プログラム表記の例示

2025 年度入試版

本試験の問、および解答においてプログラムを記述するために用いる記法について説明する。

演算子

数値に対して算術演算を行うことができる。演算子は以下の通り。なお、*、/ と % の計算が + や − よりも先に行われる。

+	加算(足し算)	−	減算(引き算)
*	乗算(掛け算)	/	除算(割り算)
%	剰余算(割り算の余り)。(例: 7%3 は 1)		

数値同士、文字列同士を比較できる。結果は真または偽である。

A == B	AとBの値が等しい。		
A != B	AとBの値が等しくない。		

数値の大小を比較できる。結果は真または偽である。

A <= B	AがB以下	A < B	AがBより小さい
A >= B	AがB以上	A > B	AがBより大きい

複数の条件を組み合わせたり、ある条件を否定するために以下の論理演算子を用いることができる。

条件1 **AND** 条件2　　条件1 **OR** 条件2　　**!**条件

算術演算と比較演算では、算術演算が先に計算される。また、ORよりもAND、ANDよりも！が優先される。式の中で（　）を使い、計算の順序を示すことができる。

文

```
count = count + 1
```
変数に式の値を代入する。左の例は変数の値を 1 増やす。

```
for i = 0 to N-1
  table[i] = 0
end
```
変数の値を0から(N-1)まで1ずつ増やして繰り返す。

```
while a[n] == b[n]
  n = n + 1
end
```
条件が成り立つ間、繰り返しを実行する。

```
while a[n] == b[n]
  if a[n] == 0
    break
  end
  n = n + 1
end
```
if文の条件が成り立てば、while文の繰り返しを打ち切る(同じように、for文の繰り返しも打ち切ることができる)。

```
if kekka != 0
  print("当選")
end
```
条件が成り立つときに実行する。

```
if kion < 30
  print("実施")
else
  print("中止")
end
```
条件が成り立つかどうかで、実行することを変える。

```
if tokuten >= 80
  print("優秀")
elif tokuten >= 60
  print("合格")
else
  print("追試")
end
```
複数の条件を順番に調べて実行することを変える。

```
return gokei
```
関数の実行を終了して値を呼び出し側へ戻す。

```
return
```
関数の実行を終了する(戻り値の必要ない場合)。

```
print("合計:", w + i )
```
値や文字列を表示する。

変数と配列: 変数(または配列)は、関数の内部で宣言した場合、その関数でのみ利用可能な変数(ローカル変数)となり、関数の外部で宣言するとどこからでも利用可能な変数(グローバル変数)となる。

```
var i, j
var total = 0
```
変数を宣言する。宣言と同時に初期値を指定できるが、指定がない変数の初期値は不定(意味のない値)である。

```
var table[10]
```
配列を宣言する。上の例ではtable[0]からtable[9]まで10個の要素が用意される。

```
var s[] = {-1, 0, 1}
```
初期値を指定した配列を宣言する。上の例ではs[0]に−1、s[1]に0、s[2]に1が設定される。

関数

```
func add(a, b)
  var sum
  sum = a + b
  return sum
end
```
関数(サブルーチン)を定義する。戻り値のある関数は return文で値を指定する。左の関数は次の例のように呼び出せる。

```
total = add(m, 500)
```

```
func show(t)
  print("答:", t)
end
```
戻り値のない関数は return文を省略できる。左の関数は次の例のように呼び出せる。

```
show(n * 2)
```

コメント: プログラムの記述中に # が現れた場合、そこから行末までの文字列はコメント(注釈)とみなし、実行されない。ただし、文字列の中に現れた # はコメントとしては扱わない。

プログラム例1: 買い物の合計額を計算する関数の定義

買い物の項目数が n、配列 price, amount に買った品物の単価と個数が格納されているとする。合計金額が 2000円以上なら送料が無料になる。変数deliveryはグローバル変数である。

```
var delivery = 500  # 送料

func shopping(price, amount, n)
  var pay = 0
  var i

  for i = 0 to n - 1
    pay = pay + price[i] * amount[i]
  end
  if pay >= 2000    # 合計が2000円以上
    return pay
  end
  return pay + delivery
end
```

プログラム例2: 三角形の種類を調べる関数の定義

```
func triangle(x, y, z)
# 引数は3辺の長さ。ただし、x ≧ y ≧ z とする。
  if x >= y + z
    print("三角形ではない")
  elif x == y OR y == z
    if x == z
      print("正三角形")
    else
      print("二等辺三角形")
    end
  else
    print("三角形")
  end
end
```

本表記方法は既存のプログラミング言語とは異なる疑似言語によるものである。