

## gzn010201 「データ構造 1」演習問題

### 問 1

計算機の中で、時刻などを記憶させる配列と 2 個のポインタを用いて表す F I F O 型のデータ構造はどれか。

- ア リスト                      イ スtring                      ウ スタック                      エ キュー

### 問 2

待ち行列に対する操作を次のとおり定義する。

ENQ n : 待ち行列にデータ n を挿入する。

DEQ : 待ち行列からデータを取り出す。

空の待ち行列に対し、ENQ 1, ENQ 2, ENQ 3, DEQ, ENQ 4, ENQ 5, DEQ, ENQ 6, DEQ, DEQ の操作を行った。次の DEQ の操作で取り出される値はどれか。

- ア 1                              イ 2                              ウ 5                              エ 6

### 問 3

キューに四つの値 8、3、6、1 がこの順に格納されている。このキューから最初に取り出される値はどれか。

- ア 1                              イ 3                              ウ 6                              エ 8

### 問 4

スタックに関する記述として、適切なものはどれか。

- ア 最後に格納したデータを最初に取り出すことができる。  
イ 最初に格納したデータを最初に取り出すことができる。  
ウ 探索キーからアドレスに変換することによって、データを取り出すことができる。  
エ 優先順位の高いデータを先に取り出すことができる。

### 問 5

空のスタックに対して次の操作を行った場合、スタックに残っているデータはどれか。ここで、“push x” はスタックへデータ x を格納し、“pop” はスタックからデータを取り出す操作を表す。

push 1 → push 2 → pop → push 3 → push 4 → pop → push 5 → pop

- ア 1 と 3                              イ 2 と 4                              ウ 2 と 5                              エ 4 と 5

### 問 6

A, B, C, Dの順に到着するデータに対して、一つのスタックだけを用いて出力可能なデータ列はどれか。

ア A, D, B, C

イ B, D, A, C

ウ C, B, D, A

エ D, C, A, B

### 問7

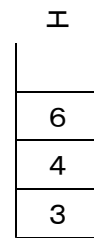
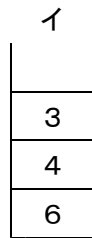
次の二つのスタック操作を定義する。

PUSH : スタックにデータ (整数 n) をプッシュする。

POP : スタックからデータをポップする。

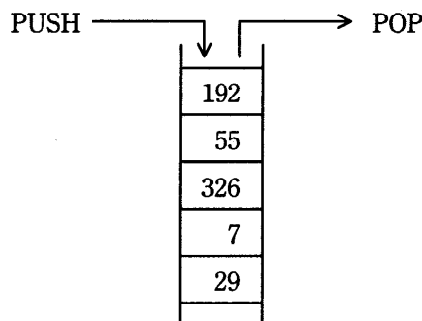
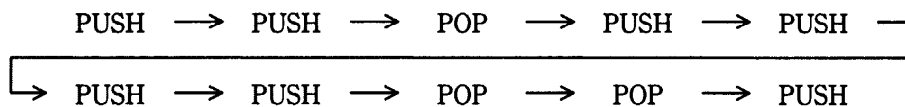
空のスタックに対して、次の順序で操作を行った結果はどれか。

PUSH 1 → PUSH 5 → POP → PUSH 7 → PUSH 6  
→ PUSH 4 → POP → POP → PUSH 3



### 問8

PUSH命令でスタックにデータを入れ、POP命令でスタックからデータを取り出す。動作中のプログラムにおいて、ある状態から次の順で10個の命令を実行したとき、スタックの中のデータは図のようになった。1番目のPUSH命令でスタックに入れたデータはどれか。



ア 7

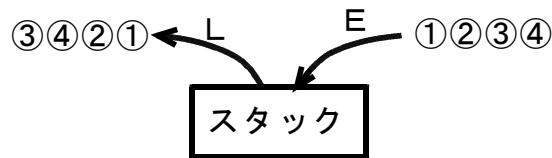
イ 29

ウ 55

エ 326

### 問9

図のように4個の要素①、②、③、④が順番に入口に並んでいる。入口からスタックに要素を入れる操作をE、スタックから出口に要素を出す操作をLとする。出口に出てくる要素の順序が③、④、②、①になる操作列はどれか。



ア EEEEELLLLL  
ウ EEEELLELL

イ EEELELLLL  
エ EEEELLELL

### 問10

十分な大きさの配列Aと初期値が0の変数pに対して、関数f(x)とg()が次のとおり定義されている。配列Aと変数pは、関数fとgだけでアクセス可能である。これらの関数が操作するデータ構造はどれか。

```
function f(x) {
  p = p + 1
  A[p] = x
  return None
}
function g() {
  x = A[p]
  p = p - 1
  return x
}
```

ア キュー  
ウ ハッシュ

イ スタック  
エ ヒープ

### 問11

関数や手続を呼び出す際に、戻り番地や処理途中のデータを一時的に保存するのに適したデータ構造はどれか。

ア 2分探索木  
ウ スタック

イ キュー  
エ 双方向連結リスト

### 問12

整数や実数などを格納するスタックがあり、次のルールで運用される。

- (1) 入力値が整数のとき、そのままスタックにプッシュする。
- (2) 入力値が実数のとき、小数部分を切り捨ててスタックにプッシュする。
- (3) 入力値が算術演算子(+、-、\*、/)のとき、スタックからポップを2回実行し、式  
(後からポップされた値) 算術演算子 (先にポップされた値)  
を実行後、結果をスタックにプッシュする。  
例えば演算子が“-”のとき、  
(後からポップされた値) - (先にポップされた値)  
の演算結果(差)をスタックにプッシュする。
- (4) 入力値がPのとき、スタック上に最新値を表示する。
- (5) 入力値がQのとき、スタックをクリアする。

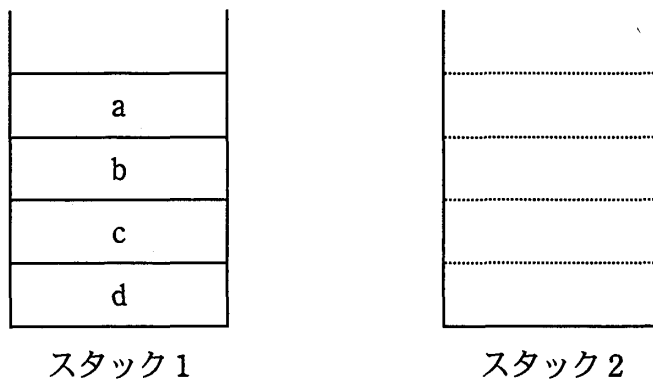
このスタックに次の5件のデータが、Qを先頭にこの順序で入力されたとき、スタック上の最新値はどれか。

Q、 -2.5、 3.5 -、 P

- ア -6                      イ -5                      ウ 1                      エ 5

**問13**

スタック1、2があり、図の状態になっている。関数fはスタック1からポップしたデータをそのままスタック2にプッシュする。関数gはスタック2からポップしたデータを出力する。b、c、d、aの順番に出力するためには、関数をどの順で実行すればよいか。



- ア f, f, g, f, f, g, g, g  
 イ f, f, g, f, g, f, g, g  
 ウ f, f, g, f, g, g, f, g  
 エ f, f, g, g, f, f, g, g

**問14**

スタックとキューの二つのデータ構造がある。次の手続きを順に実行した場合、変数 x に代入されるデータはどれか。ここで、データ a をスタックに挿入することを push(a)、スタックからデータを取り出すことを pop()、データ a をキューに挿入することを enq(a)、キューからデータを取り出すことを deq()、とそれぞれ表す。

push(a)、push(b)、enq(pop())、enq(c)、push(d)、push(deq())、x = pop()

- ア a                      イ b                      ウ c                      エ d

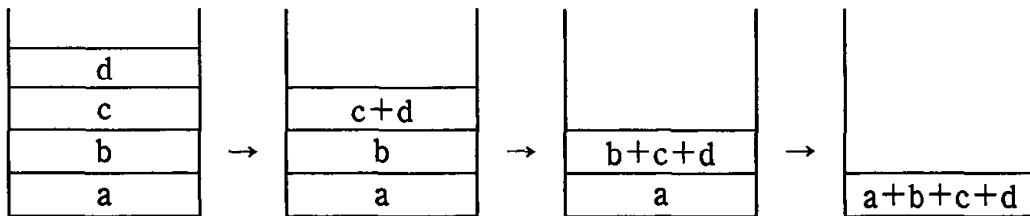
**問15**

四つのデータ A, B, C, D がこの順に入っているキューと空のスタックがある。手続 pop\_enq, deq\_push を使ってキューの中のデータを D, C, B, A の順に並べ替えるとき、deq\_push の実行回数は最小で何回か。ここで、pop\_enq はスタックから取り出したデータをキューに入れる操作であり、deq\_push はキューから取り出したデータをスタックに入れる操作である。

- ア 2                      イ 3                      ウ 4                      エ 5

**問16**

図は、逆ポーランド表記法で書かれた a b c d + + + をスタックで処理するときのスタックの変化の一部を表している。この場合、スタックの深さは最大で 4 となる。最大のスタックの深さが最も少ない逆ポーランド表記法の式はどれか。



- ア a b + c + d +                      イ a b + c d + +  
 ウ a b c + + d +                      エ a b c + d + +

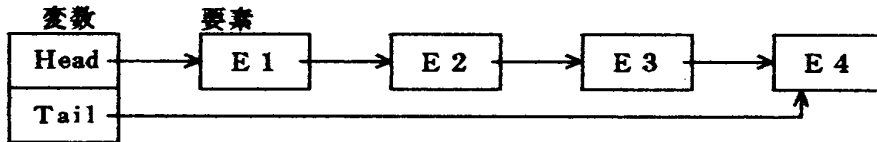
**問17**

配列と比較した場合の連結リストの特徴に関する記述として、適切なものはどれか。

- ア 要素を更新する場合、ポインタを順番にたどるだけなので、処理時間は短い。
- イ 要素を削除する場合、削除した要素から後ろにあるすべての要素を前に移動するので、処理時間は長い。
- ウ 要素を参照する場合、ランダムにアクセスできるので、処理時間は短い。
- エ 要素を挿入する場合、数個のポインタを書き換えるだけなので、処理時間は短い。

**問18**

右のような構造をもった線形リストについて、正しい記述はどれか。



- ア 要素の削除に要する処理量は、先頭と最後尾とでほぼ同じである。
- イ 要素の追加と取出し(読出し後削除)を最後尾で行うスタックとしての利用に適している。
- ウ 要素の追加に要する処理量は、先頭と最後尾とでほぼ同じである。
- エ 要素の追加は先頭に、取出し(読出し後削除)は最後尾からとするFIFO(First-in First-out)のキューとしての利用に適している。

**問19**

次のような双方向のポインタをもつリスト構造のデータがある。社員Gを社員Aと社員Kの間に追加する場合、追加後の表のポインタa~fのうち、追加前と比べて値が変わるのは何か所か。

追加前

アドレス	社員名	次ポインタ	前ポインタ
100	社員A	300	0
200	社員T	0	300
300	社員K	200	100

追加後

アドレス	社員名	次ポインタ	前ポインタ
100	社員A	a	b
200	社員T	c	d
300	社員K	e	f
400	社員G	x	y

- ア 1
- イ 2
- ウ 3
- エ 4

**問20**

多数のデータが単方向リスト構造で格納されている。このリスト構造には、先頭ポインタとは別に、末尾のデータを指し示す末尾ポインタがある。次の操作のうち、ポインタを参照する回数が最も多いものはどれか。

- ア リストの先頭にデータを挿入する。
- イ リストの先頭のデータを削除する。
- ウ リストの末尾にデータを挿入する。
- エ リストの末尾のデータを削除する。

**問21**

表は、配列を用いた連結セルによるリストの内部表現であり、リスト [東京, 品川, 名古屋, 新大阪] を表している。このリストを [東京, 新横浜, 名古屋, 新大阪] に変化させる操作はどれか。ここで、 $A(i, j)$  は表の第  $i$  行第  $j$  列の要素を表す。例えば、 $A(3, 1) = \text{“名古屋”}$  であり、 $A(3, 2) = 4$  である。また、 $\rightarrow$  は代入を表す。

		列	
	A	1	2
行	1	“東京”	2
	2	“品川”	3
	3	“名古屋”	4
	4	“新大阪”	0
	5	“新横浜”	

	第1の操作	第2の操作
ア	$5 \rightarrow A(1, 2)$	$A(A(1, 2), 2) \rightarrow A(5, 2)$
イ	$5 \rightarrow A(1, 2)$	$A(A(2, 2), 2) \rightarrow A(5, 2)$
ウ	$A(A(1, 2), 2) \rightarrow A(5, 2)$	$5 \rightarrow A(1, 2)$
エ	$A(A(2, 2), 2) \rightarrow A(5, 2)$	$5 \rightarrow A(1, 2)$

**問22**

図は単方向リストを表している。“東京” がリストの先頭であり、そのポインタには次のデータのアドレスが入っている。また、“名古屋” はリストの最後であり、そのポインタには0が入っている。アドレス150に置かれた“静岡”を、“熱海”と“浜松”の間に挿入する処理として正しいものはどれか。

先頭へのポインタ	アドレス	データ	ポインタ
10	10	東京	50
	30	名古屋	0
	50	新横浜	90
	70	浜松	30
	90	熱海	70
	150	静岡	

- ア 静岡のポインタを50とし、浜松のポインタを150とする。
- イ 静岡のポインタを70とし、熱海のポインタを150とする。
- ウ 静岡のポインタを90とし、浜松のポインタを150とする。
- エ 静岡のポインタを150とし、熱海のポインタを90とする。

**問23**

リストは、配列で実現する場合とポインタで実現する場合とがある。リストを配列で実現した場合の特徴として、適切なものはどれか。

- ア リストにある実際の要素数にかかわらず、リストの最大長に対応した領域を確保し、実際には使用されない領域が発生する可能性がある。
- イ リストにある実際の要素数にかかわらず、リストへの挿入と削除は一定時間で行うことができる。
- ウ リストの中間要素を参照するには、リストの先頭から順番に要素をたどっていくので、要素数に比例した時間が必要となる。
- エ リストの要素を格納する領域の他に、次の要素を指し示すための領域が別途必要となる。

#### 問24

次の規則に従って配列の要素  $A[0]$ ,  $A[1]$ , ...,  $A[9]$  に正の整数  $k$  を格納する。16, 43, 73, 24, 85 を順に格納したとき、85 が格納される場所はどれか。ここで、 $x \bmod y$  は  $x$  を  $y$  で割った剰余を返す。また、配列の要素はすべて 0 に初期化されている。

〔規則〕

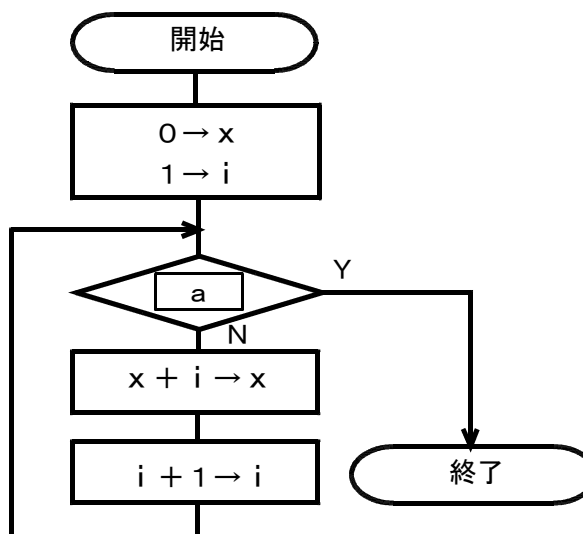
- (1)  $A[k \bmod 10] = 0$  ならば、 $k \rightarrow A[k \bmod 10]$  とする。
- (2) (1) で格納できないとき、 $A[(k+1) \bmod 10] = 0$  ならば、 $k \rightarrow A[(k+1) \bmod 10]$  とする。
- (3) (2) で格納できないとき、 $A[(k+4) \bmod 10] = 0$  ならば、 $k \rightarrow A[(k+4) \bmod 10]$  とする。

- ア  $A[3]$                       イ  $A[5]$                       ウ  $A[6]$                       エ  $A[9]$

#### 問25

流れ図は、1 から  $N$  ( $N \geq 1$ ) までの整数の総和 ( $1 + 2 + \dots + N$ ) を求め、結果を変数  $x$  に入れるアルゴリズムを示している。流れ図中の  $a$  に当てはまる式はどれか。

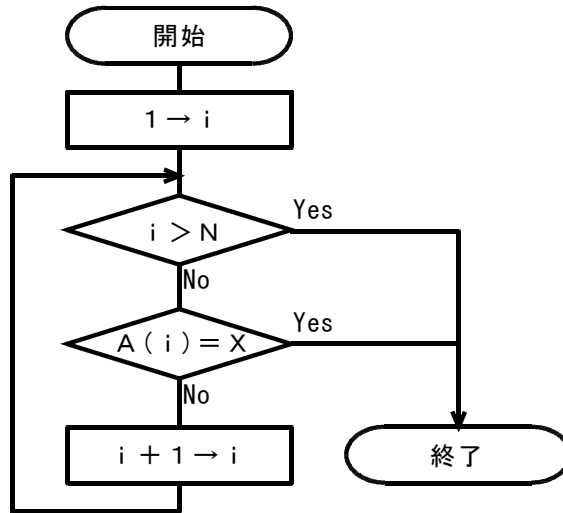
- ア  $i = N$
- イ  $i < N$
- ウ  $i > N$
- エ  $x > N$



#### 問26



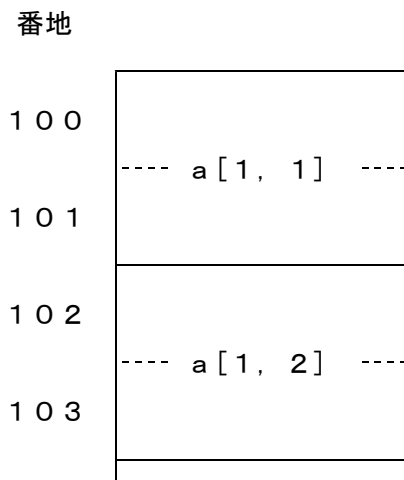
N個の要素からなる配列Aの各要素に整数が格納されている ( $N > 1$ )。Xと同じ値が何番目の要素に格納されているかを調べる流れ図である。この流れ図の実行結果として、正しい記述はどれか。



- ア Xと同じ値が配列中にない場合、iには1が設定されている。
- イ Xと同じ値が配列中にない場合、iにはNが設定されている。
- ウ Xと同じ値が配列の1番目とN番目にある場合、iには1が設定されている。
- エ Xと同じ値が配列の1番目とN番目にある場合、iにはNが設定されている。

**問27**

10行10列の2次元配列aを、次のようにメモリ上の連続した領域へ行方向に格納するとき、a[5, 6]が格納される場所の番地はどれか。ここで、番地は10進数表示とする。

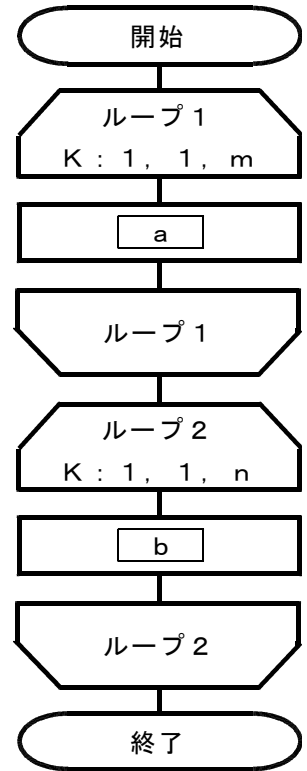


- ア 145
- イ 185
- ウ 190
- エ 208

**問28**

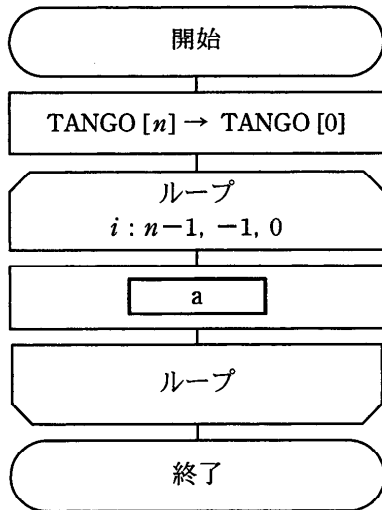
長さ  $m$ 、 $n$  の文字列を格納した配列  $X$ 、 $Y$  がある。図はこれらの文字列をこの順に連結した文字列を配列  $Z$  に格納する処理の流れ図である。  
 $a$ 、 $b$  に入れる処理として、正しいものはどれか。ここで、1文字が一つの配列要素に格納されるものとする。

	a	b
ア	$X(k) \rightarrow Z(k)$	$Y(k) \rightarrow Z(m+k)$
イ	$X(k) \rightarrow Z(k)$	$Y(k) \rightarrow Z(n+k)$
ウ	$Y(k) \rightarrow Z(k)$	$X(k) \rightarrow Z(m+k)$
エ	$Y(k) \rightarrow Z(k)$	$X(k) \rightarrow Z(n+k)$



**問29**

要素番号が 0 から始まる配列  $TANGO$  がある。  $n$  個の単語が  $TANGO[1]$  から  $TANGO[n]$  に入っている。図は、  $n$  番目の単語を  $TANGO[1]$  に移動するために、  $TANGO[1]$  から  $TANGO[n-1]$  の単語を順に一つずつ後ろにずらして単語表を再構成する流れ図である。  
 $a$  に入れる処理として、適切なものはどれか。



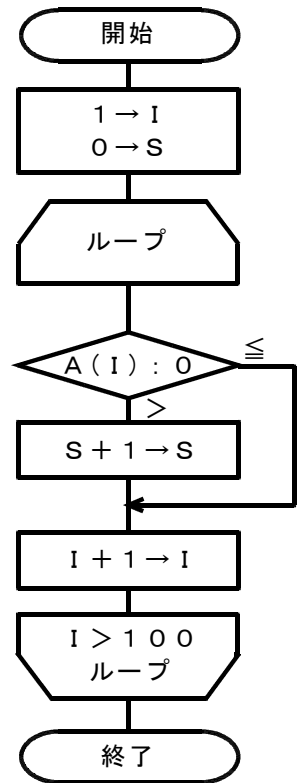
(注) ループにおける条件は、  
 変数名：初期値，増分，終値  
 を示す。

- ア  $TANGO[i] \rightarrow TANGO[i+1]$
- イ  $TANGO[i] \rightarrow TANGO[n-i]$
- ウ  $TANGO[i+1] \rightarrow TANGO[n-i]$
- エ  $TANGO[n-i] \rightarrow TANGO[i]$

**問30**

次の流れ図に対応する処理結果Sの内容として適切なものはどれか。

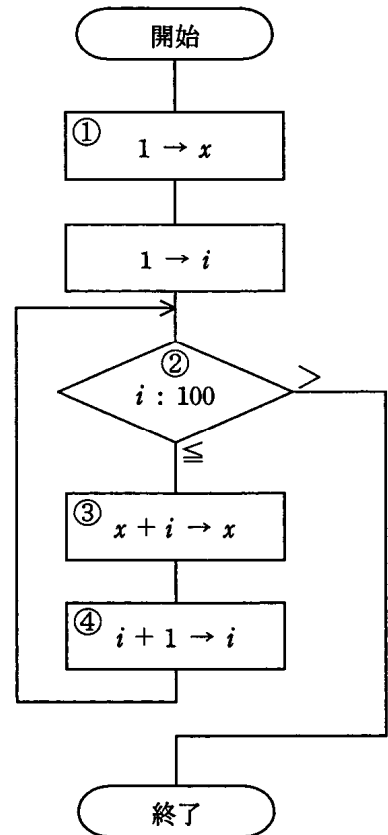
- ア 100個の数値の総和
- イ 100個の数値中の正の数値の和または0
- ウ 100個の数値中の正の数値の個数
- エ 100個の数値中の最大値



**問31**

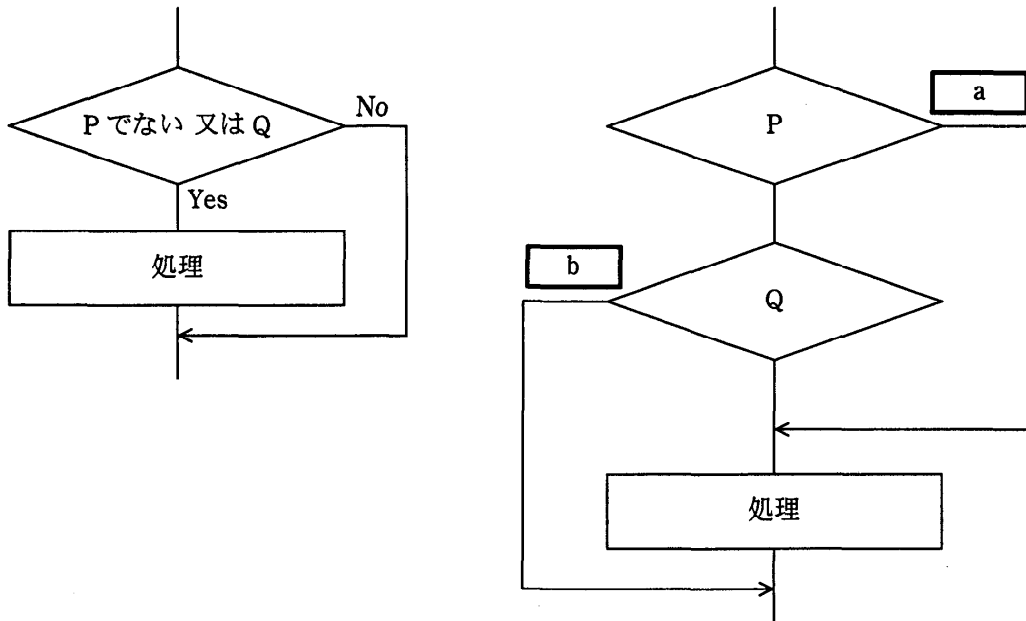
次の流れ図は、1から100までの整数の総和を求め、結果を変数xに代入するアルゴリズムを示したものであるが、一部誤りがある。どのように訂正すればよいか。

- ア ①の処理を“0 → x”にする。
- イ ②の条件判定を“i : 99”にする。
- ウ ③の処理を“x + i → i”にする。
- エ ④の処理を“x + 1 → x”にする。



**問32**

右の流れ図が左の流れ図と同じ動作をするために、a、bに入るYesとNoの組合せはどれか。



	a	b
ア	No	No
イ	No	Yes
ウ	Yes	No
エ	Yes	Yes

**問33**

節点 1, 2, ..., n をもつ木を表現するために、大きさ n の整数型配列 A[1], A[2], ..., A[n] を用意して、節点 i の親の番号を A[i] に格納する。節点 k が根の場合は A[k] = 0 とする。表に示す配列が表す木の葉の数は、幾つか。

i	1	2	3	4	5	6	7	8
A[i]	0	1	1	3	3	5	5	5

ア 1

イ 3

ウ 5

エ 7

**問34**

次のうち配列の特徴を表している適切な組合せはどれか。

- I 一つの要素の大きさはあらかじめ決まっている。
- II 配列全体の大きさは実行中に変えられる。
- III 要素は添字で識別する。
- IV 各要素は連続した領域に格納される。
- V 各要素は必ずしも連続した領域に格納されなくともよい。

- ア I、III、IV      イ I、III、V      ウ I、IV      エ II、IV

**問35**

配列Aが図2の状態のとき、図1の流れ図を実行すると、配列Bが図3の状態になった。図1のaに入れるべき操作はどれか。ここで、配列A、Bの要素をそれぞれA(i, j), B(i, j)とする。

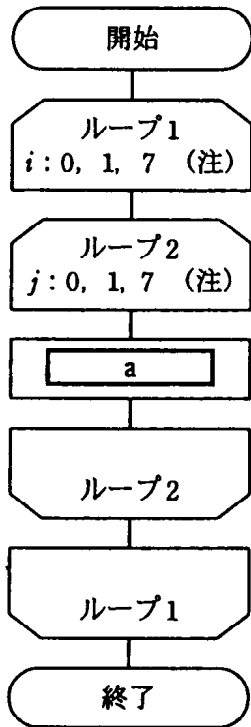


図1 流れ図

		j →							
		0	1	2	3	4	5	6	7
i ↓	0		*	*	*	*	*	*	
	1		*						
	2		*						
	3		*	*	*	*			
	4		*						
	5		*						
	6		*						
	7		*						

図2 配列Aの状態

		j →							
		0	1	2	3	4	5	6	7
i ↓	0								
	1	*	*	*	*	*	*	*	*
	2					*			*
	3					*			*
	4					*			*
	5								*
	6								*
	7								*

図3 実行後の配列Bの状態

(注) ループ端の繰返し指定は、  
変数名：初期値，増分，終値  
を示す。

- ア  $A(i, j) \rightarrow B(i, 7-j)$
- イ  $A(i, j) \rightarrow B(j, 7-i)$
- ウ  $A(i, j) \rightarrow B(7-j, i)$
- エ  $A(i, j) \rightarrow B(7-i, 7-j)$

**問36**

三次元の配列を表すとき“ $X(I, J, K)$ ”と表記するが、このとき、 $J$ が表しているものはどれか。

- ア 行                      イ 面                      ウ 要素                      エ 列

**問37**

ポインタを用いた線形リストの特徴のうち、適切なものはどれか。

- ア 先頭の要素を根とした $n$ 分木で、先頭以外の要素は全て先頭の要素の子である。  
イ 配列を用いた場合と比較して、2分探索を効率的に行うことが可能である。  
ウ ポインタから次の要素を求めるためにハッシュ関数を用いる。  
エ ポインタによって指定されている要素の後ろに、新たな要素を追加する計算量は、要素の個数や位置によらず一定である。

**問38**

2次元の整数型配列 $a$ の各要素 $a(i, j)$ の値は、 $2i + j$ である。このとき、 $a(a(1, 1) \times 2, a(2, 2) + 1)$ の値は幾つか。

- ア 12                      イ 13                      ウ 18                      エ 19

**問39**

データ構造の一つであるリストは、配列を用いて実現する場合と、ポインタを用いて実現する場合とがある。配列を用いて実現する場合の特徴はどれか。ここで、配列を用いたリストは、配列に要素を連続して格納することによって構成し、ポインタを用いたリストは、要素から次の要素へポインタで連結することによって構成するものとする。

- ア 位置を指定して、任意のデータに直接アクセスすることができる。  
イ 並んでいるデータの先頭に任意のデータを効率的に挿入することができる。  
ウ 任意のデータの参照は効率的ではないが、削除や挿入の操作を効率的に行える。  
エ 任意のデータを別の位置に移動する場合、隣接するデータを移動せずにできる。