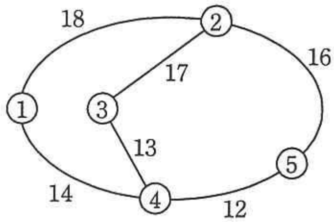
次の問8は必須問題です。必ず解答してください。

問8　次のプログラムの説明及びプログラムを読んで，設問に答えよ。

鉄道の路線がある。駅数 N は 2 以上で，各駅には駅番号（1，2，…，N）が付いている。どの任意の 2 駅間にも，それらを結ぶ経路が一つ以上存在する。また，隣接する 2 駅間を直接結ぶ経路は一つだけ存在し，その距離が与えられている。図 1 に 5 駅からなる鉄道の路線例を示す。



注記　○は駅を表し，その中の数字は駅番号を表す。
また，経路に付した数字は 2 駅間の距離を表す。

図 1　鉄道の路線例（1）

〔プログラムの説明〕

副プログラム CalcDist は，駅数 N 及び要素数 N×N の 2 次元配列 Dist を受け取り，プログラムに示すアルゴリズム（Warshall-Floyd 法）によって，配列 Dist の内容を更新しながら各駅間の最短距離を求めていく。実行が終わると，任意の 2 駅 i，j 間の最短距離が Dist[i][j] に求められている。どの 2 駅間についても，その最短距離は 999 より小さいものとする。配列 Dist の添字は 1 から始まる。

副プログラム CalcDist に渡す配列 Dist には，次の値を格納する。

なお，図 1 の路線情報を格納した配列 Dist の内容を，図 2 の初期値に示してある。

(1)　2 駅 i，j が隣接しているとき，Dist[i][j] 及び Dist[j][i] には，その駅間距離を格納する。例えば，図 1 の駅①と②は隣接しているので，Dist[1][2] 及び Dist[2][1] には，18 を格納する。

(2)　2 駅 i，j が隣接していないとき，Dist[i][j] 及び Dist[j][i] には，未確定を表す距離 999 を格納する。例えば，図 1 の駅①と③は隣接していないので，Dist[1][3] 及び Dist[3][1] には，999 を格納する。

(3)　各駅 i について，Dist[i][i] には 0 を格納する。

副プログラム CalcDist 中の Print(N, Dist) は，配列 Dist のその時点の内容を印字する副プログラムである。

〔プログラム〕

```
○副プログラム：CalcDist(整数型：N, 整数型：Dist[][])
○整数型：From, To, Via
・Print(N, Dist)
■ Via: 1, Via ≦ N, 1
|  ■ From: 1, From ≦ N, 1
|  |  ■ To: 1, To ≦ N, 1
|  |  |  ▲ Dist[From][To] > Dist[From][Via] + Dist[Via][To]
|  |  |  |  ・Dist[From][To] ← Dist[From][Via] + Dist[Via][To]
|  |  |  ▼
|  |  ■
|  ■
|  ・Print(N, Dist)
■
```

図 1 の路線情報を配列 Dist に格納して，CalcDist を実行した。配列 Dist の内容の変化を Print(N, Dist) で印字した結果は，図 2 のとおりであった。

初期値

0	18	999	14	999
18	0	17	999	16
999	17	0	13	999
14	999	13	0	12
999	16	999	12	0

Via ＝ 1

0	18	999	14	999
18	0	17	32	16
999	17	0	13	999
14	32	13	0	12
999	16	999	12	0

Via ＝ 2

0	18	a	14	34
18	0	17	32	16
a	17	0	13	b
14	32	13	0	12
34	16	b	12	0

Via ＝ 3

0	18	35	14	34
18	0	17	30	16
35	17	0	13	33
14	30	13	0	12
34	16	33	12	0

Via ＝ 4

0	18	27	14	26
18	0	17	30	16
27	17	0	13	25
14	30	13	0	12
26	16	25	12	0

Via ＝ 5

0	18	27	14	26
18	0	17	28	16
27	17	0	13	25
14	28	13	0	12
26	16	25	12	0

注記　“初期値”，“Via ＝ 1” などは，説明のために付加してある。

図 2　配列 Dist の内容の変化を印字した結果

駅数 N に関して，副プログラム CalcDist の計算量のオーダは［　c　］，またメモリ使用量のオーダは［　d　］である。そこで，駅数 N が大きい場合に，計算量やメモリ使用量を減らすための方法を考える。

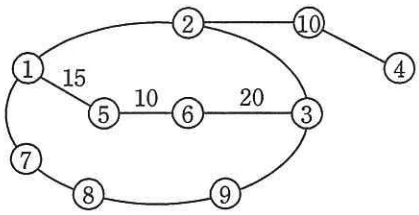
例えば，配列 Dist の対称性に着目し，プログラム中の［　　　　］の部分を，

```
■ From: 1, From ≦ N - 1, 1
│ ■ To: [        e        ]
│ │ ▲ Dist[From][To] ＞ Dist[From][Via] + Dist[Via][To]
│ │ │ • Dist[From][To] ← Dist[From][Via] + Dist[Via][To]
│ │ │ • Dist[To][From] ← Dist[From][Via] + Dist[Via][To]
│ │ ▼
│ ■
■
```

とすれば，繰返し処理中の選択処理の実行回数を約半分に減らすことができる。

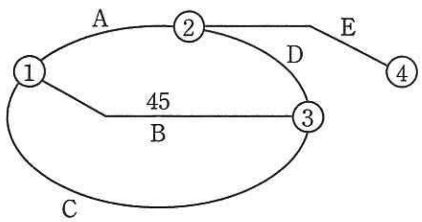
次に，少ないメモリ使用量で，任意の 2 駅 i，j 間の最短距離を求める方法を考える。

駅を，乗換駅（3 方向以上に隣接する駅がある），終端駅（1 方向だけに隣接する駅がある），中間駅（2 方向だけに隣接する駅がある）の 3 種類に分ける。乗換駅と終端駅を合わせて基幹駅と呼ぶ。基幹駅の数を K として，駅番号 1～K が基幹駅，駅番号 K+1～N が中間駅となるように駅番号を定める。ここで，K≧2 であるとする。図 3 の路線例では，駅①～③が乗換駅，駅④が終端駅，駅⑤～⑩が中間駅である。



注記　2 駅間の距離は一部だけを表示している。

図 3　鉄道の路線例（2）



注記　A～E は区間名を表す。

図 4　鉄道の路線例（2）の基幹駅と区間名

各駅間の最短距離は，次の方法で求める。

(1)　中間駅を全て除いて，基幹駅だけからなる路線を考え，二つの基幹駅を結ぶ経路ごとに一意の区間名を付ける。図 3 の路線例にこの操作を加えたものが図 4 である。例えば，図 3 の経路①－⑤－⑥－③は，図 4 の区間 B に対応する。

(2)　要素数 K×K の 2 次元配列 Dist を用意し，事前に配列 Dist に各基幹駅間の最短距離を求めておく。

(3)　全ての駅について，表 1 に示す形式の駅情報表を用意する（表 1 には，図 3 の駅情報の一部が示してある）。

中間駅の場合，Sec はその駅が属する区間の区間名，KL，KH はその駅が属する区間の両端の駅番号（KL ≦ KH とする），ToKL，ToKH はその駅から駅 KL，KH までの距離である。基幹駅の場合，その駅は自分自身の駅を両端とする距離 0 の区間に属すると考えて，表 1 の例のように中間駅と同様の情報を用意する。

これらの情報は，駅番号を添字として参照する。例えば，駅番号 5 の ToKL の値は ToKL[5] として参照する。

表 1　駅情報表の形式

駅番号	Sec	KL	ToKL	KH	ToKH
1	“1”	1	0	1	0
2	“2”	2	0	2	0
3	“3”	3	0	3	0
4	“4”	4	0	4	0
5	“B”	1	15	3	30
6	“B”	1	25	3	20
⋮	⋮	⋮	⋮	⋮	⋮

(4)　2 駅 i，j 間の最短距離を求める。一般に，両駅が属する区間が異なる場合，駅 i から駅 j へ行く経路は，

（駅 i）→（駅 i が属する区間のいずれか一端の基幹駅）

→（駅 j が属する区間のいずれか一端の基幹駅）→（駅 j）

となる。したがって，次の式で求める 4 通りの値 D1，D2，D3，D4 のうちの最小値が，2 駅 i，j 間の最短距離となる。

- D1 ← ToKL[i] ＋ Dist[KL[i]][KL[j]] ＋ ToKL[j]
- D2 ← ToKL[i] ＋ Dist[KL[i]][KH[j]] ＋ ToKH[j]
- D3 ← ToKH[i] ＋ Dist[KH[i]][KL[j]] ＋ ToKL[j]
- D4 ← ToKH[i] ＋ Dist[KH[i]][KH[j]] ＋ ToKH[j]

両駅が属する区間が同じ場合は，区間の端の基幹駅を経由せずに直接駅 i から駅 j へ行く経路がある。これも考慮すると，任意の２駅 i，j 間の最短距離 Res は次の処理で求められる。ここで，関数 min(式$_1$，式$_2$，…) は，式$_1$，式$_2$，… の値の最小値を返す関数であり，関数 abs(式) は，式の値の絶対値を返す関数である。

```
▲ [          f          ]
│ • Res ← min(abs(ToKL[i]−ToKL[j]), D1, D2, D3, D4)
├──────
│ • Res ← min(D1, D2, D3, D4)
▼
```

この方法なら，２次元配列 Dist の要素数を N×N から K×K に削減できる。ただし，表１に示す表が増えるので，例えば K=5 のとき，配列及び表が占めるメモリ使用量を削減できるのは N≧ $\boxed{\text{　g　}}$ の場合となる。ここで，表１に示す表は，１駅について配列 Dist の要素５個分のメモリを使用するものとする。

設問　本文中及び図２中の $\boxed{\qquad}$ に入れる正しい答えを，解答群の中から選べ。

a，b に関する解答群

ア　33　　　イ　34　　　ウ　35　　　エ　999

c，d に関する解答群

ア　$O(\mathrm{N})$　　　イ　$O(\mathrm{N}\log\mathrm{N})$　　　ウ　$O(\mathrm{N}^2)$　　　エ　$O(\mathrm{N}^3)$

e に関する解答群

ア　1, To ≦ From, 1　　　イ　1, To ≦ From + 1, 1

ウ　From, To ≦ N − 1, 1　　　エ　From + 1, To ≦ N, 1 fに関する解答群

ア　(KL[i] ＝ KL[j]) and (KH[i] ＝ KH[j])

イ　(KL[i] ≠ KL[j]) or (KH[i] ≠ KH[j])

ウ　Sec[i] ＝ Sec[j]

エ　Sec[i] ≠ Sec[j]

gに関する解答群

ア　6　　イ　8　　ウ　9　　エ　14