

# 令和7年度前期 情報検定

<実施 令和7年9月14日（日）>

## 基本スキル

（説明時間 13：00～13：10）

（試験時間 13：10～14：10）

- ・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、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12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。
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 1 次のプロジェクト管理に関する記述を読み、各設問に答えよ。

プロジェクトの作業の関連性や日程管理は、PERT (Program Evaluation and Review Technique) という手法を用いる。PERT を利用することで、プロジェクトが効率よく実行できるように作業順序や時間配分を計画し、プロジェクトの進行状況を管理できる。

[PERT の手順]

- ① プロジェクトにおける作業の所要日数と、その作業を実施する前に終了しておかなければならない先行作業を作業表にまとめる。

表 1 作業表

作業名	所要日数	先行作業
A	5	なし
B	3	なし
C	4	A
D	5	B
E	3	C, D
F	4	E
G	2	F
H	3	E
I	2	G
J	3	H

- ② 作業表をもとに PERT 図を作成する。PERT 図は、アローダイアグラムで関連を図示したもので、矢線は作業を、丸印は結合点 (ノード) を表す。結合点に到達する作業がすべて終了するまで、結合点から始まる作業は開始できない。
- ③ PERT 図の最初の結合点の最早結合点時刻を 0 とし、各結合点を進みながら所要日数を加算し、最早結合点時刻を求める。また、最終結合点の最早結合点時刻を最遅結合点時刻として記入する。各結合点を戻りながら所要日数を減算し、最遅結合点時刻を求める。

最早結合点時刻とは、結合点から始まる作業を最も早く開始できる時刻である。複数の作業が到達する場合は最も大きい値が選択される。なお、最終結合点の最早結合点時刻が、このプロジェクトの総所要日数である。

最遅結合点時刻とは、プロジェクトの総所要日数に影響を与えずに、この結合点から始まる作業を最も遅く開始できる時刻である。複数の作業が開始される場合は最も小さい値が選択される。

ダミー作業とは、先行作業を表すためのもので、作図上では必要となる所要日数ゼロの作業となる。

なお、図中の網掛け部分は設問の都合上表示していない。

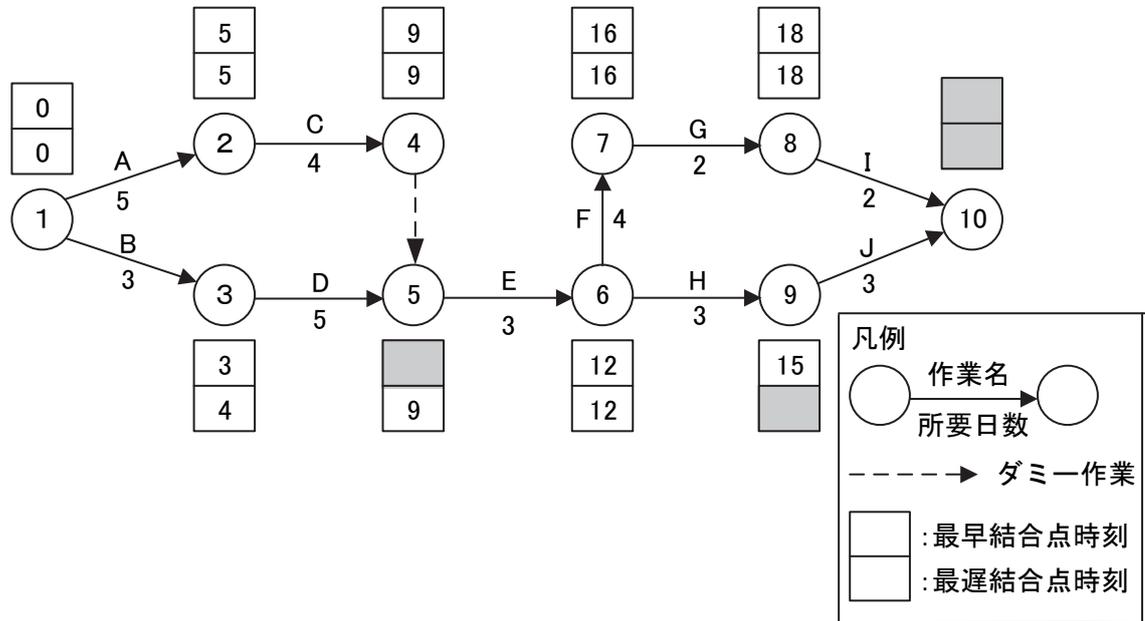


図 PERT 図

<設問 1> 次の PERT 図に関する記述中の [ ] に入れるべき適切な字句を解答群から選べ。

結合点⑤の最早結合点時刻は [ (1) ] であり、結合点⑩の最早結合点時刻(総所要日数)は [ (2) ] である。また、結合点⑨の最遅結合点時刻は [ (3) ] である。

(1) ~ (3) の解答群

- ア. 8                      イ. 9                      ウ. 10                      エ. 15  
 オ. 16                      カ. 17                      キ. 19                      ク. 20

<設問 2> 次の作業の余裕日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。

作業の余裕日数とは、その作業が遅れたとしても総所要日数に影響を与えない日数である。

この余裕日数が 0 の作業を結んだ経路を [ (4) ] と呼び、図では [ (5) ] である。

(4) の解答群

- ア. アーンドバリュー  
ウ. ステークホルダ

- イ. クリティカルパス  
エ. ブランドバリュー

(5) の解答群

- ア. A → C → E → F → G → I  
ウ. B → D → E → F → G → I

- イ. A → C → E → H → J  
エ. B → D → E → H → J

<設問 3> 次の所要日数短縮に関する記述中の  に入れるべき適切な字句を解答群から選べ。

所要日数を短縮するために、費用(コスト)をかけて短縮する場合がある。プロジェクトの PERT 図中の各作業について、各作業を 1 日短縮するのにかかる費用を表 2 に示す。最小費用で総所要日数を 1 日短縮するには、作業  (6) を 1 日短縮すればよい。また、このときにかかる費用は  (7) 万円である。

表 2 1 日短縮の費用

作業名	費用(万円)
A	3
B	3
C	1
D	4
E	2
F	4
G	2
H	1
I	3
J	4

(6) の解答群

- ア. A                      イ. B                      ウ. C                      エ. G                      オ. H

(7) の解答群

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

問題2 次のコード設計に関する記述を読み、各設問に答えよ。

コンピュータで情報を管理する時は、それぞれの情報を識別するためのコードを利用する。例えば、販売管理では伝票コードや商品コード、社員管理では社員コードなどである。代表的なコード体系には次のようなものがある。

表 コードの種類

コード	説明
区分コード	何らかの意図に基づいてグループ分けを行い、グループごとに割り当てられた範囲を利用したもの。
桁別コード	コード中の特定の位置に意味を持たせるもの。
順番コード	1から始まる連続する値を割り当てるもの。
表意コード	その値から対象となる名称が連想できるような文字列を英数字や記号を組み合わせて表現したもの。

<設問1> 次のコードに該当する設定例を解答群から選べ。

- (1) 区分コード
- (2) 桁別コード
- (3) 順番コード
- (4) 表意コード

(1) ~ (4) の解答群

- ア. アクセスログの登録番号として1から始まる連続する整数値を使う。
- イ. 学籍番号として、最初の2桁は入学した年(西暦)の下2桁、学部コード1桁、氏名を五十音順に並べて1から始まる連続する整数値を4桁表現した個人番号、これらを合わせて7桁として作成する。
- ウ. 基本スキルの受験生に対する受験番号は10000~19999、プログラミングスキルの受験生に対する受験番号は20000~29999を割り当てる。
- エ. 日本産の場合はJPN、アメリカ産の場合はUSAのように、英字3文字を使って生産国を表わす。

<設問2> 次のコードの特徴に関する記述に最もあてはまるコードを解答群から選べ。

- (5) コードはデータが発生する順に付けられるが、分類情報を含んでいないためコードだけではどのような意味を持つものかわからない。
- (6) 内容を推測できるように設定されたコードである。例えば和暦の場合は H が平成、R が令和などを意味する。
- (7) 複数の分類を1つのコードで表現できるが、分類を持たせすぎると桁数が大きくなりやすい。

**(5) ~ (7) の解答群**

- ア. 桁別コード
- イ. 区分コード
- ウ. 順番コード
- エ. 表意コード

<設問3> 次のコード設計に関する記述中の  に入れるべき適切な字句を解答群から選べ。

ある著名人が会員制の有料版 SNS を開設することにした。この著名人は誰でも閲覧できる SNS を開設しており、フォロワーは 10 万人いる。有料版に移行した時に会員登録するフォロワー数は 10% 前後になると事前アンケートから予想している。

有料版 SNS に会員登録した人には、次に示す条件で会員番号を付与する。

**[条件]**

- ・会員コードは入会した日付情報、会員区分、連番を組み合わせる。
- ・入会した日付情報は西暦の年のみを使用する。
- ・会員区分には英大文字を使う。
- ・連番は会員登録した年ごとに 1 から始まる連続した整数値を使う。
- ・2025 年から 30 年間使うことを想定する。
- ・会員区分には様々な種類を用意して入会しやすいような料金設定を考える。最初は 5 区分を用意し、10 区分まで増やすことを考えている。

この条件から最低限必要な桁数をそれぞれ考えると、入会した日付情報は  (8) 桁、会員区分は  (9) 桁、連番は  (10) 桁となる。

**(8) ~ (10) の解答群**

- ア. 1
- イ. 2
- ウ. 3
- エ. 4
- オ. 5
- カ. 6
- キ. 7
- ク. 8

問題3 次のリスト構造に関する記述を読み、各設問に答えよ。

リスト構造とはデータ要素を並べたもので、そのデータ要素間の前後関係が物理的な並び順ではなく、ポインタによって論理的に示されるデータ構造である。リスト構造はその性質によって何種類かに分類される。

<設問1> 次の単方向リストに関する記述中の  に入れるべき適切な字句を解答群から選べ。

単方向リストは、各データに次のデータの所在を示すポインタが付けられている。そのため、データ要素の並びを先頭から末尾に向かってたどることはできるが、逆方向はできない。なお、データ要素への参照は、root からだけ行うことができる。また、リストの末尾のポインタは null に設定する。

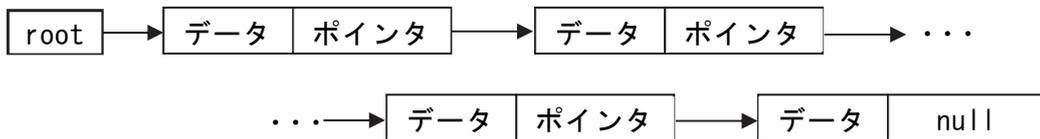


図1 単方向リストの構造

図2は、単方向リストをメモリに展開した図である。ここでは、root の値が「300」なので、リストの先頭は300番地から始まる。リストの先頭からたどると「18, 33, 45, 67, 72」の順にデータを昇順に参照できる。

root		データ	ポインタ
300	100番地	45	500
	200番地	72	null
	300番地	18	400
	400番地	33	100
	500番地	67	200

図2 メモリに展開した単方向リスト

データを昇順にたどれるようにリストへデータを追加するには、ポインタの値を入れ替えばよい。

ここで、図2の状態にデータ「50」を600番地に追加する。この場合、600番地のポ

インタに  を設定した後,  番地のポインタに 600 を設定する。さらに, データ「81」を 700 番地に追加したい場合, 700 番地のポインタに  を設定した後,  番地のポインタに 700 を設定する。

(1) ~ (4) の解答群

- |        |         |         |
|--------|---------|---------|
| ア. 100 | イ. 200  | ウ. 300  |
| エ. 400 | オ. 500  | カ. 600  |
| キ. 700 | ク. null | ケ. root |

<設問 2> 次の双方向リストに関する記述中の  に入れるべき適切な字句を解答群から選べ。

双方向リストは, 後続データを示すポインタ(次ポインタ)と先行データを示すポインタ(前ポインタ)が付けられているため, データ要素への参照は先頭(root)からも末尾(tail)からもたどることができる。



図 3 双方向リストの構造

図 4 は, 双方向リストをメモリに展開した図である。ここでは, root の値が「200」なので, リストの先頭は 200 番地から始まる。リストの先頭からたどると「15, 38, 47, 53」のようにデータの昇順に参照できる。また, tail の値が「300」なので, リストの末尾からたどると「53, 47, 38, 15」のようにデータの降順に参照できる。

root		前ポインタ	データ	次ポインタ
<input type="text" value="200"/>	100 番地	<input type="text" value="400"/>	<input type="text" value="47"/>	<input type="text" value="300"/>
	200 番地	<input type="text" value="null"/>	<input type="text" value="15"/>	<input type="text" value="400"/>
	300 番地	<input type="text" value="100"/>	<input type="text" value="53"/>	<input type="text" value="null"/>
tail				
<input type="text" value="300"/>	400 番地	<input type="text" value="200"/>	<input type="text" value="38"/>	<input type="text" value="100"/>

図 4 メモリに展開した双方向リスト

双方向リストでもデータを昇順および降順にたどれるように追加するには、ポインタの値を入れ替えばよい。

ここで、図4の状態にデータ「50」を500番地に追加する。この場合、500番地の次ポインタに  を、前ポインタに  を設定した後、100番地の次ポインタと300番地の前ポインタに  を設定する。さらに、データ「6」を600番地に追加した場合、600番地の次ポインタに  を、前ポインタに  を設定した後、 に600を設定する。また、200番地の前ポインタに600を設定する。

(5) ~ (10) の解答群

- |         |         |         |
|---------|---------|---------|
| ア. 100  | イ. 200  | ウ. 300  |
| エ. 400  | オ. 500  | カ. 600  |
| キ. null | ク. root | ケ. tail |



<設問2> 次のアドレス指定方式に関する記述中の  に入れるべき適切な字句を解答群から選べ。

各レジスタ、主記憶装置の条件と内容が図のような時の実効アドレスを考える。

[条件]

命令レジスタのアドレス部：100

命令アドレスレジスタ：10

ベースレジスタ：30

番地	100	110	120	130	140	150	160
データ	120	140	110	100	160	130	150

図 主記憶装置

ここで、ベース（基底）アドレス指定方式の場合、実効アドレスは  (7) となる。また、命令アドレスレジスタを基準とした相対アドレス指定方式の場合、実効アドレスは  (8) となる。

(7) , (8)の解答群

ア. 100

イ. 110

ウ. 120

エ. 130

問題5 次のタスク管理に関する記述を読み、各設問に答えよ。

オペレーティングシステム(OS)から見た仕事の単位をタスクという。OSは複数のタスクに順にCPUを割り当てて、仕事を実行させる。

<設問1> 次のタスクの状態に関する表中の[ ]に入れるべき適切な字句を解答群から選べ。

OSは、CPUを時分割に割り当てながら複数のタスクを同時並行的に実行させるマルチタスク機能を備えている。マルチタスク環境で複数のタスクの同時並行動作を実現するために、OSはタスクの生成から消滅までを三つの状態で管理している。

表 タスクの状態

状態名称	説明
[ (1) ]	タスクにCPUが割り当てられて、CPUを使って命令を実行している状態
[ (2) ]	タスクが命令を実行する条件は整っているが、他のタスクがCPUを使用しているので、CPUが割り当てられていない状態
[ (3) ]	タスクが入出力動作中などで、CPUを必要としない状態

(1) ~ (3) の解答群

- ア. 実行可能状態      イ. 実行状態      ウ. スレッド  
 エ. ファイルシステム      オ. フラグメンテーション      カ. 待ち状態

<設問2> 次のタスクの状態遷移に関する記述中の[ ]に入れるべき適切な字句を解答群から選べ。なお、空欄(1)~(3)は設問1と同じ字句が入る。

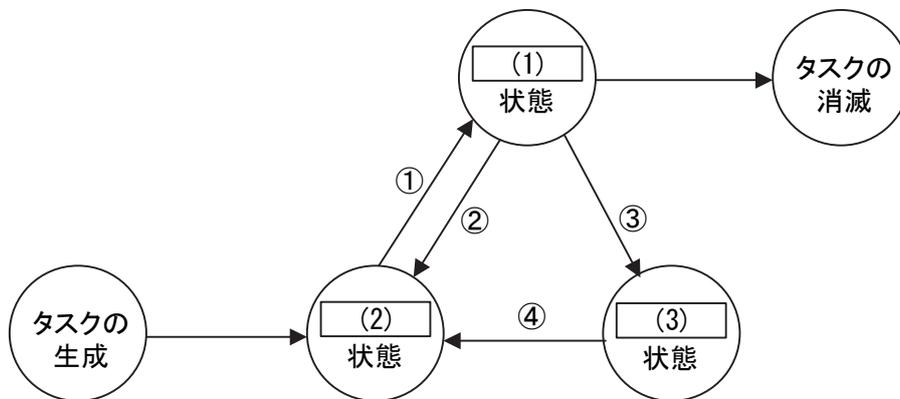


図 タスクの状態遷移

- ① (2) から (1) へ遷移する。  
CPU の割り当てを待っている行列からタスクが選択されて (1) になる。このように CPU の利用時間を割り当てることを (4) と呼ぶ。
- ② (1) から (2) へ遷移する。  
割り当てられた CPU 使用時間が終了すると、タイマ割込みなどによって (2) となり、他のタスクが CPU を利用できるようになる。このように CPU の使用を一定時間とし、全てのタスクにできるだけ公平に CPU を割り当てるタスクスケジューリングを (5) 方式と呼ぶ。
- ③ (1) から (3) へ遷移する。  
入出力装置の利用など、CPU 以外のサービスを要求すると (6) 割込みが発生して (3) になる。
- ④ (3) から (2) へ遷移する。  
入出力装置の利用が完了すると (7) 割込みが発生して (2) になる。

(4) ~ (7) の解答群

- |        |             |           |
|--------|-------------|-----------|
| ア. SVC | イ. タイマ      | ウ. ディスパッチ |
| エ. 入出力 | オ. プリエンプティブ | カ. 優先度順   |
| キ. 横取り | ク. ラウンドロビン  |           |

<メモ欄>

