

〔分野：アルゴリズムとプログラミング〕

① プログラムの基本要素

問1 次のプログラム中の に入れる正しい答えを，解答群の中から選べ。

ある施設の入場料は，0 歳から 3 歳までは 100 円，4 歳から 9 歳までは 300 円，10 歳以上は 500 円である。関数 fee は，年齢を表す 0 以上の整数を引数として受け取り，入場料を返す。

〔プログラム〕

```
○整数型: fee(整数型: age)
  整数型: ret
  if (age が 3 以下)
    ret ← 100
  elseif (  )
    ret ← 300
  else
    ret ← 500
  endif
  return ret
```

解答群

- ア (age が 4 以上) and (age が 9 より小さい)
- イ (age が 4 と等しい) or (age が 9 と等しい)
- ウ (age が 4 より大きい) and (age が 9 以下)
- エ age が 4 以上
- オ age が 4 より大きい
- カ age が 9 以下
- キ age が 9 より小さい

① プログラムの基本要素

問2 次のプログラム中の と に入れる正しい答えの組合せを，解答群の中から選べ。ここで，配列の要素番号は1から始まる。

次のプログラムは，整数型の配列arrayの要素の並びを逆順にする。

〔プログラム〕

整数型の配列: array ← {1, 2, 3, 4, 5}

整数型: right, left

整数型: tmp

for (left を 1 から (arrayの要素数 ÷ 2 の商) まで 1 ずつ増やす)

right ←

tmp ← array[right]

array[right] ← array[left]

← tmp

endfor

解答群

	a	b
ア	array の要素数 - left	array[left]
イ	array の要素数 - left	array[right]
ウ	array の要素数 - left + 1	array[left]
エ	array の要素数 - left + 1	array[right]

② データ構造及びアルゴリズム

問3 次のプログラム中の と に入れる正しい答えの組合せを，解答群の中から選べ。

手続 append は，引数で与えられた文字を単方向リストに追加する手続である。単方向リストの各要素は，クラス ListElement を用いて表現する。クラス ListElement の説明を図に示す。ListElement 型の変数はクラス ListElement のインスタンスの参照を格納するものとする。大域変数 listHead は，単方向リストの先頭の要素の参照を格納する。リストが空のときは，listHead は未定義である。

メンバ変数	型	説明
val	文字型	リストに格納する文字。
next	ListElement	リストの次の文字を保持するインスタンスの参照。初期状態は未定義である。

コンストラクタ	説明
ListElement(文字型: qVal)	引数 qVal でメンバ変数 val を初期化する。

図 クラス ListElement の説明

〔プログラム〕

大域: ListElement: listHead ← 未定義の値

○append(文字型: qVal)

```
ListElement: prev, curr
curr ← ListElement(qVal)
if (listHead が )
    listHead ← curr
else
    prev ← listHead
    while (prev.next が 未定義でない)
        prev ← prev.next
    endwhile
    prev.next ← 
endif
```

解答群

	a	b
ア	未定義	curr
イ	未定義	curr.next
ウ	未定義	listHead
エ	未定義でない	curr
オ	未定義でない	curr.next
カ	未定義でない	listHead

③ プログラミングの諸分野への適用

問4 次の記述中の ～ に入れる正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素番号は1から始まる。

要素の多くが0の行列を疎行列という。次のプログラムは、二次元配列に格納された行列のデータ量を削減するために、疎行列の格納に適したデータ構造に変換する。

関数 transformSparseMatrix は、引数 matrix で二次元配列として与えられた行列を、整数型配列の配列に変換して返す。関数 transformSparseMatrix を transformSparseMatrix({{3, 0, 0, 0, 0}, {0, 2, 2, 0, 0}, {0, 0, 0, 1, 3}, {0, 0, 0, 2, 0}, {0, 0, 0, 0, 1}})として呼び出したときの戻り値は、{{ }, { }, { }}である。

[プログラム]

```
○整数型配列の配列: transformSparseMatrix(整数型の二次元配列: matrix)
  整数型: i, j
  整数型配列の配列: sparseMatrix
  sparseMatrix ← {{}, {}, {}} /* 要素数0の配列を三つ要素にもつ配列 */
  for (i を 1 から matrixの行数 まで 1 ずつ増やす)
    for (j を 1 から matrixの列数 まで 1 ずつ増やす)
      if (matrix[i, j] が 0 でない)
        sparseMatrix[1]の末尾 に iの値 を追加する
        sparseMatrix[2]の末尾 に jの値 を追加する
        sparseMatrix[3]の末尾 に matrix[i, j]の値 を追加する
      endif
    endfor
  endfor
  return sparseMatrix
```

解答群

	a	b	c
ア	1, 2, 2, 3, 3, 4, 5	1, 2, 3, 4, 5, 4, 5	3, 2, 2, 1, 2, 3, 1
イ	1, 2, 2, 3, 3, 4, 5	1, 2, 3, 4, 5, 4, 5	3, 2, 2, 1, 3, 2, 1
ウ	1, 2, 3, 4, 5, 4, 5	1, 2, 2, 3, 3, 4, 5	3, 2, 2, 1, 2, 3, 1
エ	1, 2, 3, 4, 5, 4, 5	1, 2, 2, 3, 3, 4, 5	3, 2, 2, 1, 3, 2, 1