

No.5 A足すB問題

入力の受取

プログラムの実行時にデータの入力を受け取る方法を見ていきましょう。「入力機能」を使うことにより、プログラムを書き換えなくても様々な計算を行えるようになります。

入力を受け取るには`read*`, を使います。

次のプログラムは、入力として受け取った数値を10倍にして出力するプログラムです。

```
program main
  implicit none
  integer(4):: a

  read*, a
  print*, 10*a
end program
```

入力1

5

実行結果1

50

実際に入力を受け取っているのは次の部分です。

```
read*, a
```

整数以外のデータの入力

整数以外のデータを受け取りたいときは、そのデータに合わせた方の変数を使います。

```
program main
  implicit none
  real(4):: b          ! ←実数型の変数として宣言
  character(10):: c    ! ←長さ10の文字型の変数として宣言
  real(8):: d          ! ←倍精度実数の変数として宣言

  read*, b
```

```
      read*, c  
      read*, d  
  
      print*, b,c,d  
end program
```

入力2

```
4.1  
ojama simasu  
5d-2
```

5d-2とはFortranの倍精度実数の表記の仕方で、 5×10^{-2} を表します。

実行結果2

```
4.09999990      ojava      5.00000000000000003E-002
```

実数型は二進数で小数点の桁を表しているため、4.1で入力したはずの数値が4.09999990と誤差によって異なる値になります。このように、実数型は常に誤差をはらんでいるため、整数型で良い場面は整数型で扱ってあげてください。

入力の二行目のojama simasuは空白で区切られ、前半のojama部分のみがcに入力されています。

入力3

```
4.1  
ojama_simasu  
5d-2
```

実行結果3

```
4.09999990      ojama_sima  5.00000000000000003E-002
```

入力の二行目をojama_simasuとアンダースコアで繋いだため、区切られることなくcに入力されます。しかし、cは長さ10までの文字列しか扱えません。よって入力の10文字目以降は無視されて、ojama_simaの10文字がcに入力されています。

つなげて入力

先程の例にあるように、print文は、print*, b,c,dとすることでb,c,dを一行にまとめて出力させることができます。

同様に入力が行に複数ある場合は、スペースか改行で区切られていれば分解して入力してくれます。

```
program main
  implicit none
  integer(4):: a,b,c

  read*, a,b,c
  print*, a*b*c
end program
```

入力4

```
2 3
4
```

実行結果4

```
24
```

aに2、bに3、cに4が入力されています。

問題文

2つの整数 A, B が与えられます。 $A + B$ の計算結果を出力してください。

```
program main
  implicit none

  !ここにプログラムを追記
end program
```

制約

- $0 \leq A, B \leq 100$
- A, B は整数

入力

入力は次の形式で標準入力から与えてください。

```
A B
```

出力

$A + B$ の計算結果を出力してください。

入力例1

1 2

出力例1

3

入力例2

100 99

出力例2

199