

令和8年度 情報学群 総合型選抜 B区分

プログラミング 1/7

次の文章を読み、後の問1～5のすべてに答えなさい。

重さがまちまちである多数の荷物がある。アキラ、イズミ、ウキョウの3人で分担してそれらの荷物を運ぶことになった。なるべく、それぞれの運ぶ荷物の重さの合計が3人とも近い値になるようにしたいとする。(担当する荷物の個数は均等でなくてもよい。)

荷物は  $n$  個あり、重さの小さい順に荷物0, 荷物1, ..., 荷物  $n-1$  と呼ぶ。各荷物の重さは0より大きい。荷物の重さを格納した配列 Nimotu が与えられる。Nimotu の第  $i$  要素 ( $i$  は0以上  $n$  未満) は荷物  $i$  の重さを格納している。下記の表1は、 $n=9$  のときの配列 Nimotu の例である。

表1: 配列 Nimotu の例 ( $n=9$ )

添字	0	1	2	3	4	5	6	7	8
Nimotu	2	2	4	5	6	6	6	7	10

問1  $n$  が3の倍数と仮定する。荷物を分配する方法として以下の方針1を考えた。

方針1 荷物0, 1, ...,  $n-1$  の順に以下の規則に沿って割り当てる。

- 最初の3個を、アキラ、イズミ、ウキョウの順に割り当てる。
- 次の3個を、ウキョウ、イズミ、アキラの順に割り当てる。
- 以降、荷物がなくなるまで、上の1と2を交互に繰り返す。

次ページのプログラム1は、方針1に従ってアキラに割り当てられる荷物の重さの合計を出力するプログラムである。ただし、(04)行目の式「 $i \% 6$ 」は、 $i$  の値を6で割ったときの余りを表す。以下の文章の空欄 **ア** ~ **エ** に入る数を答えなさい。ただし、空欄 **ア**・**イ** の解答の順序は問わない。また、プログラム1の空欄 **オ** に入る語句を、プログラム1の下の解答群の中から一つ選びなさい。

- 方針1に従うと、荷物  $i$  ( $i$  は0以上  $n$  未満) がアキラに割り当てられるのは、 $i$  を6で割った余りが **ア** または **イ** のときである。

令和8年度 情報学群 総合型選抜 B区分

プログラミング 2/7

- 表1の Nimotu と  $n = 9$  が与えられたとき、アキラに割り当てられる荷物の重さの合計は **ウ**，ウキョウに割り当てられる荷物の重さの合計は **エ** である。

プログラム 1

- (01) 配列 Nimotu と  $n$  の値を外部から受け取る
- (02) akira = 0
- (03)  $i$  を 0 から  $n - 1$  まで 1 ずつ増やしながらか繰り返す:
- (04) | amari =  $i \% 6$
- (05) | もし amari == **ア** or amari == **イ** ならば:
- (06) | | akira = akira + **オ**
- (07) 表示する ("アキラの荷物の重さの合計は", akira)

**オ** の解答群

- ①  $n$     ② 1    ③  $i$     ④ amari    ⑤ akira    ⑥ Nimotu[ $n - 1$ ]  
⑦ Nimotu[ $i$ ]    ⑧ Nimotu[amari]    ⑨ Nimotu[akira]

問2 荷物を分配する方法として以下の方針2を考えた。なお、 $n$  は必ずしも3の倍数でなくてもよいとする。

方針2  $i = 0, 1, \dots, n - 1$  の順に荷物  $i$  を以下の規則に沿って割り当てる。

1. 3人のうち、割り当て済みの荷物の重さの合計が最小の人に、荷物  $i$  を割り当てる。該当する人が複数いる場合は、その中で番号が最小の人に割り当てる。ただし、各人の番号は、アキラ、イズミ、ウキョウの順に 0, 1, 2 とする。
2. 荷物がなくなるまで、上の1を繰り返す。

方針2に従うとき、例えば割り当て済みの荷物の重さの合計が3人とも等しい場合は、アキラに次の荷物を割り当てることに注意しなさい。

令和8年度 情報学群 総合型選抜 B区分

プログラミング 3/7

下記のプログラム2は、方針2に従ってアキラ、イズミ、ウキョウにそれぞれ割り当てられる荷物の重さの合計を出力するプログラムである。以下の文章の空欄 **カ** ~ **ク** に入る数を答えなさい。また、プログラム2の空欄 **ケ**・**コ** に入る語句を、プログラム2の下それぞれの解答群の中から一つずつ選びなさい。

- 表1の Nimotu と  $n = 9$  が与えられたとき、方針2に従って荷物を割り当てると、アキラ、イズミ、ウキョウに割り当てられる荷物の重さの合計はそれぞれ **カ** , **キ** , **ク** である。

```
プログラム2
(01) 配列 Nimotu と n の値を外部から受け取る
(02) akira = 0 , izumi = 0 , ukyo = 0
(03) i を 0 から n - 1 まで 1 ずつ増やしながら繰り返す:
(04)   もし akira ケ izumi コ akira ケ ukyo ならば:
(05)   |   akira = akira + オ
(06)   |   そうでなくもし izumi < akira コ izumi <= ukyo ならば:
(07)   |   |   izumi = izumi + オ
(08)   |   そうでなければ:
(09)   |   |   ukyo = ukyo + オ
(10) 表示する ("アキラ, イズミ, ウキョウの荷物の重さの合計はそれぞれ",
           akira, "と", izumi, "と", ukyo)
```

**ケ** の解答群

① ==      ② !=      ③ >      ④ <      ⑤ >=      ⑥ <=

**コ** の解答群

① and      ② or      ③ not

令和8年度 情報学群 総合型選抜 B区分

プログラミング 4/7

問3 荷物の重さに関して、最初に述べた仮定（「荷物  $0, 1, \dots, n-1$  は重さの小さい順に並んでおり、かつ、各荷物の重さは0より大」）のみを仮定する。つまり、

$$0 < \text{Nimotu}[0] \leq \text{Nimotu}[1] \leq \dots \leq \text{Nimotu}[n-1] \quad \dots\dots \textcircled{1}$$

である。

$n \geq 5$  とし、問2の方針2に従って荷物を割り当てるものとする。このとき、荷物0は必ずアキラに割り当てられる。以下の (i) ~ (iv) に答えなさい。

- (i) 荷物1も必ず決まった人に割り当てられる。それが誰かを理由とともに答えなさい。
- (ii) 荷物2も必ず決まった人に割り当てられる。それが誰かを理由とともに答えなさい。
- (iii) 荷物3も必ず決まった人に割り当てられる。それが誰かを理由とともに答えなさい。
- (iv) 荷物4も必ず決まった人に割り当てられる。それが誰かを理由とともに答えなさい。

問4 方針2を変更して、荷物  $0, 1, \dots, n-1$  の順に割り当てる代わりに荷物  $n-1, n-2, \dots, 1, 0$  の順に割り当てるようにし、これを方針2改と呼ぶ（下記）。

方針2改  $i = n-1, n-2, \dots, 1, 0$  の順に荷物  $i$  を以下の規則に沿って割り当てる。

1. 3人のうち、割り当て済みの荷物の重さの合計が最小の人に、荷物  $i$  を割り当てる。該当する人が複数いる場合は、その中で番号が最小の人に割り当てる。ただし、各人の番号は、アキラ、イズミ、ウキョウの順に  $0, 1, 2$  とする。
2. 荷物がなくなるまで、上の1を繰り返す。

プログラム2の(03)行目を「 $i$  を  $n-1$  から  $0$  まで  $1$  ずつ減らしながら繰り返す:」に変えることで、方針2改に従って各人に割り当てられる荷物の重さの合計を出力するプログラムになる。このプログラムをプログラム2改と呼ぶ（次ページ）。

プログラム 2改

(01) 配列 Nimotu と n の値を外部から受け取る

(02) akira = 0 , izumi = 0 , ukyo = 0

(03) i を n - 1 から 0 まで 1 ずつ減らしながら繰り返す:

(04)   もし akira  izumi  akira  ukyo ならば:

(05)   |   akira = akira +

(06)   |   そうでなくもし izumi < akira  izumi <= ukyo ならば:

(07)   |   izumi = izumi +

(08)   |   そうでなければ:

(09)   |   ukyo = ukyo +

(10) 表示する ("アキラ, イズミ, ウキョウの荷物の重さの合計はそれぞれ",  
akira, "と", izumi, "と", ukyo)

- (1) プログラム 2改に表 1 の Nimotu と  $n = 9$  が入力されたときの動作について、以下の (i)・(ii) に答えなさい。

表 1: 配列 Nimotu の例 ( $n = 9$ ) (再掲)

添字	0	1	2	3	4	5	6	7	8
Nimotu	2	2	4	5	6	6	6	7	10

- (i) (03) ~ (09) 行目の繰り返しの内部の処理を 1 回実行した直後, 2 回実行した直後, 3 回実行した直後, 4 回実行した直後の変数 akira, izumi, ukyo の値をそれぞれ答えなさい。
- (ii) (10) 行目に表示される内容は「アキラ, イズミ, ウキョウの荷物の重さの合計はそれぞれ  と  と 」である。空欄  ~  に入る数を答えなさい。
- (2)  $n \geq 4$  とする。荷物の重さに関して、最初に述べた仮定 (問 3 の式①が表す条件) が成り立っているとする。方針 2改に従って各荷物を順に割り当てるとき、荷物  $n - 4$  がアキラに割り当てられるのは、Nimotu の要素がどのような条件を満たすときか。同様に、荷物  $n - 4$  がイズミに割り当てられるのはどのようなときか。それぞれの必要十分条件を、式または文章で答えなさい。

令和8年度 情報学群 総合型選抜 B区分

プログラミング 6/7

問5 「それぞれの運ぶ荷物の重さの合計が3人とも近い値になるようにする」ことをやめて、アキラは、自分の担当する荷物の重さの合計が自分の決めた目標値以下でかつ最大になるように荷物を選ぶことにした。そこで、配列 `Nimotu` と値 `n` に加えて目標値 `mokuhyo` を入力として取り、0個以上の荷物の重さの合計としてあり得る値のうち `mokuhyo` 以下の最大の値を出力するプログラム（次ページのプログラム3）を作った。なお、この間では、荷物の重さはすべて正の整数であり、すべての荷物の重さの合計は1000を超えないとする。また、`mokuhyo` の値は1000以下の正の整数とする。

プログラム3の空欄 `セ` ~ `ツ` に入る語句を、プログラム3の下のそれぞれの解答群の中から一つずつ選びなさい。ただし、同じ語句を繰り返し選択してもよい。

【プログラム3の説明】

プログラム3で使われている `Tukureru` は1001要素の配列である。(08)行目の実行直前の時点で `Tukureru` が以下を満たすようになっている。

0以上1000以下の整数  $x$  に対して、 $x$  が0個以上の荷物の重さの合計としてあり得るならば `Tukureru[x] = 1`、そうでなければ `Tukureru[x] = 0` ……②

そして、(03)~(07)行目は以下の考えに従って作られている。ただし、空欄 `ソ` はプログラム3の(07)行目の空欄 `ソ` と同じものが入る。

1. どの荷物も使わないことで、重さの合計を0にできる((03)行目)。
2. 0以上  $n$  未満の  $i$  について、番号が  $i-1$  以下の荷物のうち0個以上を使って重さの合計を  $x$  にできるならば、番号が  $i$  以下の荷物のうち0個以上を使って重さの合計を  $x$  または `ソ` にできる。

(08)~(11)行目は、上記の条件②を満たす `Tukureru` を使って、(12)行目の実行直前の時点で `saidai` の値が「0個以上の荷物の重さの合計としてあり得る値のうち `mokuhyo` 以下で最大のもの」になるように作られている。

プログラミング 7/7

プログラム 3

- (01) 配列 Nimotu と n と mokuhyo の値を外部から受け取る
- (02) Tukureru のすべての値を 0 にする
- (03) Tukureru[0] = 1
- (04) i を 0 から n - 1 まで 1 ずつ増やしながら繰り返す:
- (05) | x を 1000 から 0 まで 1 ずつ減らしながら繰り返す:
- (06) | | もし Tukureru[x]  セ 0 ならば:
- (07) | | | Tukureru[  ソ ] = 1
- (08) saidai = 0
- (09) x を  タ 繰り返す:
- (10) | もし Tukureru[x]  チ 0 ならば:
- (11) | | saidai =  ツ
- (12) 表示する ("荷物の重さの合計のうち目標値以下で最大は", saidai)

セ ·  ソ ·  チ ·  ツ の解答群

- ① n                    ② 1                    ③ i                    ④ ==                    ⑤ !=
- ⑥ Nimotu[i]                    ⑦ Nimotu[x]                    ⑧ Nimotu[i + x]
- ⑨ x + Nimotu[i]

タ の解答群

- ① 0 から 1000 まで 1 ずつ増やしながら
- ② 1000 から 0 まで 1 ずつ減らしながら
- ③ 0 から mokuhyo まで 1 ずつ増やしながら
- ④ mokuhyo から 0 まで 1 ずつ減らしながら