3 文字目，1 文字目の順番で出力する

```
string S;
```

```
cin >> S;
```

`S.at(1)`，`S.at(2)`，`S.at(0)` の順番で出力すれば良さそう。

`S[1]`，`S[2]`，`S[0]` でも OK！

添字番号が 0 から始まる点に注意！

```
// 多分こんな感じ
#include <bits/stdc++.h>
using namespace std;

int main () {
    string S;
    cin >> S;
    cout << S.at(1) << S.at(2) << S.at(0) << endl;
}
```

下記問題に挑戦しよう！

演習 ■ 計算 (JOI 2020/2021 一次予選 (第 3 回) 問題 A) [条件分岐]

https://atcoder.jp/contests/joi2021yo1c/tasks/joi2021_yo1c_a

演習 ■ Weather Forecast (AtCoder Beginner Contest 218 問題 A) [文字列]

https://atcoder.jp/contests/abc218/tasks/abc218_a

演習 ■ 3 つの整数 (JOI 2019/2020 一次予選 (第 1 回) 問題 A) [条件分岐]

https://atcoder.jp/contests/joi2020yo1a/tasks/joi2020_yo1a_a

演習 ■ Plural Form (AtCoder Beginner Contest 179 問題 A) [文字列]

https://atcoder.jp/contests/abc179/tasks/abc179_a

終わったら下記も進めてみよう。

演習 ■ アイスクリーム (JOI 2021/2022 一次予選 (第 3 回) 問題 B) [条件分岐]

https://atcoder.jp/contests/joi2022yo1c/tasks/joi2022_yo1c_b

演習 ■ Tires (AtCoder Beginner Contest 224 問題 A) [文字列]

https://atcoder.jp/contests/abc224/tasks/abc224_a

演習 ■ 試験 (JOI 2019/2020 一次予選 (第 2 回) 問題 A) [条件分岐]

https://atcoder.jp/contests/joi2020yo1b/tasks/joi2020_yo1b_a

演習 ■ Registration (AtCoder Beginner Contest 167 問題 A) [文字列]

https://atcoder.jp/contests/abc167/tasks/abc167_a

演習 ■ 2 番目に大きい整数 (JOI 2019/2020 一次予選 (第 1 回) 問題 A) [条件分岐]

https://atcoder.jp/contests/joi2021yo1a/tasks/joi2021_yo1a_a

演習 ■ chukodai (AtCoder Beginner Contest 236 問題 A) [文字列]

https://atcoder.jp/contests/abc236/tasks/abc236_a

ここまで楽勝！の人は③以降に進んでください。

③ 12:50～13:40『繰り返しの方法を確認しよう』

C++ 文法の確認

繰り返しを記述する場合は、`for` や `while` を使おう！

使用例： `for(i=0; i<n; i++) {}` // n 回ループするときの定型文！

`while(i<n) { i++; }`

`for` は「繰り返し回数が固定ないしは入力値で決まる」場合、`while` は「繰り返し回数が入力により例外的に変わる」場合に使うことが多い。

こちらも確認！

K - 1.10.`while` 文 (https://atcoder.jp/contests/apg4b/tasks/APG4b_k)

L - 1.11.`for` 文・`break`・`continue` (https://atcoder.jp/contests/apg4b/tasks/APG4b_l)

AP4 - 付録 4.ループの裏技 `rep` マクロ (https://atcoder.jp/contests/apg4b/tasks/APG4b_an)

`rep` マクロとは？

```
// 多分こんな感じ
#include <bits/stdc++.h>
using namespace std;

#define rep(i, s, n) for (int i = s; i < (int)(n); i++)

int main() {
    int N, i;
    string S;
    cin >> N >> S;
    rep(i, 0, N) {
        // for (int i=0; i<N; i++) { を短く書ける
    }
}
```

例題 ■ 次の文字（JOI 2021/2022 一次予選（第 2 回）問題 C）

https://atcoder.jp/contests/joi2022yo1b/tasks/joi2022_yo1b_c

入力文字列(J か I か O のいずれか)

入力：文字数 N と文字列 S

出力：J があった場合その一文字前の文字と改行を出力

注意点：1 文字～N 文字 は、プログラム上の要素番号が 0～N-1 になる！

```
// 多分こんな感じ
#include <bits/stdc++.h>
using namespace std;

int main() {
    int N, i;
    string S;
    cin >> N >> S;
    for(i=1;i<=N-1;i++) {
        if (S.at(i) == 'J') {
            cout << S.at(i-1) << endl;
        }
    }
}
```

例題 ■ Wwwwvvvvvv (AtCoder Beginner Contest 279 問題 A)

https://atcoder.jp/contests/abc279/tasks/abc279_a

入力：文字列 S (v か w の 2 文字から構成)

出力：下に尖っている個数 (v なら 1, w なら 2 でカウント)

文字数は $S.size()$ で取得できる

個数カウントの変数を別途準備し, `for`, `if` を使えばいけそう

```
// 多分こんな感じ
#include <bits/stdc++.h>
using namespace std;

int main() {
    string S;
    int i, c=0;
    cin >> S;
    for(i=0; i<(int)S.size(); i++) { // (int) がないと警告が出る(実行は可能)
        if (S.at(i) == 'v') {
            c += 1;
        } else if (S.at(i) == 'w') {
            c += 2;
        }
    }
    cout << c << endl;
}
```

※ $S.size()$ が返す数値は, `long unsigned int` 型なので, `(int)` で `int` 型に型キャストする

※ より詳しくは https://atcoder.jp/contests/apg4b/tasks/APG4b_m の 応用 の欄を参照

下記問題に挑戦しよう！

演習 ■ Rightmost (AtCoder Beginner Contest 276 問題 A)

https://atcoder.jp/contests/abc276/tasks/abc276_a

演習 ■ IOI 文字列 (JOI 2020/2021 一次予選 (第 3 回) 問題 B)

https://atcoder.jp/contests/joi2021yo1c/tasks/joi2021_yo1c_b

演習 ■ 運動会 (JOI 2021/2022 一次予選 (第 3 回) 問題 C)

https://atcoder.jp/contests/joi2022yo1c/tasks/joi2022_yo1c_c

簡単簡単！という方は下記にもチャレンジ。

演習 ■ JOI ソート (JOI 2020/2021 一次予選 (第 1 回) 問題 B)

https://atcoder.jp/contests/joi2021yo1a/tasks/joi2021_yo1a_b

演習 ■ 巻物 (JOIG 2021 問題 B)

https://atcoder.jp/contests/joig2021-open/tasks/joig2021_b

演習 ■ 文字列の反転 (JOI 2019/2020 一次予選 (第 2 回) 問題 B)

https://atcoder.jp/contests/joi2020yo1b/tasks/joi2020_yo1b_b

楽勝楽勝という選ばれし民は④および⑦を進めましょう。

④ 13:50～14:40『リストの使い方を確認しよう』

C++ 文法の確認

リストを扱う場合は `vector` を使おう！

```
使用例： vector<int> vec; // int 型のリスト，配列変数の宣言
         vec = {0, 2, 3, 4, 6}; // リストへの初期値の代入
         for(i=0; i<vec.size(); i++) {} // 要素ごとの処理
         if (vec.at(i) == hoge hoge) { // 要素へのアクセス
```

下記の配列表記は C 言語その他の言語でも多く使われるが

- ・ C 言語の配列は要素数を保持できなかったり，
- ・ 範囲外の要素アクセスで正しくエラーを出せないなどの欠点がある

ので c++ では，使わない方がよい！

```
int vec[] = {0, 2, 3, 4, 6};
for(i=0;i<5;i++) { if (vec[i] == hoge hoge) ..... }
# [] 標記配列でも要素数を保持し，範囲外アクセスで正しくエラーが出せる言語もある (Java など)
```

`vector` の全要素のデータを加算する必要がある場合は，範囲 `for` 文を使ってもよい

```
vector<int> A(100);
// ここで A に 100 個のデータが入ったとする
for (int a: A) sum += a;
// 注意：この書き方だと，A の要素自体を書き換えることは不可能
```

たとえば，指定した個数のデータを取り込むのであれば

```
int N;
cin >> N;
vector<int> A(N);
for (int &a: A) cin >> a;
のように，&をつけると書き換え可能.
```

こちらも確認！

N - 1.13.配列 (https://atcoder.jp/contests/apg4b/tasks/APG4b_n)

R - 2.01.ループの書き方と範囲 `for` 文 (https://atcoder.jp/contests/apg4b/tasks/APG4b_r)

(応用編) U - 2.04.参照 (https://atcoder.jp/contests/apg4b/tasks/APG4b_u)

例題 ■ 共通要素（JOI 2020/2021 一次予選（第 1 回）問題 C）

https://atcoder.jp/contests/joi2021yo1a/tasks/joi2021_yo1a_c

入力：N M A(長さ N) B(長さ M)

出力：A と B の両方に存在する整数をすべて昇順で出力

ポイント、A と B の「両方」にあるという部分をどう考えるか。二重ループで比較する方法もあるが効率が悪いため、存在する・しないを管理するリスト `existA` と `existB` を準備し、A にある数字を `existA` に、B にある数字を `existB` に記録したあと両方にあるものだけを出力するのが良さそう。あとで A と B の中身は使わないので、`vector` で A,B を宣言する必要もなさそう！

`existA,B` は、100 個ではなく、101 個の要素を持つように宣言しよう！（1～100 の数字をカウントしたいが、配列・リストは要素番号が 0 から始まるので、100 個で宣言してしまうと、0～99 しか使えず 100 があるかどうかの判定に使えない）

```
// 多分こんな感じ
#include<bits/stdc++.h>
using namespace std;

int main() {
    int N,M,i,num;
    cin >> N >> M;
    vector<int> existA(101,0); // 101 個の要素を 0 で初期化
    vector<int> existB(101,0); // 101 個の要素を 0 で初期化
    for(i=0;i<N;i++) {
        cin >> num;
        existA.at(num) = 1;
    }
    for(i=0;i<M;i++){
        cin >> num;
        existB.at(num) = 1;
    }
    for(i=1; i<=100; i++){
        if (existA.at(i) + existB.at(i) == 2) {
            cout << i << endl;
        }
    }
}
```

例題 ■ ピアノコンクール(JOIG 2021/2022 本選 問題 A)

https://atcoder.jp/contests/joig2022-open/tasks/joig2022_a

入力 N A(N 個)

出力 A の総和-A の最大値-A の最小値 を求める

最小値, 最大値はどうやって求めよう

方法 1: 典型的方法を使う. 仮最小値, 仮最大値を先頭の値で行い, 2 つ目以降一個ずつ見て必要であれば更新する方法

```
int N,min,max,num,sum,i;
cin >> N;
cin >> num;
min = max = sum = num; // 複数の変数に同じ値をいれるときに使える構文
for(i=1; i<N; i++){
    cin >> num;
    if (min > num) min = num; // 仮最小値よりも小さければ更新
    if (max < num) max = num; // 仮最大値よりも小さければ更新
    sum += num;
}
```

方法 2: 最大値, 最小値を求める標準ライブラリを使う.

```
int min,max,sum,i;
cin >> N;
vector<int> A(N);
for(i=0;i<N;i++) { cin >> A.at(i); sum += A.at(i); }
min = *min_element(A.begin(), A.end());
max = *max_element(A.begin(), A.end());
```

方法 1 の方が `vector` を使わずに実装できるのでおすすめだが, C++ に `vector` 内の最大値, 最小値を計算するライブラリがあることは知っておくと良い

なお, 2 つの数の小さい方大きい方を求めるなら `min()`, `max()` を使えばいい.

例えば, A,B,C 3 つの数の最小値を求めるなら

`min(A, min(B, C))` とすれば求まる.

`min({A,B,C})` でも可能

```
// 多分こんな感じ：方法 1
#include <bits/stdc++.h>
using namespace std;

int main() {
    int N,min,max,sum,num,i;
    cin >> N;
    cin >> num;
    min=max=sum=num;
    for(i=1; i<N; i++) {
        cin >> num;
        if (min > num) min = num;
        if (max < num) max = num;
        sum += num;
    }
    cout << sum - min - max << endl;
}
```

```
// 多分こんな感じ：方法 2
#include <bits/stdc++.h>
using namespace std;

int main() {
    int N,min,max,sum=0,i;
    cin >> N;
    vector<int> A(N);
    for(i=0; i<N; i++) {
        cin >> A.at(i);
        sum += A.at(i);
    }
    min = *min_element(A.begin(), A.end());
    max = *max_element(A.begin(), A.end());
    cout << sum - min - max << endl;
}
```

以下の課題に挑戦しよう！

演習 ■ 分割 (JOI 2020/2021 一次予選 (第 2 回) 問題 C)

https://atcoder.jp/contests/joi2021yo1b/tasks/joi2021_yo1b_c

演習 ■ 最頻値 (JOI 2019/2020 一次予選 (第 2 回) 問題 C)

https://atcoder.jp/contests/joi2020yo1b/tasks/joi2020_yo1b_c

終わったら以下の課題にもチャレンジ！

演習 ■ ボールの移動 (JOI 2021/2022 一次予選 (第 3 回) 問題 D)

https://atcoder.jp/contests/joi2022yo1c/tasks/joi2022_yo1c_d

演習 ■ 希少な数 (JOI 2021/2022 一次予選 (第 2 回) 問題 D)

https://atcoder.jp/contests/joi2022yo1b/tasks/joi2022_yo1b_d

問題が簡単すぎる！という選ばれし人は、⑦の演習を進めましょう！

※ ⑤にはまだ手を付けないように！

⑤ 15:00～16:00『JOI2023/2024 一次予選にバーチャル参加しよう』

それでは、15:05～15:55 の 50 分の時間を使い、JOI2023/2024 一次予選にバーチャル参加してみましよう。

<https://atcoder.jp/contests/joi2024yo1a>

を開き



バーチャル参加

をクリックしたあと、下記の 4 つの問題を 50 分でできる限り解いてみてください。

予選バーチャル体験演習 ■ 果物 (JOI 2023/2024 一次予選 (第 1 回) 問題 A)

https://atcoder.jp/contests/joi2024yo1a/tasks/joi2024_yo1a_a

予選バーチャル体験演習 ■ 和の判定 (JOI 2023/2024 一次予選 (第 1 回) 問題 B)

https://atcoder.jp/contests/joi2024yo1a/tasks/joi2024_yo1a_b

予選バーチャル体験演習 ■ ハミング距離 (JOI 2023/2024 一次予選 (第 1 回) 問題 C)

https://atcoder.jp/contests/joi2024yo1a/tasks/joi2024_yo1a_c

予選バーチャル体験演習 ■ 現れている数字 (JOI 2023/2024 一次予選 (第 1 回) 問題 D)

https://atcoder.jp/contests/joi2024yo1a/tasks/joi2024_yo1a_d

時間内に終わった人は、⑦の問題集を実施しましょう

⑥ 16:20～16:30『まとめ&次回初中級編に向けて』

2 日目にむけて、

<https://atcoder.jp/contests/apg4b>

こちらを復習・予習しておくことをおすすめします。

また、ほぼ毎週実施されている abc コンテスト (AtCoder Beginner Contest)

<https://atcoder.jp/contests/archive?ratedType=1&category=0&keyword=>

に積極的にリアルタイム参加することもいいでしょう。

レギオ講習会用のコンテストも設定してあります。他会場の参加者も参加するかもしれませんのでこちらでもチャレンジしてみてください。

<https://atcoder.jp/contests/regio-workbook/>

⑦ 課題が簡単すぎて物足りない人向け課題集

https://ryamamoto9020.github.io/prog_seminar_website2025/first.html

の下部に問題を掲載しております。ここまでの問題で物足りない方は取り組んでみてください。

int の Sort について

```
vector<int> A(N);
for (int &a: A) cin >> a;

sort(A.begin(), A.end()); // 小さい順に並べ替える
sort(A.rbegin(), A.rend()); // 大きい順に並べ替える
```

char の Sort について

```
string str;
cin >> str;
// vector<char> str(N); // これでも OK
// rep(i,0,N) cin >> str.at(i); // これでも OK

sort(str.begin(), str.end()); // 中の文字を辞書順に並び替える
```

string の Sort について

```
vector<string> strList;
rep(i,0,N) cin >> strList.at(i);

sort(strList.begin(), strList.end()); // 文字列を辞書順に並び替える
```