

---

# 05 配列

---

## 目標

- 配列のイメージを掴む.
- 配列を使うことができるようになる.

## 例題

### 【最頻値】

出典：JOI 2019/2020 一次予選（第 2 回）C

### 問題文

長さ  $N$  の数列  $A_1, A_2, \dots, A_N$  が与えられる. この数列の各項は 1 以上  $M$  以下の整数である.

長さ  $M$  の新たな数列  $B_1, B_2, \dots, B_M$  を以下のように定義する.

- 各  $j$  ( $1 \leq j \leq M$ ) に対して,  $B_j$  の値は  $A_i = j$  を満たす整数  $i$  ( $1 \leq i \leq N$ ) の個数に等しい.

$B_1, B_2, \dots, B_M$  の最大値を求めよ.

### 制約

- $1 \leq N \leq 100$ .
- $1 \leq M \leq 100$ .
- $1 \leq A_i \leq M$  ( $1 \leq i \leq N$ ).

### 入力

入力は以下の形式で標準入力から与えられる.

```
N M
A1 A2 … AN
```

### 出力

$B_1, B_2, \dots, B_M$  の最大値を 1 行で出力せよ.

### 入力例 1

```
4 3
1 1 2 3
```

### 出力例 1

```
2
```

数列  $B_1, B_2, B_3$  は 2,1,1 となり、これらの最大値は 2 である。

### 入力例 2

```
6 5
3 3 2 1 2 3
```

### 出力例 2

```
3
```

### ■ 考察

- 配列の各要素  $B[j]$  を定義通り順に求めてもいいが、少し複雑になる。
- $B[j]$  をあらかじめ 0 で初期化しておいて、 $A[i]$  を順に見ながら  $B[A[i]-1]$  に 1 加算すればよい。

## 解答例

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int N, M, A[100], B[100];
    cin >> N >> M;
    for(int i=0; i<N; i++){
        cin >> A[i];
    }
    for(int i=0; i<M; i++){
        B[i] = 0;
    }
    for(int i=0; i<N; i++){
        B[A[i]-1]++;
    }
    int ans = 0;
    for(int i=0; i<M; i++){
        if(ans < B[i]){
            ans = B[i];
        }
    }
    cout << ans << endl;
}
```

## 解説

### ① 配列の定義

複数の変数をひとまとめにしたものが **配列** である。

```
int A[100];
```

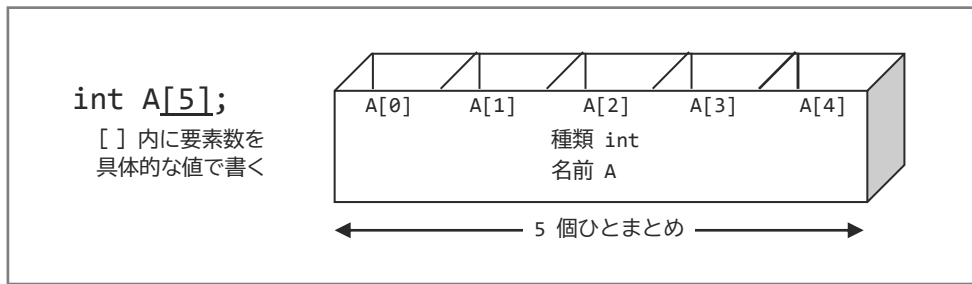
上のように定義したとき、int 型変数を 100 個まとめた配列を作り、A という名前をつけるという意味になる。このとき、まとめる個数を **要素数** と呼ぶ。要素数を変数で指定することはできないので、問題の制約から必要になりそうな最大の個数を確保しておく。

### ② 配列のアクセス

100 個の要素からなる配列には、先頭の要素から順に 0 から 99 の番号が振られている。これを使って、文字列のときと同じように、各要素にアクセスする。

```
for(int i=0; i<N; i++){
    cin >> A[i];
}
```

この場合、N 個の整数を入力から受け取り、配列の先頭から順に代入している。



### ③ 配列のポイント

例題の B のように、各値がある配列に何個含まれるのかを求めるのは JOI 一次予選で頻出である。

また、問題で与えられた数列とは別に配列を定義しなければならない問題も多い。配列の各要素が何を意味するのか、よく考えるようにしよう。

## 表現

(1) 標準入力から整数  $N$  と正整数列  $A_1, \dots, A_N$  を順に受け取り,  $A_1, \dots, A_N$  のうち偶数のものだけを入力する

```
int N, A[100000];
cin >> N;
for(int i=0; i<N; i++){
    cin >> A[i];
}
for(int i=0; i<N; i++){
    if(A[i] % 2 == 0){
        cout << A[i] << endl;
    }
}
```

(2) 標準入力から整数  $N$  と整数列  $A_1, \dots, A_N$  を順に受け取り,  $A_1, \dots, A_N$  を逆順で出力する

```
int N, A[100000];
cin >> N;
for(int i=0; i<N; i++){
    cin >> A[i];
}
for(int i=0; i<N; i++){
    cout << A[N-1-i] << endl;
}
```

(3) 標準入力から整数  $N$  と整数列  $A_1, \dots, A_N$ , 整数  $l, r$  ( $1 \leq l \leq r \leq N$ ) を順に受け取り,  $A_l + \dots + A_r$  を出力する

```
int N, A[100000], l, r;
cin >> N;
for(int i=0; i<N; i++){
    cin >> A[i];
}
cin >> l >> r;
int sum = 0;
for(int i=l-1; i<r; i++){
    sum += A[i];
}
cout << sum << endl;
```

(4) 標準入力から整数  $N$  を受け取り、フィボナッチ数列の第  $N$  項を出力する  
ただし、フィボナッチ数列の第 1 項を 1、第 2 項目を 1 とする

```
int N, fib[30];
cin >> N;
fib[0] = 1;
fib[1] = 1;
for(int i=2; i<N; i++){
    fib[i] = fib[i-2] + fib[i-1];
}
cout << fib[N-1] << endl;
```

※ フィボナッチ数列は 1 つ前と 2 つ前の項を足して次の項を得る数列である。上のプログラムは、1,1,2,3,5,8,13,... と出力する。

## ■ 類題

- 分割 (JOI 2020/2021 一次予選 (第 2 回) C)
- 未提出者は誰だ (JOI 2006/2007 予選 B)
- Counting Roads (AtCoder Beginner Contest 061 B)
- Kagami Mochi (AtCoder Beginner Contest 085 B)
- 共通要素 (JOI 2020/2021 一次予選 (第 1 回) C)