

# 競技プログラミング向けのプログラミング言語 アルバト **Albat**

競技プログラミング（きょうぎプログラミング、英：Competitive programming、略称：競プロ）とは、解くべきプログラミング課題が与えられ、プログラムを作成し、解くまでにかかった時間や正解数などを競い合う競技

開発駆動コース川合ゼミ  
24Dk 鳥海翔一

Albatの特徴：競技プログラミングにおける3つの利点

## C++へのトランスパイル

## 標準ライブラリ

## 高速入出力

AlbatはC++コードにトランスパイルされるため、高速で効率的なプログラミングが可能です。

競技プログラミングでよく使われるデータ構造やアルゴリズムが標準で利用できます。

高速な入出力処理により、時間制約の厳しい競技プログラムでも優れた性能を発揮します。

↑ この2つがあるとなんで嬉しいの？ ↑



## モチベーション

IO	Time (ms)
cin, cout	1211
scanf, printf	364
自作FastIO	122

64bit整数入出力の速度比較

競技プログラミングにはC++が向いている※けれど  
・ 自前のライブラリやマクロが複雑で冗長だよ  
・ using namespace std; とか書きたくないよ



※コンパイル言語の高速性と豊富な標準ライブラリ、C++ユーザーの多さなどの理由によりC++が向いているとされています。

## 自前のライブラリやマクロが複雑

自作のデータ構造やライブラリ、マクロをすべて記述していると複雑に... 逐一ライブラリを貼ったりするのも、時間が大事な競技プログラミングにおいてはネックに...

## using namespace std; とか

競技プログラミング特有の #include <bits/stdc++.h> とか using namespace std; みたいな おまじないと説明されるような作法を省略したい



## コード例

### 例題

整数N,Kと長さNの整数列P,Qが与えられます。  
PとQから1つずつの整数を選び、その合計をKにすることはできますか？  
答えを出力するプログラムを作成してください。

## 説明

### 入出力

@変数1, 変数2 ... → 入力としてFastIOが呼び出されます。vectorとかstringの入力も行えます。  
\$(hoge) → 出力を行います。\$\$にすることで改行出力になります。

### ライブラリ

yesno(arg1, str1, str2) → arg1のTrue, Falseに応じてstr1かstr2が返されます。デフォルトではyesかnoを返す。

### その他(展開)

vi: vector<int>, fore(x, X): for(auto &x: X)

```
1 #include <iostream>
2 using namespace std;
3 int N, K;
4 int P[109], Q[109];
5 bool ok = false;
6 int main() {
7     cin >> N >> K;
8     for (int i = 1; i <= N; i++) cin >> P[i];
9     for (int i = 1; i <= N; i++) cin >> Q[i];
10    for (int x = 1; x <= N; x++) {
11        for (int y = 1; y <= N; y++) {
12            if (P[x] + Q[y] == K) ok = true;
13        }
14    }
15    if (ok == true) cout << "Yes" << endl;
16    else cout << "No" << endl;
17    return 0; }
```

C++  
17行

```
1 N, K = map(int, input().split())
2 P = list(map(int, input().split()))
3 Q = list(map(int, input().split()))
4 ok = False
5
6 for x in range(N):
7     for y in range(N):
8         if P[x] + Q[y] == K:
9             ok = True
10
11 if ok == True:
12     print("Yes")
13 else:
14     print("No")
```

Python  
14行

```
1 int@N,K;
2 vi@(P(N),Q(N));
3 int ok = 0;
4 fore(p,P) fore(q,Q) ok |= (p + q == K);
5 $$$(yesno(ok));
```

Albat

5行!!!

↑これが

C++に翻訳されると...

こうなる→

```
me@PC124-D03:~/a$ tail -n 30 a.cpp
#define rd io.read
string yesno(int ok, string yes = "Yes", string no = "No"){
    return ok ? yes : no;
}
int main(){
    int N, K;
    rd(N);
    rd(K);
    vector<int> P(N), Q(N);
    rd(P);
    rd(Q);
    int ok = 0;
    for(auto &p : P){
        for(auto &q : Q){
            ok |= (p + q == K);
        }
    }
    wln(yesno(ok));
    return 0;
}
//---endl---
// --- original ---
// int@N,K;
// vi@(P(N),Q(N));
// int ok = 0;
// fore(p,P) fore(q,Q) ok |= (p + q == K);
// $$$(yesno(ok));
```

## 機能

### ・競技プログラミングするときを使うような機能

→ 演算子 +α, 条件分岐, 繰り返し, 関数定義, ラムダ式 とか

#### Albat独自の演算子

```
1 //演算子の例
2 int a = 4, b = 6;
3
4 //べき乗
5 $$$(a ** b); // 4096
6
7 //2点間のユークリッド距離を求める
8 int x1 = 1, y1 = 2, x2 = 3, y2 = 5;
9 $$$(sqrt((x1 - x2) ** 2 + (y1 - y2) ** 2)); //3.6055
```

#### 11 //余り切り上げの割り算

```
12 $$$(a /+ b); // 1
```

```
14 //負の数を正の数で割った余りを非負で返す
15 $$$(a - b) % b); // -2
16 $$$(a - b) %+ b); // 4
17
18 //ソングバッファとしてarrを扱うとき
19 vector<int> arr = {1,2,3,4,5,6};
20 $$$(arr[(a-b)%b]); //a[-2]は未定義動作
21 $$$(arr[(a-b)%+b]); //a[4] = 5
```

#### 競技プログラミングで使いそうなデータ構造

```
1 //UnionFind
2 UnionFind uf(100);
3
4 //SegTree
5 int op(int a, int b) { return max(a, b); }
6 int e() { return -1; }
7 vector<int> A(100, 0);
8 _SegTree<int,op,e> seg(A);
9
10 //modint
11 modint a = 1;
12
13 //Rexを高速で求めるライブラリ
14 MexSet s;
15 etc ...
```

#### 競技プログラミングでよく見るようなマクロ(展開されます)

```
INFI // 2**30 - 2**15 = 1073709056
INFL // 2**62 - 2**31 = 4611686016279904256
MOD // デフォルトでは1000000009
ll // long long
vi // vector<int>
P // pair<int, int>
rep(i,n) //for(int i=0;i<n;i++)
etc ...
```

## 今後の展望

### ・今後もこの言語を使ってコンテストに挑戦!!!

→ 実際に問題を解くことで欲しい機能を見つけていきたい

### ・gcc以外にも対応

→ 現状だとgcc以外のコンパイラだと動かないので対応させたい

### ・競技プログラミング以外にも有用な使い方を見つける

→ 競技プログラミング以外の用途でも日常的に使えると嬉しい

### ・演算子記号の定義

→ 言語に組み込むほどではないけど演算子扱えたら嬉しいときに演算子を定義できるようにしたい