サンプル問題

Ｐｙｔｈｏｎプログラミング能力認定試験

２　級

正答・解説

サーティファイ
情報処理能力認定委員会

＜解説＞

問１

(1)	(2)	(3)	(4)	(5)	(6)
ア	イ	ア	イ	ア	ア

(2) Python では，事前に変数の型を定義する必要がなく，変数に値を代入した時点で型が自動的に決まる。

(4) # を使ったコメントは，行の途中からでも挿入することができる。

問２

(7)	(8)	(9)	(10)	(11)
イ	ア	エ	ウ	ウ

(7) 「from モジュール名 import 関数名」の形でモジュールや関数などをインポートする。

(8) print 関数は，引数に出力する値を複数指定した場合，順番に同じ行に出力される。'end'は行末の文字列を指定するもので，end=','とした場合は，行末が','となる。なお，end を省略した場合は，デフォルトの end='¥n'（改行文字）が呼び出され，改行される（改行文字で使用している ¥ 記号は，環境によって表示が異なる場合がある）。

(9) min 関数は引数に指定されたリストやタプルなどの要素の中で最小の要素，または複数の引数の中で最小の引数の値を返す関数である。すなわち，min([6, 3, 5, 2, 7])はリストの最小要素である 2 を返す。したがって，print 関数の引数は 15 / 2 = 7.5 となる。なお，Python における除算は整数同士でも浮動小数点型となるため，注意が必要である。

(10) 関数から戻り値を返すには，return 文を使用する。

(11) 関数の内側で global 宣言を行わない場合はローカル変数となるため，関数の外側の同名の変数は別物として扱われる。

問３

(12)	(13)	(14)	(15)	(16)	(17)
ウ	イ	ウ	ア	ウ	ア

(12) リストに格納されている値とその要素番号の対応は次のようになっている。

要素番号	0	1	2	3	4	5	6	7	8
a	3	1	4	1	5	9	2	6	5

これにより a[4]の値は 5 なので，a[4] / 2 の値「2.5」が出力される。

(13) a[6]の値は 2 であり，「**」はべき乗を意味するので，2^3 + 3 の値「11」が出力される。

(14) シングルクォーテーション「'」で囲まれた部分は文字列として扱われるため，「No.」は文字列である。str 関数は引数に指定したオブジェクトを文字列として取得できるので，「str(a[5])」は a[5]の値「9」を文字列として取得する。文字列は「+」で結合することができるため，'No.'と'9'が文字列として結合され，「No.9」と出力される。

(15) ()内の式は優先的に計算されるため，7 – b[5]が優先的に計算され，-1 となる。「//」は除算の商の小数部分を切り捨て，整数部分（除算の結果を超えない最大の整数）を求める演算子である。したがって，-1 ÷ 3 = -0.3333…となり，これを超えない最大の整数「-1」が出力される。

(16) len 関数は，リストの要素数や文字列の長さなどを返す関数であり，「%」は剰余（除算の余り）を求める演算子である。b は要素番号 0～5 の計 6 個の要素を持つリストなので，それを 4 で割った余り「2」が出力される。

(17) b[2:4]はスライスであり，リスト b の要素番号 2 から 3 までの要素をリストとして取得する。したがって，「[1, 8.0]」が出力される。

問４

(18)	(19)	(20)	(21)	(22)	(23)	(24)
ア	エ	イ	ウ	イ	エ	イ

(18) if 文を使って，文字列の格納された変数 a，b の値を調べるプログラムである。

4 行目の「'サッカー' in a」で変数 a に文字列'サッカー'が含まれているかを，「'バスケット' in b」で変数 b に文字列'バスケット'が含まれているかを判定している。ここでは，a に'サッカー'が含まれているが，b に'バスケット'は含まれていないため，4 行目の条件式は False である。

6 行目の startswith は，引数に指定した文字列で始まるかどうかを判定する。a は'わ'から始まるため，条件式の結果は False である。なお，True だった場合の 7 行目の処理 replace は，一つ目の引数の文字列を二つ目の引数の文字列で置き換えるメソッドである。

8 行目の find は，引数に指定した文字列を探し出し，あればその文字列が現れた先頭文字の位置を返す。b のインデックス 4（0 から始まる）の文字は'は'であるため，条件式は True となる。したがって，9 行目の処理が実行され，「サッカーはスポーツです。」が表示される。

(19) if 文を使って変数 a，b，c が None であるかどうかの組合せで判別するプログラムである。

1 行目は変数 a をリスト型，2 行目は b を文字列型のデータとして定義している。また 3 行目は 1 == 2 という論理式の評価結果を変数 c に代入したものであり，その値は False となる。つまり，いずれの変数も None ではない。

5，7，9 行目の論理式のうち，True となるのは 9 行目のみである。したがって，10 行目が実行されて，「False []」と表示される。

(20) while を使った繰返し処理のプログラムである。

変数 a および counter を 0 で初期化し，while のブロックの中で a に counter の値を加算後，counter に 1 を加算している。この繰返し処理は，6 行目時点で a が 4 より大きいか，繰返し処理の先頭時点で counter が 6 より小さくなければ終了する。a と counter の値の推移は次の通りとなる。

a（4 行目）： 0 → 0 → 1 → 3 → 6

counter（5 行目）： 0 → 1 → 2 → 3 → 4

ここで a の値が 4 より大きいという条件を満たし，break 文によって終了する。したがって，「6 4」が表示される。

(21) for n in numbers で，リスト numbers に格納された要素を順に取り出して n に代入し，繰返し処理を行うプログラムである。

n % 2 == 0 で，n が偶数か奇数かを判定し，偶数であれば変数 ans に n を加算している。繰返し処理の終了後，ans *= 2 で ans を 2 倍した値を再代入している。リストに格納されている偶数は 2 と 4 であり，その合計は 6 であるから，「12」が表示される。

(22) find は，引数に指定した文字列が含まれる場合はその位置を番号で返し，その文字列が含まれない場合は-1 を返す。4，5 行目では，name_b に含まれる文字を一つずつ順番に取り出し，name_a に含まれる文字かどうかを判断している。つまり，strs は，name_b から name_a と共通する文字だけを抜き出して並べたものと等しくなる。したがって，「ttntna」が表示される。なお，ここで，大文字と小文字は区別されることに注意する必要がある。

(23) 1 行目で row = 3 で初期化されるため，for i in range(row)では，0 から 2（指定した数値-1）の整数の間で繰返し処理を行う。sp * (row - i)や mark * (i * 2 + 1)などは文字列と数値の乗算であるが，これは文字列をその数値分繰り返して結合することを意味する。また lr + center + lr も，文字列の結合を意味する。したがって，「ΔΔΔ*ΔΔΔ」「ΔΔ***ΔΔ」「Δ*****Δ」が 1 行ずつ順に出力される。

(24) プログラム内のある値やパターンに対して多くの条件がある場合に，match を使用するとコードの冗長性を減らすことができる。3 行目でリスト commands に格納されている要素を順に取り出して cmd に格納し，5 行目以降の case 文で cmd と指定した値が一致しているかを判定し，一致していたらその case 文の処理を実行する。すなわち，3 行目以降は以下の順で処理される。

- commands から'forward'を取り出し，5 行目の case 文で指定した値と一致するため，6 行目の処理が実行される。
- commands から'back'を取り出し，7 行目の case 文で指定した値と一致するため，8 行目の処理が実行される。
- commands から'Back'を取り出し，5，7，9，11 行目のいずれの case 文で指定した値とも一致しないが，13 行目の case 文で一致するため，14 行目の処理が実行される（13 行目にあるような指定はキャプチャパターンと呼ばれ，必ずパターンマッチングが成功する）。
- commands から'turn_left'を取り出し，11 行目の case 文で指定した値と一致するため，12 行目の処理が実行される。

したがって，「前へ移動」「後ろへ移動」「Back は存在しないコマンド」「左へ回る」の順で表示される。

問５

(25)	(26)	(27)	(28)	(29)
ウ	ア	ウ	イ	イ

(25) 5 行目から 12 行目の繰返し処理で，list_a から取り出した x の値に応じて，list_b の各要素に値を代入する。x が 80 以下の場合は'low'，80 より大きく 200 以下の場合は'middle'，それ以外の場合は'high'が代入される。したがって，list_b は['high', 'low', 'middle', 'low', 'middle']となる。

(26) 16 行目の list_a += list_c では，list_a のあとに list_c を結合している。結合したリストは list_a として保持される。したがって，list_a は[214, 54, 178, 3, 200, 4, 2, 6, 2]となる。

(27) 19 行目の list_d = list_c.copy()で，list_c を list_d にコピーしている。list_d は list_c と同じ要素をもつが，別のリストとしてコピーされているため，list_c は list_d の影響を受けない。20 行目で list_d[-2:]が指定されており，スライスして末尾から 2 つの要素をリストとして取り出すため，[6, 2]が選ばれる。21 行目の list_d.insert(2, x)で list_d の要素番号 2 の位置に，順に 6 と 2 を追加する。したがって，list_d は[4, 2, 2, 6, 6, 2]となる。

(28) remove は，引数に指定された値をリストから取り除くメソッドである。複数存在する場合は，最初に合致した要素のみ取り除く。25 行目で list_e から最初に合

致した要素の 2 を取り除くと[4, 6, 2]となる。List_c は List_e と同じリストを示すため，list_c は[4, 6, 2]となる。

(29) 31 行目から 35 行目で使用している append は，リストの末尾に値を追加するメソッドである。また，この条件分岐内では，list_f から取り出した要素 x を 2 乗した値によって，処理内容を分けている。x の 2 乗が 10 以上 30 未満であれば x の 2 倍を，30 以上であれば x の 2 乗を，それ以外であれば x の値をそのまま list_f に追加している。したがって，list_f の値「[0, 3, 36, 81]」が表示される。

問６

(30)	(31)	(32)	(33)	(34)
ア	ア	イ	ウ	イ

(30) ＜処理手順＞(2)の「出題数が単語数よりも多い（大きい）場合」の処理である。数値の比較には不等号の演算子を用いればよい。以上や以下の場合は等号（=）が必要であるが，本問では不要である。したがって，「<」となる。

(31) ＜処理手順＞(6)の「「問題□:○○の意味は？」と表示する」の処理である。問題として出題する英単語に対応するリスト words の要素番号は，ans_index に格納しているので，一つ目の要素番号は ans_index である。タプルの指定は，左から順に 0，1，…という番号で指定すればよく，本問では英語，日本語訳の順にデータを格納しているので，二つ目の要素番号には 0 を指定すればよい。したがって，「words[ans_index][0]」となる。

(32) ＜処理手順＞(7)の「選択肢は 2 行で表示し，1 行につき 2 つ並べてカンマ','で区切って表示する」処理である。2 行で対応するため，繰返し回数は 2 回である（取り出す値 i は 0 と 1 となる点に注意）。したがって，「2」となる。

(33) ＜処理手順＞(7)の「日本語訳の選択肢として，□:○○」と表示する。ここで，□は選択肢の 4 つの番号（1 から 4），○○はリスト words の先頭の 4 つの要素（日本語訳）が入る」処理である。1 行で二つ分の選択肢を表示する必要があり，一つ目の選択肢が空欄となっている。問題番号は 1 から始まるのに対し，リスト words は 0 から始まる点に注意すれば，「{i * 2 + 1}:{words[i * 2][1]}」であることが分かる。

(34) ＜処理手順＞(9)の「正解の番号と比較し，一致した場合は「正解！」と表示」する処理である。標準出力に表示される番号は 1 から 4 までの数字であるが，単語のリストの要素番号は 0 から 3 の数値で 1 だけずれることに注意する必要がある。なお，correct はこれまでに正解した問題数である。したがって，「ans_index + 1」となる。

〔 メ モ 用 紙 〕