

引数と配列

05

配列は、同じ種類のデータをまとめて扱うときに便利だということはすでに学習しました。ここでは、この配列のすべてを引数として関数に渡す方法について学びます。関数を渡す場合との違いを把握しましょう。

基本例
4

int型の配列に格納された数値の中から最小値を求めます。ただし、配列の変更のうちの指定した範囲における最小値がどこにあるかを求めます。配列と最初の要素、それ以降の範囲の値を指定して、最小値の位置を求める関数を作成し、呼び出してみましょう。

実行結果

```
22 58 36 42 85 72 19 28 66 51
関数: 2
値数: 4
最小値はdata[2]の36
```

← 標準入力から入力された値

学習

STEP1

配列を引数に渡す

● 基本例のやり直しには、関数を実行する前に、関数で配列を利用するわけではないので、関数を実行前には書きません。

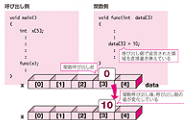
引数として値を渡すときには、関数側にある引数に、渡された値が代入されることは、前節で学習しました。だから、関数側で引数の値をいくらか変更しても、呼び出し側には何の影響もありませんでした。しかし、引数として配列を渡したときは、状況は少し異なります。配列の場合には、実引数は関数側にコピーされるのではなく、関数側から呼び出し側の配列を直接見ることになります。

図24 配列を引数とする



そのため、関数側で配列型の引数を変更すると、呼び出し側にも影響し、関数が終了しても、配列は消滅しません。

図25 関数側の処理が呼び出し側に影響



● このように値を渡すのではなく、直接参照する引数の受け渡し方法もCall by Referenceといえます。

配列型の引数は、実引数として渡された配列を参照しますから、関数側では、同じ名前(仮引数名)でも実際に参照している場所は、実引数によって異なります。

図26 実引数として指定された配列を参照

