

=====

日商プログラミング検定 STANDARD (Python) 類似問題 セット2

解答・解説

=====

【知識科目】第1問(知識4択:20問)

1. 正解: ① _

解説: Python の対話モードでは、直前に評価した式の結果が特殊変数「_」に自動的に保存される。ans や last、# は特殊変数ではない。

2. 正解: ③ 1024

解説: 「**」はべき乗演算子。2の10乗 = 1024。

3. 正解: ② # これはコメントです

解説: Python の行コメントは「#」から行末までがコメント扱いになる。
「//」はC言語系、「/* */」もC言語系のコメント記法であり
Pythonでは使用できない(構文エラーになる)。

4. 正解: ① /

解説: 「/」は実数(浮動小数点数)の商を返す演算子。
「//」は切り捨て除算(商を整数に丸める)、「%」は剰余を返す。

5. 正解: ③ 1total

解説: Python の変数名は数字から始めることができない。
1total は変数名として使用不可。total_1、_sum、Total はいずれも
使用可能(Total と total_1 は大文字小文字を区別する別の名前として
有効)。

6. 正解: ② A-B

解説: print() の sep 引数は、複数の値を表示するときの区切り文字を
指定する。sep='-' なので、'A' と 'B' の間が「-」で区切られ
「A-B」と表示される。

7. 正解: ① (3)と(4)

解説:

- (1) not True → False
 - (2) True and False → False
 - (3) not False → True
 - (4) False or True → True
- True になるのは (3) と (4)。

8. 正解: ③ 24 >> 2

解説: シフト演算はビットをずらす操作。

- ① 2 << 1 = 4 (2進数 10 を左に1桁 → 100 = 4)
- ② 2 << 2 = 8 (10 を左に2桁 → 1000 = 8)
- ③ 24 >> 2 = 6 (24 = 11000 を右に2桁 → 110 = 6)

④ $24 \gg 3 = 3$ (11000 を右に3桁 $\rightarrow 11 = 3$)
結果が6になるのは③。

9. 正解: ③ C

解説: $x=5, y=10$ のとき、 $x > y$ は False、 $x == y$ も False (5と10は等しくない)なので、else節の `print('C')` が実行される。

10. 正解: ③

解説: 「 x が偶数のとき『偶数』と表示、奇数なら何もしない」という仕様に対して、③「`if x % 2 == 1: print('偶数')`」は x が奇数のときに『偶数』と表示してしまい、条件が逆になっているため誤り。
①②④はいずれも「偶数のときだけ『偶数』と表示する」という同じ動作になる。

11. 正解: ② 5 (doneは表示されない)

解説: `while...else` の else節は、ループがbreakされずに正常終了したときのみ実行される。
 $x=20 \rightarrow 15(15!=5) \rightarrow 10(10!=5) \rightarrow 5(5==5, \text{breakする!})$
breakが実行されたため else節 (`print('done')`) は実行されない。
ループを抜けた後、`print(x)` が実行され、 $x=5$ が表示される。

12. 正解: ① 5 4 3 2 1

解説: `range(5, 0, -1)` は 5 から始まり 0 未満 (0は含まない) まで -1ずつ減少するので 5,4,3,2,1 を生成する。`end=' '` なので改行せず半角スペース区切りで一行に表示される。

13. 正解: ② [1, 3, 5]

解説: `L1[::2]` は「先頭から末尾まで、2つおきに」要素を取り出すスライス。インデックス0,2,4の要素、つまり1,3,5が取り出される。

14. 正解: ② [3, 2, 1]

解説: `reverse()` はリストの並び順をその場 (in-place) で逆転させるメソッド。戻り値は None だが、`L1` 自体の中身が [3,2,1] に書き換わるため、`print(L1)` では [3, 2, 1] が表示される。

15. 正解: ② `data.index(5)`

解説: `index(値)` はリストの中から指定した値を検索し、最初に見つけた位置 (インデックス) を返す。`find()` や `search()` というリストメソッドはPythonの標準リストには存在しない。

16. 正解: ③ Tue

解説: タプルもリストと同様に添字 (インデックス) でアクセスできる。`S[1]` は2番目の要素 'Tue' を返す。

17. 正解: ② `menu['サラダ'] = 300`

解説: 辞書に新しい要素を追加・更新するには「辞書名[キー] = 値」の形で代入する。`add()` や `insert()` という辞書のメソッドは存在しない。

18. 正解: ② {2, 3}

解説: 「&」は集合の積集合(共通部分)を求める演算子。

A={1,2,3}, B={2,3,4} の共通部分は {2, 3}。

19. 正解: ② False

解説: 「in」演算子は辞書に対して使うと「キーが存在するか」を調べる。d には 'x' と 'y' というキーしかなく 'z' は存在しないので False。

20. 正解: ③ S.count('a')

解説: count(値) は文字列やリストの中で、その値が出現する回数を数える。find() や index() は最初に見つかった「位置」を返すメソッドであり、回数を数えるものではない。

【知識科目】第2問(穴埋め:2問)

【問題1】要注意データ(低スコア)の検出

【1】正解: (2) mean - 15

解説: 「平均点より15点以上低い」データを判定する基準は「平均値 - 15」。

【2】正解: (3) x <= lower_limit

解説: 「15点以上低い」は「差が15以上(=15点低い場合も含む)」という意味なので、境界値ちょうど15点低いデータも要注意データに含める必要がある。そのため、単なる x < lower_limit(15点より低いデータのみ)ではなく、x <= lower_limit(15点以上低いデータすべて)が正しい条件になる。

【問題2】機械の生産数増加(1.2倍ずつ)

【1】正解: (3) 100

解説: 1時間目の生産数は100個であり、この時点の累計 total も100個になる。

【2】正解: (2) total + hourly

解説: 累計生産数は「これまでの累計」に「今回新たに生産した数(hourly)」を加算して更新する。

【3】正解: (3) n

解説: ループを抜けた時点の n(時間数)が「何時間目に超えたか」の答えになる。

【知識科目】第3問(読解:1問)

正解: ① 文字列 s が回文(前から読んでも後ろから読んでも同じ)かどうかを判定する。

解説: s の前半部分(range(len(s)//2))について、前から i 番目の文字

(s[i])と後ろから i 番目の文字 (s[len(s)-1-i])を1文字ずつ比較している。1か所でも一致しない文字があれば is_ok を False にすることで、回文かどうかを判定するプログラムになっている。
'racecar' は実際に回文なので、実行結果は True になる。

【実技科目】

【問題1】正解: if temp >= 30:

解説: 気温が30以上のとき「猛暑」と判定するので、条件式は「temp >= 30」。

【問題2】正解: 【1】x 【2】x

解説: 現在調べている値 x が、これまでの最大値 max_val より大きければ max_val を x に更新し、逆にこれまでの最小値 min_val より小さければ min_val を x に更新する。これをすべてのデータについて繰り返すことで、最終的にリスト全体の最大値・最小値が求まり、その差(レンジ)が計算できる。

【問題3】

【1】正解: (t - 1) // size

解説: 通学時間 t を「1引いてから15(size)で割った商」にすると、
t=1~15分は商0(0~15分の区分)、t=16~30分は商1
(16~30分の区分)というように、1から数える範囲を0始まりの
区分番号に変換できる。

【2】正解: dosu[idx] + 1

解説: 該当する時間帯の人数(度数)を1人分増やす。

【3】正解: i in range(5)

解説: 5つの時間帯すべてについて、それぞれの人数を順番に表示するため、i を 0~4 の範囲でループさせる。