

# 令和4年度後期 情報検定

<実施 令和5年2月12日（日）>

## プログラミングスキル

(説明時間 10:00~10:10)

(試験時間 10:10~11:40)

- ・試験問題は試験開始の合図があるまで開かないでください。
- ・解答用紙（マークシート）への必要事項の記入は、試験開始の合図と同時に行いますので、それまで伏せておいてください。
- ・試験開始の合図の後、次のページを開いてください。＜受験上の注意＞が記載されています。必ず目を通してから解答を始めてください。
- ・試験問題は、すべてマークシート方式です。正解と思われるものを1つ選び、解答欄の○をHBの黒鉛筆でぬりつぶしてください。2つ以上ぬりつぶすと、不正解になります。
- ・辞書、参考書類の使用および筆記用具の貸し借りは一切禁止です。
- ・電卓の使用が認められます。ただし、下記の機種については使用が認められません。

### <使用を認めない電卓>

1. 電池式（太陽電池を含む）以外の電卓
2. 文字表示領域が複数行ある電卓（計算状態表示の一行は含まない）
3. プログラムを組み込む機能がある電卓
4. 電卓が主たる機能ではないもの
  - \* パソコン（電子メール専用機等を含む）、携帯電話（PHS）、スマートフォン、タブレット、電子手帳、電子メモ、電子辞書、翻訳機能付き電卓、音声応答のある電卓、電卓付き腕時計、時計型ウェアラブル端末等
5. その他試験監督者が不適切と認めるもの

## ＜受験上の注意＞

1. この試験問題は32ページあります。ページ数を確認してください。  
乱丁等がある場合は、手をあげて試験監督者に合図してください。  
※問題を読みやすくするために空白ページを設けている場合があります。
2. 解答用紙（マークシート）に、受験者氏名・受験番号を記入し、受験番号下欄の数字をぬりつぶしてください。正しく記入されていない場合は、採点されませんので十分注意してください。
3. 試験問題についての質問には、一切答えられません。自分で判断して解答してください。
4. 試験中の筆記用具の貸し借りは一切禁止します。筆記用具が破損等により使用不能となった場合は、手をあげて試験監督者に合図してください。
5. 試験を開始してから30分以内は途中退出できません。30分経過後退出する場合は、もう一度、受験番号・マーク・氏名が記載されているか確認して退出してください。なお、試験終了5分前の合図以降は退出できません。試験問題は各自お持ち帰りください。
6. 試験後の合否結果（合否通知）、および合格者への「合格証・認定証」はすべて、Web認証で行います。
  - ①情報検定（J検）Webサイト合否結果検索ページ及びモバイル合否検索サイト上で、デジタル「合否通知」、デジタル「合格証・認定証」が交付されます。
  - ②団体宛には合否結果一覧ほか、試験結果資料一式を送付します。
  - ③合否等の結果についての電話・手紙等でのお問い合わせには、一切応じられませんので、ご了承ください。

## <問題の構成>

必須問題 全員解答

問題 1 ～ 問題 4	2 ページ～15 ページ
-------------------------	--------------

選択問題 次の問題から 1 問選択し解答せよ。

(選択した問題は解答用紙「選択欄」に必ずマークすること)

※選択欄にマークがなく、解答のみマークした場合は採点を行いません。

・ C 言語の問題	17 ページ～19 ページ
・ 表計算の問題	20 ページ～28 ページ
・ アセンブラの問題	29 ページ～32 ページ

## 必須問題

問題 1 次の配列処理に関する記述を読み、各設問に答えよ。

一次元配列に対してデータを格納する方法を考える。今回、データを格納するさいに使用する方法はハッシュ法を用いる。

[ハッシュ法について]

一次元配列 A にデータを格納する位置を決定する。ここでは、配列 A に格納するデータはすべて正の整数とし、配列 A の要素数は 11 (A[0]~A[10]) とする。配列 A の未使用領域には -1 が格納されている。

データの格納方法については次の手順で行う。

- ① 格納したい値を要素数で割った余りを  $i$  とする。
- ② A[i] が未使用の場合は A[i] へ値を格納する。A[i] が使用中の場合は、A[i+1] から未使用の場所を順次探索し格納する。ただし、配列 A の最後まで探索し未使用の場所がない場合は、配列 A の先頭から順次未使用の場所を探索していく。

<設問 1> 配列 A が図 1 の状態のとき、次の記述中の  に入れるべき適切な字句を解答群から選べ。

	0	1	2	3	4	5	6	7	8	9	10
配列 A	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

図 1 配列 A の初期状態

図 1 の状態から、56 → 92 → 8 → 81 → 38 → 76 → 43 の順にデータを格納していくとき、92 の格納場所は  (1) になる。また、43 の格納場所は  (2) になる。データを格納する際に、格納したい位置に既にデータが存在する場合、シノニムが発生したという。データをすべて格納する際にシノニムは  (3) 回発生したことになる。また、すべてのデータを格納した配列 A は  (4) の状態になっている。

(1) ~ (2) の解答群

- |         |         |         |
|---------|---------|---------|
| ア. A[0] | イ. A[2] | ウ. A[4] |
| エ. A[6] | オ. A[8] | カ. A[9] |

(3) の解答群

- |      |      |      |      |
|------|------|------|------|
| ア. 1 | イ. 2 | ウ. 3 | エ. 4 |
|------|------|------|------|

(4) の解答群

ア. 

-1	56	-1	-1	81	92	38	-1	8	43	76
----	----	----	----	----	----	----	----	---	----	----

イ. 

-1	56	-1	8	38	92	81	-1	-1	76	43
----	----	----	---	----	----	----	----	----	----	----

ウ. 

43	56	-1	-1	92	81	38	-1	8	-1	76
----	----	----	----	----	----	----	----	---	----	----

エ. 

43	56	-1	-1	92	38	81	-1	8	-1	76
----	----	----	----	----	----	----	----	---	----	----

<設問 2> 次のシノニムに関する記述中の  に入れるべき適切な字句を解答群から選べ。

格納するデータが 1 ~ 100,000 の範囲でランダムに発生するデータ 4 件を、ハッシュ法を使用して要素数が 10 である配列 B に格納していく。配列 B は図 2 の状態から開始する。

	0	1	2	3	4	5	6	7	8	9
配列 B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

図 2 配列 B の初期状態

データを格納していく際に、シノニムが発生する確率は次の手順で求めることができる。最初に確率の分子を考えていく。1 つ目のデータを格納するとき、シノニムが発生しないパターンで配列 B に格納する場合、10 通りとなる。次に 2 つ目のデータを格納するとき、シノニムが発生しないパターンで配列 B に格納する場合、既に 1 つ目のデータが格納されていることを考慮すると  (5) 通りとなる。3 つ目、4 つ目のデータも同様にシノニムが発生しないパターンで何通りかを算出し、すべての値を掛け合わせる。

次に確率の分母を考えていく。分母はすべてのパターンになるため、1 つ目のデータから 4 つ目のデータまですべて 10 通りである。そのため、 $10 \times 10 \times 10 \times 10 = 10,000$  となる。

シノニムが発生する確率は次の式で求めることができる。

$$\text{シノニムが発生する確率} = 1 - \text{シノニムが発生しない確率}$$

よって、今回シノニムが発生する確率は  (6) である。

(5) の解答群

ア. 6

イ. 7

ウ. 8

エ. 9

(6) の解答群

ア. 0.302

イ. 0.496

ウ. 0.554

エ. 0.72

問題2 次の流れ図の説明を読み、各設問に答えよ。

[流れ図の説明]

二つの自然数  $a$ ,  $b$  について、 $a$  を  $b$  で割ったときの商を  $q$ , 余りを  $r$  とすると「 $a$  と  $b$  の最大公約数」は、「 $b$  と  $r$  の最大公約数」に等しいという定理がある。

この定理を使って、二つの自然数の最大公約数を求める方法がユークリッドの互除法であり、その手順と流れ図を次に示す。なお、 $\text{MOD}(x, y)$  は  $x$  を  $y$  で割ったときの余りを求める関数である。

[ユークリッドの互除法の手順]

- ①  $a \div b$  の商を  $q$ , 余りを  $r$  とする。
- ②  $r=0$  ならば  $b$  が最大公約数となり手順を終了する。  
 $r \neq 0$  ならば  $b$  を  $a$  に、 $r$  を  $b$  にして①へ戻る。

[ユークリッドの互除法の流れ図]

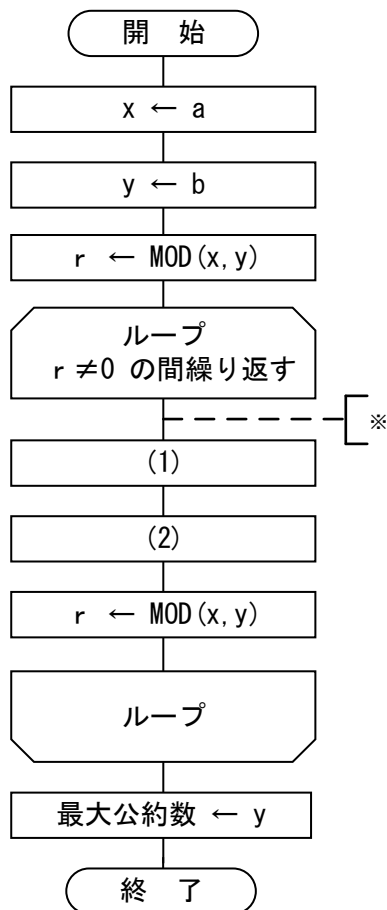


図1 ユークリッドの互除法の流れ図

<設問 1> 図 1 の流れ図中の  に入れるべき適切な字句を解答群から選べ。

(1) , (2) の解答群

ア.  $x \leftarrow q$

イ.  $x \leftarrow r$

ウ.  $x \leftarrow y$

エ.  $y \leftarrow q$

オ.  $y \leftarrow r$

カ.  $y \leftarrow x$

<設問 2> 次の流れ図のトレースに関する記述中の  に入れるべき適切な字句を解答群から選べ。

$a=850$ ,  $b=365$  として図 1 の流れ図を実行すると、流れ図中の※を 2 回目に通過するとき、 $x$  の値は  (3) であり、 $y$  の値は  (4) である。

(3) , (4) の解答群

ア. 5

イ. 60

ウ. 120

エ. 365

オ. 650

カ. 850

<設問 3> 次の最小公倍数に関する記述中の  に入れるべき適切な字句を解答群から選べ。

二つの自然数  $a$ ,  $b$  の最小公倍数は「 $a \times b \div (a, b \text{ の最大公約数})$ 」で求めることができる。設問 2 と同じ  $a=850$ ,  $b=365$  のとき最小公倍数は  (5) である。

(5) の解答群

ア. 1825

イ. 4250

ウ. 62050



問題3 次の文字列の操作に関する記述を読み、各設問に答えよ。

文字列を扱う処理では、データそのもの以外に文字数や部分文字列を扱うことがある。そこで、文字列を扱うクラスを作成し、必要な情報を返すメソッドを用意した文字列の処理を考える。なお、ここで用いる配列の要素番号は 0 から始まるものとし、配列を操作する上で十分な領域が確保されているものとする。

<設問1> 次の文字列を扱うクラス WordClass に関する記述中の  に入れるべき適切な字句を解答群から選べ。

次のようなクラス WordClass を考える。クラス WordClass のメンバ変数とメソッドは表1のとおりである。なお、メンバ変数を外部から直接参照することはできないものとし、メソッドは外部から参照できるものとする。

表1 WordClass のメンバ変数とメソッド

メンバ変数	説明
len	配列 word に格納された文字数を格納する。
word[]	文字列を格納する配列で、1要素に1文字を格納する。

メソッド	説明
getLen()	メンバ変数の len の値を返す。
getChar(idx)	メンバ変数 word の idx の位置に格納されている文字を返す。

WordClass をインスタンス化する時には必ず引数で文字列を与えるものとする。

インスタンス化した時に動作するメソッドをコンストラクタと呼ぶが、クラス WordClass のコンストラクタはメンバ変数 len と word[] に引数で与えられた情報を格納する。例えば、次の[処理]のようにした場合は、メンバ変数 len には 3、 (1) には順番に "a", "b", "c" を格納する。

[処理] word ← WordClass("abc")

※ word はインスタンスを参照する変数となる

インスタンスを参照する時は、ドット演算子を用いてメンバ変数やメソッドを指定する。例えば、インスタンスを参照する変数が word でありメンバ変数 len を参照するには次のようになる。

[インスタンスを参照する処理] word.len

ただし、クラス WordClass はメンバ変数を外部から直接参照できないので、メンバ

変数 len の値を取得するために getLen() メソッドを利用して次のように処理する。

[インスタンスのメンバ変数 len の値を取得する処理] word.getLen()

このように、データや操作を一つにまとめて内部を秘匿し、メンバ変数などを外部から直接参照できないようにすることを (2) と呼ぶ。

(1) の解答群

- ア. word[0]~word[2]
- ウ. word[1]~word[2]

- イ. word[0]~word[3]
- エ. word[1]~word[3]

(2) の解答群

- ア. インヘリタンス
- ウ. 特化

- イ. カプセル化
- エ. 汎化

<設問 2> 次の流れ図の説明を読み、流れ図中の  に入れるべき適切な字句を解答群から選べ。

[流れ図の説明]

設問 1 のクラス WordClass のインスタンスを 2 つ受け取り、文字列の大小関係を返す comp である。comp の仕様は表 2 のとおりである。なお、文字の大小は文字コードにより決まる。

表 2 comp の仕様

書式	comp(インスタンス 1, インスタンス 2)
返却値	インスタンス 1 の文字列が小さければ -1, 同じ文字列であれば 0, インスタンス 1 の文字列が大きければ 1。

比較はインスタンスのメンバ変数 word[] の先頭から順番に同じ要素の位置にある文字同士で行い、異なった文字の場合に大小関係で判断する。最後まで同じであれば同じ文字列と判断する。また、次の図 1 のような場合は、4 文字目まで一致するがインスタンス 1 の文字列には 5 文字目が無いため、インスタンス 1 の文字列が小さいと判断して -1 を返却する。

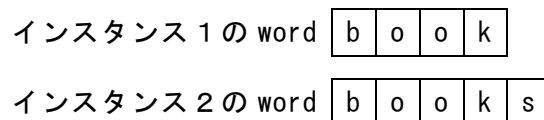
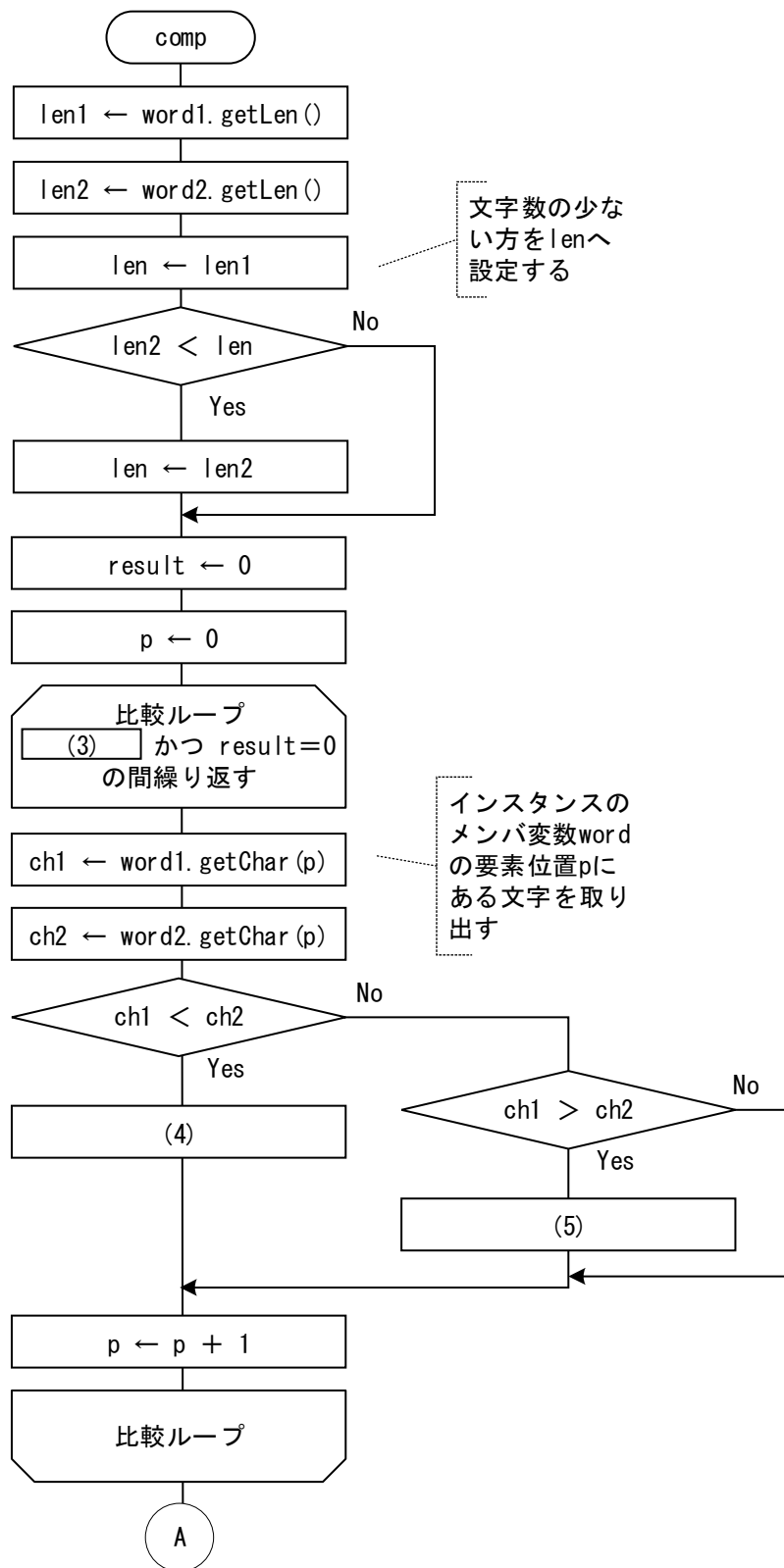


図 1 文字列の大小を判定する例

また、流れ図ではインスタンス 1 を参照する変数を word1、インスタンス 2 を参照する変数を word2 とする。

[流れ図]



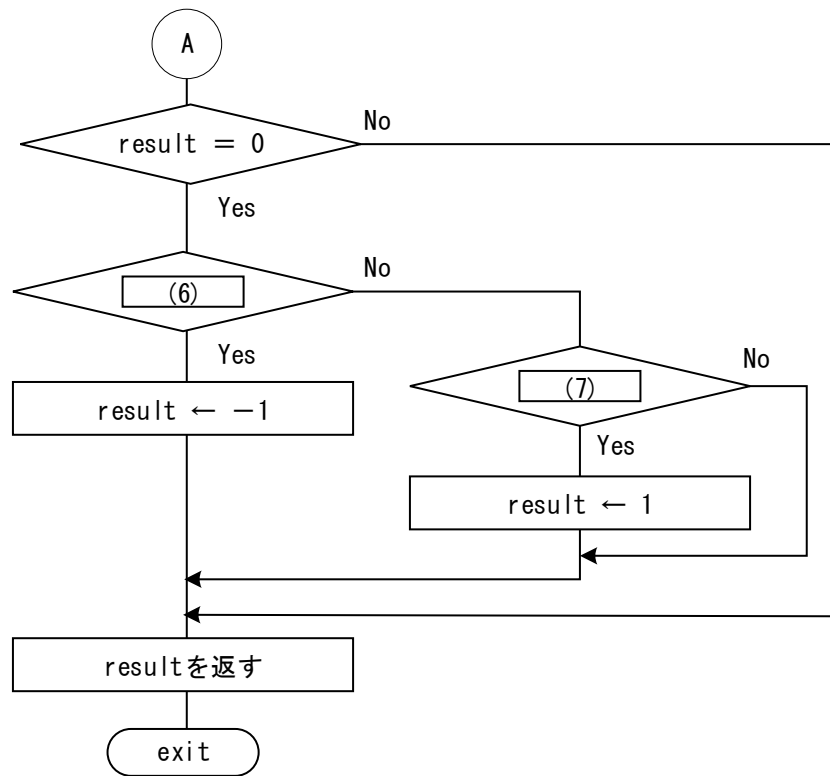


図2 comp の流れ図

(3) の解答群

ア.  $p \leq \text{len}$

ウ.  $p \geq \text{len}$

イ.  $p < \text{len}$

エ.  $p > \text{len}$

(4) , (5) の解答群

ア.  $\text{result} \leftarrow -1$

ウ.  $\text{result} \leftarrow 1$

イ.  $\text{result} \leftarrow 0$

エ.  $\text{result} \leftarrow p$

(6) , (7) の解答群

ア.  $\text{len1} < \text{len2}$

ウ.  $\text{len1} > \text{len2}$

イ.  $\text{len1} = \text{len2}$

エ.  $\text{len1} + p = \text{len2}$

<設問 3 > 次の流れ図の説明を読み、流れ図中の  に入れるべき適切な字句を解答群から選べ。

[流れ図の説明]

設問 2 の comp を利用して、挿入法により文字列を昇順に並べ替える流れ図である。

挿入法は、以下の手順①～⑤の k について 1～n-1 まで変化させながら行う。ここでは、配列を ary, 配列の要素数を n とする。

- ① ary[k]の内容をsvに退避する。
- ② jをk-1とする。
- ③ ary[j]の文字列がsv以下であれば⑤に進む
- ④ ary[j]の内容をary[j+1]へ格納してjから1を引き, ③へ戻る。
- ⑤ svをary[j+1]へ格納する。

流れ図では、配列 ary にはクラス WordClass のインスタンスを格納しており、変数 sv はインスタンスを格納する作業用の変数として用いる。また、配列 ary の各要素のインスタンスには文字列が格納済みであり、配列の要素数は変数 n に格納されている。

(8) , (9) の解答群

ア. flag ← 0

イ. flag ← 1

ウ. j ← j - 1

エ. j ← j + 1

(10) の解答群

ア. ary[k] ← sv

イ. ary[k+1] ← sv

ウ. ary[j] ← sv

エ. ary[j+1] ← sv

[流れ図]

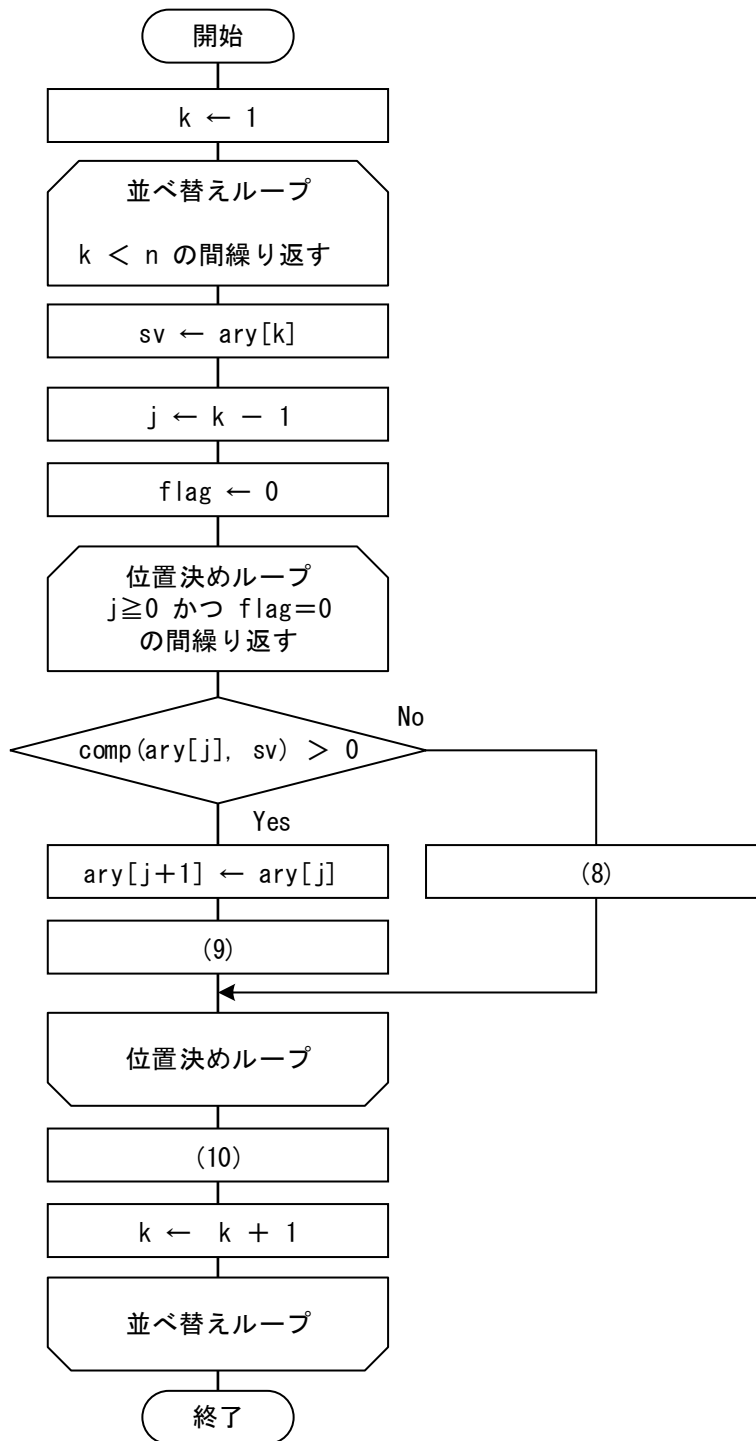


図3 並べ替えをする流れ図

問題4 次のプログラムの説明を読み、各設問に答えよ。

[プログラムの説明]

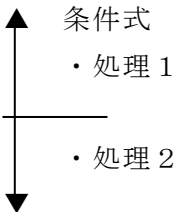
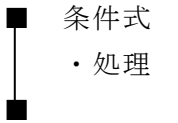
正の整数  $n$  について、 $1 \sim n$  の和  $F(n)$  を求めるプログラム Sum\_1 である。  
 $F(n)$  は次の式で求めることができる。

$$F(n) = n + (n-1) + \dots + 1 \quad (n \geq 1)$$

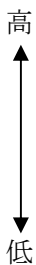
例えば、 $F(5) = 5 + 4 + 3 + 2 + 1 = 15$  であり、15 を返す。

なお、 $n$  が 0 以下の場合は -1 を返す。

[擬似言語の記述形式の説明]

記述形式	説明
○	手続き、変数などの名前、型などを宣言する。
・変数 ← 式	変数に式の値を代入する。
/* 文 */	注釈を記述する。
	選択処理を示す。 条件式が真の時は処理 1 を実行し、 偽の時は処理 2 を実行する。
	前判定繰り返し処理を示す。 条件式が真の間、処理を実行する。

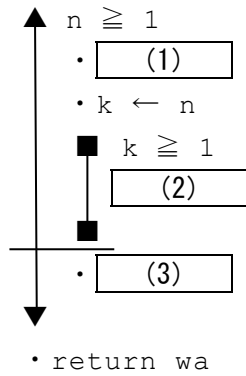
[演算子と優先順位]

演算の種類	演算子	優先順位
単項演算	+, -, not	
乗除演算	*, /, %	
加減演算	+, -	
関係演算	>, <, ≥, ≤, =, ≠	
論理積	and	
論理和	or	

注記 整数同士の除算では、整数の商を結果として返す。%演算子は剰余算を表す。

[プログラム]

```
○Sum_1( 整数型 : n )  
○整数型 : k, wa  
/* 1 ~ n の和を求める */
```



<設問 1> プログラム中の  に入れるべき適切な字句を解答群から選べ。

(1), (3) の解答群

ア. wa ← -1

イ. wa ← 0

ウ. wa ← 1

エ. wa ← k

(2) の解答群

ア. k ← k - 1

イ. k ← k + 1

・ wa ← wa + k

・ wa ← wa + k

ウ. wa ← wa + k

エ. wa ← wa + k

・ k ← k - 1

・ k ← k + 1

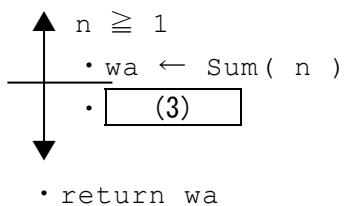
F(n)は次のように再帰的に表現することができる。

$F(n) = n + F(n-1)$  ( $n \geq 2$ ) ただし,  $n=1$  のときは,  $F(n) = n$  とする。

上記に基づいて, プログラム Sum\_2 を作成した。なお,  (3) にはプログラム Sum\_1 と同じものが入る。

[プログラム]

```
○Sum_2( 整数型 : n )  
○整数型 : wa  
/* 1 ~ n の和を求める */
```

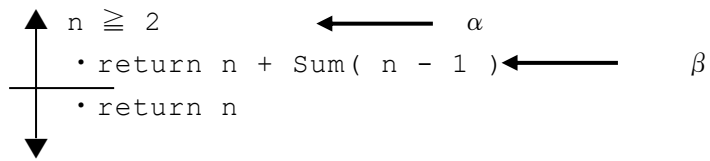




○Sum ( 整数型 : n )

○整数型 : wa

/\* 再帰処理 \*/



<設問 2 > 次のプログラム Sum\_2 の実行に関する説明文の  に入れるべき適切な字句を解答群から選べ。

n が 5 の場合、プログラム中の  $\alpha$  を実行するときの変数 n をトレースすると順に、 (4) となる。また、プログラム中の  $\beta$  を  (5) に変更すると、n が奇数の場合は奇数の和を、n が偶数の場合は偶数の和を求めることができる。

(4) の解答群

順番	1	2	3	4	5
ア.	0	1	2	3	4
イ.	1	2	3	4	5
ウ.	4	3	2	1	0
エ.	5	4	3	2	1

(5) の解答群

ア. return n - 2 + Sum( n - 2 )    イ. return n - 1 + Sum( n - 1 )  
ウ. return n + Sum( n - 2 )        エ. return n + 1 + Sum( n - 1 )

## < 選 択 問 題 >

選択問題は問題から1つ選択し解答せよ。

選択した問題は必ず、解答用紙「選択欄」にマークすること。

※選択欄にマークがなく、解答のみの場合は採点を行いません。

各構成は以下のとおり。

### 選択問題

- |            |               |
|------------|---------------|
| ・ C 言語の問題  | 17 ページ～19 ページ |
| ・ 表計算の問題   | 20 ページ～28 ページ |
| ・ アセンブラの問題 | 29 ページ～32 ページ |

魔方陣に関する次の記述を読み、各設問に答えよ。

[魔方陣について]

魔方陣とは、正方形のマス目において縦・横・対角線の並びに配置した数の合計が同じになるものである。N×Nのマスであれば、配置する数は1からN<sup>2</sup>までの数をすべて配置する。なお、Nは3以上の奇数とする。図はNが3の場合の魔方陣の例である。

4	9	2
3	5	7
8	1	6

図 Nが3の場合の魔方陣の例

[魔方陣の作成手順]

- ① N×Nの全てのマスに0を格納する。
- ② 最下行の中央のマスに最初の数である1を格納する。この位置を現在位置とする。
- ③ 以下の④～⑥をすべてのマスが埋まるまで繰り返す。
- ④ 現在位置の右下のマスが0かどうか確認する。このとき、最下行(N-1行)の下は最上行(0行)とし、最右列(N-1列)の右は最左列(0列)とする。
- ⑤ ④で確認したマスが0の場合は、そこを新しい現在位置とし、そうでない場合は現在位置の上のマス新しい現在位置とする。
- ⑥ マスに格納する数を一つ増やし、現在位置にその数を配置する。

<設問1> 次の魔方陣に関する記述中の□に入れるべき適切な字句を解答群から選べ。

魔方陣では、縦・横・対角線の各並びの数は合計が同じである。例えば、図のような3×3のマス目では15である。次に、N×Nのマス目の場合の和を考える。

N×Nの全マスの合計は、1～N<sup>2</sup>の和となるので次の式1で表せる。

$$N^2 \times (N^2 + 1) \div 2 \quad \dots \text{式1}$$

これがN行、またはN列で構成されているので、式1を□(1)で割った値が縦・横・対角線の各並びの数の合計であり、次の式2で表せる。

$$(N^3 + N) \div 2 \quad \dots \text{式2}$$

(1) の解答群

- ア. N                      イ. N + 1                      ウ. N<sup>2</sup>                      エ. N<sup>2</sup> + 1

<設問 2> 次のプログラムの説明を読み、プログラム中の  に入れるべき適切な字句を解答群から選べ。

[プログラムの説明]

魔方陣の1辺のマスの数Nを受け取り、魔方陣の作成手順に従って魔方陣を作成して表示する関数 mahojin である。なお、Nが偶数の場合とNが20以上の場合はエラーメッセージを表示して戻るものとする。

[関数の説明]

mahojin 関数

引 数 : n(1辺のマス数: 整数値)

機 能 : 魔方陣を表示する。ただし、nが偶数の場合と20以上の場合はエラーメッセージを表示して戻る。

戻り値 : なし。

[プログラム]

```
void mahojin(int n) {
    int maho[n][n]; /* 魔方陣を格納する配列 */
    int row, col, svRow, svCol, val;
    /* nに誤りが無いかを調べる */
    if (  (2) ) {
        printf("マスの数に誤りがあります。¥n");
        return;
    }
    /* 配列を初期化する */
    for(row = 0; row < n; row++) {
        for(col = 0; col < n; col++) {
            maho[row][col] = 0;
        }
    }
    /* 魔方陣を作成する */
    row = n - 1; /* 最初の現在行の設定 */
    col =  (3); /* 最初の現在列の設定 */
    val = 1; /* マスに格納する数の初期値設定 */
    maho[row][col] = val; /* 現在位置に最初の値を設定 */
    while(val <  (4)) { /* 全ての数を格納するまで繰り返す */
        svRow = row; /* 現在の位置を退避する */
        svCol = col;
        row++; /* 新しい現在位置を計算する */
        col++;
    }
}
```

```

if ( (5) ) { /* 新しい現在位置の行が最下行の */
    row = 0; /* 下であれば最上行に移動する */
}
if ( (6) ) { /* 新しい現在位置の列が最右列の */
    col = 0; /* 右であれば最左列に移動する */
}
if (maho[row][col] != 0) { /* 新しい現在位置に値が */
    row = (7); /* あれば直前に値を格納した */
    col = (8); /* 上の位置に移動する */
}
val++;
maho[row][col] = val; /* 現在位置に値を格納する */
}
/* 完成した魔方陣を表示する */
for(row =0; row < n; row++) {
    for(col = 0; col < n; col++) {
        printf(" %3d", maho[row][col]);
    }
    printf("¥n");
}
}

```

(2) の解答群

ア.  $n \% 2 == 0 \ \&\& \ n \geq 20$                       イ.  $n \% 2 == 0 \ || \ n \geq 20$   
ウ.  $n \% 2 == 1 \ \&\& \ n \geq 20$                       エ.  $n \% 2 == 1 \ || \ n \geq 20$

(3) , (4) の解答群

ア.  $n / 2$                       イ.  $n$                       ウ.  $n + 1$   
エ.  $n * 2$                       オ.  $n * n - 1$                       カ.  $n * n$

(5) , (6) の解答群

ア.  $col > n - 1$                       イ.  $col > n$                       ウ.  $col > n + 1$   
エ.  $row > n - 1$                       オ.  $row > n$                       カ.  $row > n + 1$

(7) , (8) の解答群

ア.  $svCol - 1$                       イ.  $svCol$                       ウ.  $svCol + 1$   
エ.  $svRow - 1$                       オ.  $svRow$                       カ.  $svRow + 1$

次の表計算ソフトの仕様を読み、各設問に答えよ。

この問題で使用する表計算ソフトの仕様は下記のとおりである。

COUNTIF 関数

指定した検索範囲の中で、条件に一致するセルの個数を返す。

書式：COUNTIF(検索範囲, 条件)

DATE 関数

年, 月, 日に該当する日付を表すシリアル値を返す。

書式：DATE(年, 月, 日)

IF 関数

条件が真のときに真の場合, 偽のときに偽の場合の計算結果や値を返す。

書式：IF(条件, 真の場合, 偽の場合)

MATCH 関数

検査範囲から検査値が存在するセルの相対的な位置を返す。位置は1から始まる相対的な値である。検査範囲は1行または1列である。検査の型は, 検査値と等しい最初の値を検索する場合は0, 検査値以下の最大の値を検索する場合は1, 検査値以上の最小の値を検索する場合は-1を指定する。なお。検査範囲のデータは検査値が1の場合は昇順に, 検査値が-1の場合は降順に整列されている必要がある。

書式：MATCH(検査値, 検査範囲, 検査の型)

MID 関数

文字列中の指定した位置から始まる指定した文字数の文字列を返す。位置と文字数は1以上の整数値である。

書式：MID(文字列, 位置, 文字数)

MONTH 関数

日付のシリアル値から月の値を返す。

書式：MONTH(日付のシリアル値)

WEEKDAY 関数

日付を表すシリアル値から曜日を表す数値(1~7: 1が日曜, 2が月曜, ..., 7が土曜)を返す。

書式：WEEKDAY(日付のシリアル値)

式

=に続いて計算式や関数などを入力する。

セル番地の絶対参照

セル番地に\$を付けることで、絶対番地(絶対参照)を表す。

別シートの参照

ワークシート名に「!」を付けてセル位置を指定することにより別シートを参照できる。

例：シート名「集計」のセル A1 を参照する場合は、「集計!A1」と記述する。

この問題で使用するマクロの仕様は下記のとおりである。

[マクロの記述]

マクロは一意のマクロ名を付けて宣言する。ワークシートには複数のマクロを格納できる。なお、マクロではワークシートと同様の関数を式内で利用することができる。

変数の型には、数値型、文字列型及び論理型があり、宣言することで使用できる。また変数名に続けて添字を「[」と「]」で囲むことで配列変数として宣言できる。添字が複数ある場合は「,」で区切る。添字は0から始まる。

[マクロの記述形式]

記述形式	説明
○マクロ：マクロ名	マクロを宣言する。
○変数の型：変数名	変数を宣言する。
・変数 ← 式	変数に式の値を代入する。
/* 文 */	注釈を記述する
▲ 条件式       ・処理 1  _____ ▼   ・処理 2	選択処理 条件式が真の時は処理 1 を実行し、偽の時は処理 2 を実行する。
■ 条件式       ・処理   ■	前判定繰り返し処理 条件式が真の間、処理を実行する
■ 変数：初期値，条件式，増分       ・処理   ■	繰返し処理 開始時点で変数に初期値が格納され、条件式が真の間、処理を繰り返す。また、繰り返すごとに、変数に増分を加える。

[演算子と優先順位]

演算の種類	演算子	優先順位
単項演算	+, -, Not	高 ↑ ↓ 低
乗除演算	*, /, %	
加減演算	+, -	
関係演算	>, <, ≥, ≤, =, ≠	
論理積	And	
論理和	Or	

[セルの参照]

マクロ中からセル番地を指定して、セルの値を参照することができる。セル番地を指定した絶対参照と、基点となるセル番地とそこからの相対的な位置を指定した相対参照がある。相対参照では、セル番地、行方向の相対位置、列方向の相対位置を「,」で区切って順に指定する。

セルの絶対参照と相対参照の例

- wk\_a ← 絶対(A1) /\* A1 番地 \*/
- wk\_b ← 絶対(Sheet1!A1) /\* Sheet1 の A1 番地 \*/
- wk\_c ← 相対(A1, 1, 2) /\* C2 番地 \*/

(注) セル A1 から 1 行, 2 列移動したセル C2 が参照される。

飲食店を経営している J さんは10名のアルバイトを採用しており、1日当たり5名を出勤させるシフトを組んでいる。シフト作りに労力を費やされることを回避したい J さんは、表計算ソフトを利用してシフトを組むことを考えた。

[メンバ表ワークシートについて]

	A	B	C	D	E	F	G	H
1	氏名	日	月	火	水	木	金	土
2	岩下 莉沙		X	X				
3	上村 希	X			X		X	
4	前野 亮							X
5	多部 紗季					X	X	
6	奥寺 勤		X					X
7	五大 広之				X			X
8	大竹 敏和		X	X				
9	八木 宏						X	X
10	井村 恵梨香	X			X			
11	中田 里奈	X						

図1 「メンバ表」ワークシート



アルバイトの氏名と出勤希望に関する情報をまとめたものである。

- ・出勤できる曜日は空欄，出勤できない曜日は“X”を入力する。
- ・各曜日で出勤できるアルバイトの人数は5人以上いるものとする。

[シフト表ワークシートについて]

	A	B	C	D	E	F	G	…	AD	AE	AF	AG
1	2023年											
2	3月											
3			1	2	3	4	5	…	28	29	30	31
4	氏名	出勤日数	水	木	金	土	日	…	火	水	木	金
5	岩下 莉沙	0						…	X			
6	上村 希	0	X		X		X	…		X		X
7	前野 亮	0				X		…				
8	多部 紗季	0		X	X			…			X	X
9	奥寺 勤	0				X		…				
10	五大 広之	0	X			X		…		X		
11	大竹 敏和	0						…	X			
12	八木 宏	0			X	X		…				X
13	井村 恵梨香	0	X				X	…		X		
14	中田 里奈	0					X	…				
15	未設定者数		7	9	7	6	7	…	8	7	9	7
16	出勤者数		0	0	0	0	0	…	0	0	0	0

図2 シフトを割り当てる前の「シフト表」ワークシート

アルバイトのシフトを入力する。

- ・該当する年はセルA1，月はセルA2に入力する。
- ・セルA5～A14の氏名は，「メンバ表」ワークシートから複写した。
- ・セルC3～AG3は，日付を入力するが，29日以降は数式で表示する。
- ・セルC4～AG4は，日付に合わせた曜日を表示する。
- ・セルC5～AG14は，初期状態では出勤できない日に“X”が入力されており，マクロにより出勤日は“○”，調整日は“ー”を入力する。なお，調整日は出勤に含めない。
- ・セルB5～B14は，出勤日数を表示する。
- ・セルC15～AG15は，シフトに割り当てられていない人数を表示する。ただし，調整日は数えない。
- ・セルC16～AG16は，その日にシフトを割り当てた人数を表示する。

	A	B	C	D	E	F	G	...	AD	AE	AF	AG
1	2023	年										
2	3	月										
3			1	2	3	4	5	...	28	29	30	31
4	氏名	出勤日数	水	木	金	土	日	...	火	水	木	金
5	岩下 莉沙	14		○		○	○	...	X	○		○
6	上村 希	16	X	○	X	○	X	...	○	X	○	X
7	前野 亮	16	○		○	X	○	...	○		○	○
8	多部 紗季	14	○	X	X	○		...	—		X	X
9	奥寺 勤	16	○		○	X	○	...	○	○	○	—
10	五大 広之	16	X		○	X	○	...	○	X	○	○
11	大竹 敏和	17	○	○		○		...	X	○	○	
12	八木 宏	14	○	○	X	X	○	...	○	○		X
13	井村 恵梨香	16	X	○	○	○	X	...		X		○
14	中田 里奈	16			○		X	...		○		○
15	未設定者数		2	4	2	1	2	...	2	2	4	1
16	出勤者数		5	5	5	5	5	...	5	5	5	5

図3 シフトを割り当てた「シフト表」ワークシート

<設問1> 次の「シフト表」ワークシートの作成に関する記述中の  に入れるべき適切な字句を解答群から選べ。

- セル C3～AD3 に日付"1"～"28"を入力した。セル A1 と A2 に入力した年・月により1か月の日数が異なるので、次の式をセル AE3 に入力し、セル AF3 と AG3 に複写した。

=IF(AD3="", "", IF( (1) =\$A2, AD3+1, ""))

- セル C4～AG4 は、日付に対する曜日を表示する。次の式をセル C4 に入力し、セル D4～AG4 に複写した。

=IF(C3="", "", MID("日月火水木金土",  (2), 1))

- セル B5～B14 は、出勤日数を表示する。次の式をセル B5 に入力し、セル B6～B14 に複写した。

= (3)

- セル C15～AG15 は、その日のシフトに割り当てられていない人数を表示する。次の式をセル C15 に入力し、セル D15～AG15 に複写する。

=IF(C3="", "",  (4))

- セル C16～AG16 は、その日の出勤者数を表示する。次の式をセル C16 に入力し、セル D16～AG16 に複写する。

=IF(C3="", "",  (5))

(1) の解答群

- ア. MONTH (DATE (\$A1, \$A2, AD3))
- ウ. MONTH (DATE (A\$1, A\$2, AD3))

- イ. MONTH (DATE (\$A1, \$A2, AD3 + 1))
- エ. MONTH (DATE (A\$1, A\$2, AD3 + 1))

(2) の解答群

- ア. WEEKDAY (DATE (\$A1, \$A2, C3))
- ウ. WEEKDAY (DATE (A\$1, A\$2, C3))

- イ. WEEKDAY (DATE (\$A1, \$A2, C3 + 1))
- エ. WEEKDAY (DATE (A\$1, A\$2, C3 + 1))

(3) ~ (5) の解答群

- ア. COUNTIF (C5:C14, "")
- ウ. COUNTIF (C5:C14, "○")
- オ. COUNTIF (C5:AG5, "X")

- イ. COUNTIF (C5:C14, "X")
- エ. COUNTIF (C5:AG5, "")
- カ. COUNTIF (C5:AG5, "○")

<設問 2> シフトの割り当て作成に関する記述中の  に入れるべき適切な字句を解答群から選べ。

アルバイトのシフトを割り当てるため、「シフト表」ワークシートでは、初期表示を行うマクロ `init` と出勤日を割り当てるマクロ `make` を実行する。マクロ `make` を何回か繰り返した後、人の手によって最終調整を行う。

[マクロ `init` の説明]

「メンバ表」ワークシートに入力した曜日の出勤に関する情報を「シフト表」ワークシートに展開する。

1. セル C5~AG14 をクリアする。
2. セル C5~AG14 に、「メンバ表」ワークシートの各曜日の値を格納する。

なお、ここでは次の関数を使用する。

<p><code>clear</code> 関数</p> <p>セル範囲で指定したセルの内容をクリアする。</p> <p>書式：<code>clear(セル範囲)</code></p>
--

[マクロ init]

○マクロ : init

○数値型 : row, col, wr

○文字列型 : week

```
・ clear(C5:AG14)          /* シフト割り当て領域をクリア */
■ row: 0, row ≤ 9, 1      /* メンバ 1 人ずつの処理 */
  ・ col ← 0
    ■ 相対(C3, 0, col) ≠ ""      /* 日付が無い位置まで繰り返す */
      ・ week ← 相対(C4, 0, col)  /* 曜日の文字を取得 */
      ・ wr ← (6)
      ・ 相対(C5, row, col) ← 相対(メンバ表!B2, row, wr)
      ・ col ← col + 1          /* 次の日付のセルへ移動 */
    ■
  ■
```

### (6) の解答群

- ア. match(week, メンバ表!A2:A11, 0)
- イ. match(week, メンバ表!A2:A11, 0) + 1
- ウ. match(week, メンバ表!B1:H1, 0) - 1
- エ. match(week, メンバ表!B1:H1, 0)

[マクロ make の説明]

次の手順に従って、「シフト表」ワークシートのシフトに割り当てられていないセルに“○”と“-”を埋める。ただし、連続して出勤するのは三日までとする。

<手順>

1. 以下の a ~ d の処理を繰り返す。
  - a. 出勤者数が 5 人未満の日の中で、シフトに割り当てられていない未設定者数の最小値を求める。
  - b. a で求めた人数である列の中から、出勤者数が最も少ない列を探す。複数ある場合は、日の小さい方を優先する。なお、該当する列が存在しなければ割り当ての繰り返しを終える。
  - c. b で求めた列の中で、シフトに割り当てられていない人の中で出勤日数が最小である人の行を探す。複数ある場合は、上位の行を優先する。
  - d. c で求めた行と b で求めた列の位置に“○”を格納する。
2. 1 の終了後、四日以上連続出勤になる場所を探し、四日目の該当する位置に“-”を格納する。

なお、ここでは次の関数を使用する。関数で利用する行と列は、セル C5 を基点とした相対位置で表す。

### minCol 関数

列 C~列 AG において、16 行の出勤者数が 5 人未満で 15 行の未設定者数が最小となる列の中から、16 行の出勤者数が最小の列位置を返す。該当する列が存在しない場合は -1 を返す。

書式 : minCol()

### minRow 関数

引数の列の中で、シフトに割り当てられていない、B 列の出勤日数が最小の行位置を返す。

書式 : minRow(列)

### [マクロ make]

○マクロ : make

○整数型 : col, row, cnt, k

/\* 手順の 1 に従って出勤日を割り当てる \*/

• col ← minCol()

■ (7)

• row ← minRow(col)

• 相対(C5, row, col) ← "○"

• col ← minCol()

/\* 手順の 2 四日以上連続出勤になっている場合の調整処理 \*/

■ row: 0, row ≤ 9, 1

/\* メンバ全員を調べる \*/

• col ← 3

/\* 4 日の日付から調べる \*/

■ 相対(C3, 0, col) ≠ ""

/\* 日付がある間調べる \*/

• cnt ← 0

/\* 出勤日数を四日分求める \*/

■ k: 0, k ≤ 3, 1

▲ 相対(C5, row, col - k) = "○"

• cnt ← cnt + 1

▲ cnt ≥ 4

/\* 四日以上なら調整日にする \*/

• 相対(C5, row, col) ← "-"

col ← col + 1

/\* 次の日付に進む \*/

(7) の解答群

ア.  $col = 0$

ウ.  $col > 0$

イ.  $col \neq 0$

エ.  $col \geq 0$

## 選択問題 アセンブラの問題

次のアセンブラ言語CASL II プログラムの説明を読み、各設問に答えよ。

[プログラムの説明]

要素数  $N$  の 1 次元配列 TBL を、挿入法により降順に整列する副プログラム SSORT である。

配列 TBL は、TBL+0 番地以降に連続して格納済みであり、 $N$  は  $N$  番地に格納済みである。ただし、 $N$  は、2 以上 100 以下とする。

SSORT は、図 1 のような形式でパラメタが格納された先頭番地を GR1 に設定して呼び出される。

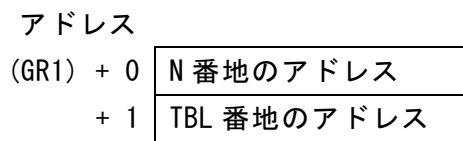


図 1 パラメタの構造

[挿入法の手順]

1 次元配列 TBL が、TBL+0~TBL+k-1 番地まで降順に整列されているとき、TBL+k 番地の内容を格納すべき位置を見つけて挿入する手順は、次のようになる。なお、 $k$  の値は 1~ $N-1$  とする。

- ① TBL+k 番地の内容を WRK に退避する。
- ②  $m$  を  $k-1$  とする。
- ③ TBL+m 番地の内容が WRK 以上であれば、格納すべき位置を見つけたことになるので④に進む。そうでなければ、TBL+m 番地の内容を TBL+m+1 番地へ格納して  $m$  から 1 を引き、③へ戻る。
- ④ WRK を TBL+m+1 番地へ格納する。

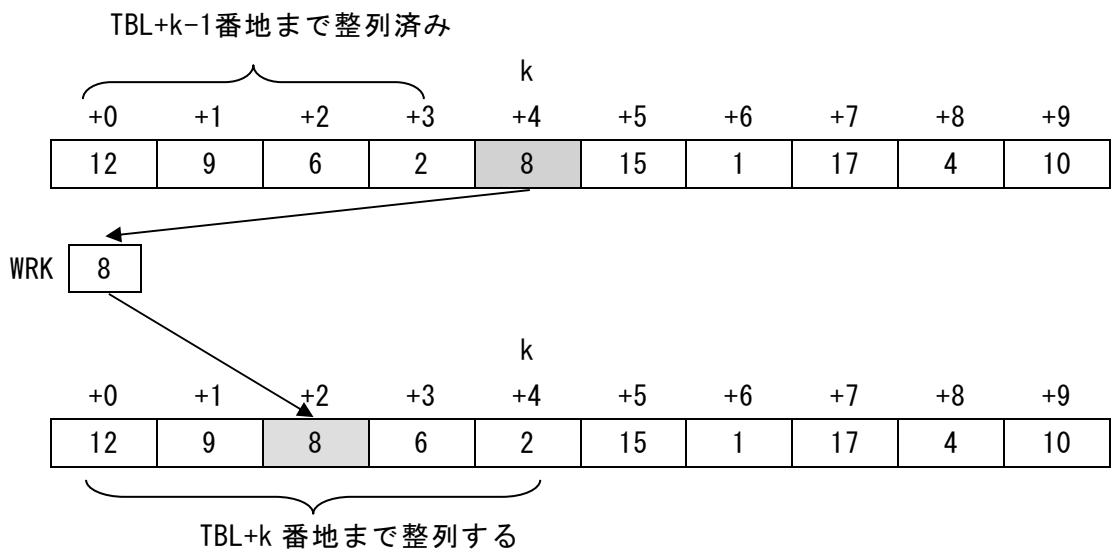


図 2 挿入法の例 ( $N=10$ )

[プログラム]

行番号	ラベル	命令	オペランド	コメント
100	SSORT	START		
110		RPUSH		
120		LD	GR3,0,GR1	;N番地のアドレスをGR3へ
130		LD	GR2,1,GR1	;TBL番地のアドレスをGR2へ
140		ST	GR2, TOP	;データの先頭番地をTOPへ
150		LD	GR4,0,GR3	
160		ADDL	GR4,GR2	
170			(1)	
180		ST	GR4, LAST	;データの末尾番地をLASTへ
190	LOOP1	ADDL	GR2,=1	
200			(2)	;挿入データをWRK (GR1)へ
210		LD	GR3,GR2	
220	LOOP2	SUBL	GR3,=1	
230		CPL	GR3, TOP	;先頭まで比較したか?
240		JMI	FIND	
250		CPA	GR1,0,GR3	
260			(3)	;挿入位置が確定した
270		JZE	FIND	;挿入位置が確定した
280		LD	GR4,0,GR3	
290			(4)	;データを右隣りへ
300		JUMP	LOOP2	
310	FIND	ST	GR1,1,GR3	
320			(5)	;全データを処理したか?
330		JMI	LOOP1	
340		RPOP		
350		RET		
360	TOP	DS	1	
370	LAST	DS	1	
380		END		



<設問 1> プログラム中の  に入れるべき適切な字句を解答群から選べ。

(1) の解答群

ア. ADDL GR4,=1  
ウ. SUBL GR4,=1

イ. ADDL GR4, TOP  
エ. SUBL GR4, TOP

(2) の解答群

ア. LD GR1, GR2  
ウ. LD GR1, 0, GR2

イ. LD GR1, GR3  
エ. LD GR1, 0, GR3

(3) の解答群

ア. JMI FIND  
ウ. JPL FIND

イ. JMI LOOP1  
エ. JPL LOOP1

(4) の解答群

ア. ST GR4, 0, GR2  
ウ. ST GR4, 0, GR3

イ. ST GR4, 1, GR2  
エ. ST GR4, 1, GR3

(5) の解答群

ア. CPL GR1, LAST  
ウ. CPL GR3, LAST

イ. CPL GR2, LAST  
エ. CPL GR4, LAST

<設問 2> プログラム中、行番号 190 の命令を別の命令に置き換えてもプログラムに影響しない命令を解答群から選べ。ただし、ラベル(L00P1)はそのままとする。

(6) の解答群

ア. LAD GR2,=1  
ウ. OR GR2,=1

イ. LAD GR2, 1, GR2  
エ. XOR GR2,=1

<設問 3 > 次のトレースに関する記述を読み、記述中の  に入れるべき適切な字句を解答群から選べ。

N=10 で、TBL+0 から TBL+9 番地のデータが図 3 の状態でプログラムを実行すると、挿入データ (TBL+1 番地から TBL+9 番地) のうち、データ領域の先頭 (TBL+0 番地) に挿入される場合が  (7) 回ある。

+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
6	12	9	2	8	15	1	17	4	10

図 3 TBL+0 から TBL+9 番地の内容

(7)の解答群

ア. 1

イ. 2

ウ. 3

エ. 4

オ. 5

<メモ欄>

