

AtCoder Beginners Contest (ABC)

C問題までの基本的なC++コード集

テンプレート

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

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);

    // ここにコードを書く

    return 0;
}
```

大量の入出力を扱うときに、上記の高速化のおまじないを入れることでプログラムの実行速度が向上します。

入出力

単一の入力

```
int n;
cin >> n;
```

1つの整数を入力として受け取ります。

複数の入力

```
int a, b;
cin >> a >> b;
```

複数の変数をまとめて入力できます。

ループを使った入力

```
int n;
cin >> n;
vector<int> a(n);
for (int i = 0; i < n; ++i) {
    cin >> a[i];
}
```

n個の整数を入力して、ベクトルaに格納します。

出力

```
cout << "Hello, World!" << endl;
cout << a << " " << b << endl;
```

標準出力に文字列や変数を出力します。endlは改行を意味します。

条件分岐 (if文)

```
if (a > b) {
    cout << "aはbより大きい" << endl;
} else if (a < b) {
    cout << "aはbより小さい" << endl;
} else {
    cout << "aはbと等しい" << endl;
}
```

if, else if, elseを用いて条件に応じた処理を実行します。

ループ処理 (for, while)

```
int sum = 0;
for (int i = 0; i < 10; ++i) {
    sum += i;
}
```

for文: 繰り返し処理。例: 0から9までの整数を加算する。結果: 合計 = 45

```
int n = 10;
while (n > 0) {
    cout << n << endl;
    n--;
}
```

while文: 条件が成り立つ間繰り返す。例: 10から1までを出力する。

配列とベクトル

```
int arr[5] = {1, 2, 3, 4, 5};
```

```
vector<int> vec = {10, 20, 30};
vec.push_back(40);
cout << "ベクトルのサイズ: " << vec.size() << endl;
```

配列は固定長、vectorは可変長の配列。push_backで要素を追加可能。

文字列の操作

```
string s = "atcoder";
cout << "文字列の長さ: " << s.size() << endl;
cout << "先頭の文字: " << s[0] << endl;
```

stringは文字列型。配列のように[]で文字へアクセス可能。

最小値・最大値の計算

```
int a = 10, b = 20;
int min_val = min(a, b);
int max_val = max(a, b);
cout << "最小値: " << min_val << ", 最大値: " << max_val << endl;
```

min/max関数を使うと、2つの値の最小値・最大値を簡単に求められる。

AtCoderやパソコン甲子園でよく使うテクニック

```
// 配列全体をソート
sort(a.begin(), a.end()); // 昇順ソート

// 文字列のソート
string s = "dcba";
sort(s.begin(), s.end()); // -> "abcd"

// 累積和 (部分和の高速計算)
vector<int> prefix(n+1, 0);
for (int i = 0; i < n; ++i) prefix[i+1] = prefix[i] + a[i];

// ペアやタプルの利用
vector<pair<int,int>> vp;
vp.push_back({1, 2});

// 範囲for文
for (int x : a) cout << x << " ";
```

sort, 累積和, pair/tuple, 範囲for文はC問題レベルで頻出です。