第２問　次の文章を読み，後の問い(問１～３)に答えよ。

Mさんは，18歳になって選挙権が得られたのを機に，比例代表選挙の当選者を決定する仕組みに興味を持った。そこで各政党に配分する議席数（当選者数）を決める方法を，友人のKさんとプログラムを用いて検討してみることにした。

問１　次の文章の空欄［ア］～［ウ］に入れる最も適当なものを，後の解答群のうちから一つずつ選べ。同じものを繰り返し選んでもよい。

Mさん：表１に，最近行われた選挙結果のうち，ある地域のブロックについて，各政党の得票数を書いてみたよ。

表１　各政党の得票数

	A党	B党	C党	D党
得票数	1200	660	1440	180

Kさん：今回の議席数は６人だったね。得票の総数を議席数で割ると580人なので，これを基準得票数と呼ぶのがいいかな。平均して１議席が何票分の重みがあるかを表す数ということで。そうすると，各政党の得票数が何議席分に相当するかは，各政党の得票数をこの基準得票数で割れば求められるね。

Mさん：その考え方に沿って政党ごとの当選者数を計算するプログラムを書いてみよう。まず，プログラムの中で扱うデータを図１と図２にまとめてみたよ。配列 Tomei には各政党の党名を，配列 Tokuhyo には各政党の得票数を格納することにしよう。政党の数は４つなので，各配列の添字は０から３だね。

i	0	1	2	3
Tomei	A党	B党	C党	D党

図１　各政党名が格納されている配列

i	0	1	2	3
Tokuhyo	1200	660	1440	180

図２　得票数が格納されている配列

Mさん：では，これらのデータを使って，各政党の当選者数を求める図３のプログラムを書いてみよう。実行したら図４の結果が表示されたよ。

```
(01) Tomei = ["A党","B党","C党","D党"]
(02) Tokuhyo = [1200,660,1440,180]
(03) sousuu = 0
(04) giseki = 6
(05) mを0から [ア] まで1ずつ増やしながら繰り返す:
(06)     sousuu = sousuu + Tokuhyo[m]
(07) kizyunsuu = sousuu / giseki
(08) 表示する("基準得票数:",kizyunsuu )
(09) 表示する("比例配分")
(10) mを0から [ア] まで1ずつ増やしながら繰り返す:
(11)     表示する(Tomei[m],":",[イ] / [ウ] )
```

図3　得票に比例した各政党の当選者数を求めるプログラム

Kさん：得票数に比例して配分すると小数点のある人数になってしまうね。小数点以下の数はどう考えようか。例えば，A党は2.068966だから２人が当選するのかな。

Mさん：なるほど。切り捨てで計算すると，A党は２人，B党は１人，C党は２人，D党は０人になるね。あれ？ 当選者数の合計は５人で，６人に足りないよ。

```
基準得票数：580
比例配分
A党：2.068966
B党：1.137931
C党：2.482759
D党：0.310345
```

図4　各政党の当選者数の表示

Kさん：切り捨ての代わりに四捨五入したらどうだろう。

Mさん：そうだね。ただ，この場合はどの政党も小数点以下が0.5未満だから，切り捨てた場合と変わらないな。だからといって小数点以下を切り上げると，当選者数が合計で９人になるから３人も多くなってしまう。

Kさん：このままでは上手くいかないなぁ。先生に聞いてみよう。

[ア]～[ウ]の解答群

⓪ 0　① 1　② 2　③ 3　④ 4　⑤ 5　⑥ 6　⑦ Tomei[m]
⑧ Tokuhyo[m]　⑨ sousuu　ⓐ giseki　ⓑ kizyunsuu

問２　次の文章の空欄［エ］～［ス］に入れる最も適当なものを，後の解答群のうちから一つずつ選べ。同じものを繰り返し選んでもよい。

Mさん：先生，比例代表選挙では各政党の当選者数はどうやって決まるのですか？ 当選者数が整数なので，割合だけだと上手くいかなかったのです。

先　生：様々な方法があるけど，日本では各政党の得票数を１，２，３，…と整数で割った商の大きい順に定められた議席を配分していく方法を採用しているよ。この例だと表２のように，❶から❻の順に議席が各政党に割り当てられるんだ。C党が❶の議席を取っているけど，このとき，何の数値を比較したか分かるかな。

表２　各政党の得票数と整数で割った商

	A党	B党	C党	D党
得票数	1200	660	1440	180
1で割った商	❷1200	❹ 660	❶1440	180
2で割った商	❺ 600	330	❸ 720	90
3で割った商	400	220	❻ 480	60
4で割った商	300	165	360	45

Mさん：１で割った商です。A党から順に 1200，660，1440，180 ですね。

先　生：そうだね。ではA党が❷の議席を取るとき，何の数値を比較したのだろうか。

Mさん：C党は１議席目を取ったので，1440 を２で割った商である 720 を比較します。A党から順に 1200，660，720，180 ですね。この中で数値が大きいA党が議席を取ります。なるほど，妥当な方法ですね。

Kさん：この考え方で手順を考えてみようよ。

先　生：まずは候補者が十分足りるという条件で手順を考えてみるのがいいですよ。

Kさん：各政党に割り当てる議席を決めるために，比較する数値を格納する配列 Hikaku がいるね。

Mさん：各政党に配分する議席数（当選者数）を格納する配列 Tosen も必要だね。最初は議席の配分が行われていないから，初期値は全部 0 にしておくね。

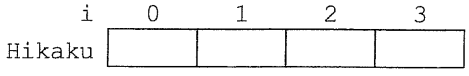

図５　整数で割った値を格納する配列

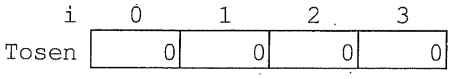

図６　当選者数を格納する配列

Kさん：「2で割った商」の「2」のように，各政党の得票数を割るときに使う数字はどうすればいいかな。

Mさん：その政党の当選者数＋1でいいよね。配列 Tosen が使えるね。そうだ，変化したところだけ計算し直せばいいんじゃない？ 議席を配分する手順を書いてみよう。

手順1 配列 Tokuhyo の各要素の値を配列 Hikaku の初期値として格納する。
手順2 配列 Hikaku の要素の中で最大の値を調べ，その添字 maxi に対応する配列 Tosen[maxi] に1を加える。
手順3 Tokuhyo[maxi] を Tosen[maxi]＋1 で割った商を Hikaku[maxi] に格納する。
手順4 手順2と手順3を当選者数の合計が議席数の6になるまで繰り返す。
手順5 各政党の党名（配列 Tomei）とその当選者数（配列 Tosen）を順に表示する。

図7　手順を書き出した文章

Kさん：この図7の手順が正しいか確認するために，配列 Hikaku と配列 Tosen の中がどう変化していくか確認してみよう。図8のようになるね。

	配列 Hikaku の変化				配列 Tosen の変化			
i	0	1	2	3	0	1	2	3
手順1終了時	1200	660	1440	180	0	0	0	0
1回目の手順3終了時	1200	660	720	180	0	0	1	0
2回目の手順3終了時	600	660	エ	180	1	0	ケ	0
3回目の手順3終了時	600	660	オ	180	1	0	コ	0
4回目の手順3終了時	600	330	カ	180	1	1	サ	0
5回目の手順3終了時	400	330	キ	180	2	1	シ	0
6回目の手順3終了時	400	330	ク	180	2	1	ス	0

図8　配列 Hikaku と配列 Tosen の変化

Mさん：先生に教えてもらった結果と同じように，議席数が6になるまで議席を配分できたね。この手順でプログラムを考えてみよう。

エ ～ ス の解答群

⓪ 0	① 1	② 2	③ 3	④ 4	⑤ 180
⑥ 288	⑦ 360	⑧ 400	⑨ 480	ⓐ 600	ⓑ 720

問3　次の文章の空欄［セ］～［テ］に入れる最も適当なものを，後の解答群のうちから一つずつ選べ。

Mさん：図9のプログラムを作ってみたよ。商を整数で求めるところは小数点以下を切り捨てる「切り捨て」という関数を使ったよ。

Kさん：実行したら図10のように正しく政党名と当選者数が得られたね。

```
(01) Tomei = ["A党","B党","C党","D党"]
(02) Tokuhyo = [1200,660,1440,180]
(03) Tosen = [0,0,0,0]
(04) tosenkei = 0
(05) giseki = 6
(06) m を 0 から［ア］まで 1 ずつ増やしながら繰り返す:
(07)     Hikaku[m] = Tokuhyo[m]
(08) ［セ］< giseki の間繰り返す:
(09)     max = 0
(10)     i を 0 から［ア］まで 1 ずつ増やしながら繰り返す:
(11)         もし max < Hikaku[i]ならば:
(12)             ［ソ］
(13)             maxi = i
(14)     Tosen[maxi] = Tosen[maxi] + 1
(15)     tosenkei = tosenkei + 1
(16)     Hikaku[maxi] = 切り捨て(［タ］/［チ］)
(17) k を 0 から［ア］まで 1 ずつ増やしながら繰り返す:
(18)     表示する(Tomei[k],"：",Tosen[k],"名")
```

図9　各政党の当選者数を求めるプログラム

先　生：できたようだね。各政党の当選者数は求められたけど，政党によっては候補者が足りない場合もあるから，その場合にも対応してみよう。図11のように各政党の候補者数を格納する配列 Koho を追加してみたらどうだろう。例えば，C党の候補者が足りなくなるように設定してみよう。

```
A党:2名
B党:1名
C党:3名
D党:0名
```

図10　各政党の当選者数の表示

i	0	1	2	3
Koho	5	4	2	3

図11　候補者数を格納する配列

Mさん：候補者が足りなくなったらどういう処理をすれば良いのですか？

先　生：比較した得票で次に大きい得票数の政党が繰り上がって議席を取るんだよ。

Mさん：なるほど。では，図9の(11)行目の条件文を次のように修正すればいいですね。当選していない候補者はどこかの政党には必ずいるという前提だけど。

(11)　　もし max < Hikaku[i] [ツ] [テ] ならば:

Kさん：先生，候補者が不足するほかに，考えるべきことはありますか？

先　生:例えば,配列Hikakuの値が同じになった政党の数が残りの議席の数より多い場合，このプログラムでは添字の小さい政党に議席が割り当てられてしまうので不公平だね。実際には，この場合はくじ引きで議席を割り当てるようだよ。

[セ], [タ]・[チ]の解答群

⓪ max	① maxi	② tosenkei
③ Tokuhyo[maxi]	④ Tokuhyo[maxi] + 1	⑤ Tokuhyo[max]
⑥ Tosen[maxi]	⑦ Tosen[maxi + 1]	⑧ (Tosen[maxi] + 1)

[ソ]の解答群

⓪ max = max + 1	① max = Tokuhyo[i]	② max = Hikaku[i]
③ Hikaku[i] = max	④ Tokuhyo[i] = max	⑤ Tokuhyo[i] = Hikaku[i]

[ツ]の解答群

⓪ and	① or	② not

[テ]の解答群

⓪ Koho[i] >= Tosen[i] + 1	① Koho[i] < Tosen[i] + 1
② Koho[i] >= Tosen[i]	③ Koho[i] < Tosen[i]