

# 令和2年度前期 情報検定

<実施 令和2年9月13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（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15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. 試験後にお知らせする合否結果（合否通知）、および合格者に交付する「合格証・認定証」はすべて、Webページ（PC、モバイル）での認証によるデジタル「合否通知」、デジタル「合格証・認定証」に移行しました。
  - ①団体宛にはこれまでと同様に合否結果一覧ほか、試験結果資料一式を送付します。
  - ②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。

問題を読みやすくするために、  
このページは空白にしてあります。

問題1 次のソフトウェア規模の見積りに関する記述を読み、各設問に答えよ。

ソフトウェア開発では、開発期間や必要な要員数について計画段階から正確に把握することが重要である。

<設問1> 次のファンクションポイント法に関する記述中の□□□□に入れるべき適切な字句を解答群から選べ。

ファンクションポイント法とは、ソフトウェア仕様から表1に示す5項目の分類で機能数を洗い出し、機能ごとの複雑度から重み付けを行い、ファンクションポイント(FP値)を算出して、ソフトウェア規模の見積りを行う手法である。

いま、機能ごとのFP値を次式により算出し、その合計をソフトウェアのFP値とする。

機能ごとのFP値＝機能の個数×複雑度の重み

ソフトウェアのFP値＝機能ごとのFP値の合計×補正係数(0.75)

表1に、あるソフトウェアの機能の個数および複雑度の重みを示す。

表1 機能の個数および複雑さの重み

機能	機能の個数	複雑度の重み
外部入力	4	3
外部出力	6	3
外部参照	3	2
内部論理ファイル	3	6
外部インタフェース	4	4

表において、外部出力のFP値は□(1)であり、すべての機能ごとのFP値の合計は□(2)である。したがって、このソフトウェアのFP値は□(3)となる。

(1)～(3)の解答群

ア. 12

イ. 18

ウ. 52.5

エ. 70



問題2 次のデータ圧縮に関する記述を読み、各設問に答えよ。

静止画や動画などのコンテンツは、そのまま記録するとファイルサイズが巨大になるため、データ伝送や保存する場合に効率が悪くなる。そこで、圧縮して記録し再生時に伸張する方式が一般的である。画像のみならず文字データにも適用できる。

<設問1> 次のランレングス符号化に関する記述中の  に入れるべき適切な値を解答群から選べ。

ランレングス符号化は、同じデータ(英字)が連続して出現することに着目して圧縮する方法であり、ここでは3文字以上連続している場合に「データ(英字)+連続した個数」の形でまとめる。ただし、3文字未満の場合はそのままとする。

表1の①～③はランレングス符号化により圧縮したものである。

表1 ランレングス符号化

	圧縮前	圧縮後
①	AAAAABBBBAACCCCCDDDD	A5B3AAC6D4
②	EEEEFFFFGGGHHHHH	<input type="text" value="(1)"/>
③	EFGH	<input type="text" value="(2)"/>

圧縮率を次のように定義する。

$$\text{圧縮率(\%)} = (\text{圧縮後の文字数}) \div (\text{圧縮前の文字数}) \times 100$$

①の圧縮率は  %となる。

また、①と②のように連続するデータが多いと圧縮後のデータ量が減少するが、③のように連続するデータが無いと圧縮が行われなためデータ量は減少しない。

(1), (2) の解答群

- ア. EFGH      イ. E1F1G1H1      ウ. E3F2G3H2      エ. E3F5G3H5

(3) の解答群

- ア. 10      イ. 25      ウ. 50      エ. 75

<設問2> 次のハフマン符号化に関する記述中の  に入れるべき適切な値を解答群から選べ。

文字コードには固定長のビットが割り当てられている。例えば JIS 8 単位符号では 1 文字が 8 ビットである。ハフマン符号は、出現頻度の高い文字には短い符号を、出現頻度の低い文字には長い符号を割り当てる方法である。

ある文書の文字の出現確率を調べたところ、表 2 の結果を得た。

表 2 各文字の出現確率

文字	出現確率 (%)
A	60
B	24
C	12
D	4

出現確率の低い方から文字を二つ選び、それらを葉とする 2 分木を作成し、親となる節には葉の出現確率の合計を記述する(図 1)。なお、出現確率の高い方を左側の葉とする。また、2 分木では子を持つ場合を“節” 子を持たない場合を“葉” と呼び、節や葉を結ぶ線を“枝” と呼ぶ。

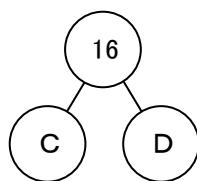


図 1 最初の 2 分木

次に、この節の出現確率と他の文字の出現確率の中から同様に出現確率の低い二つを選び 2 分木を作成する(図 2)。

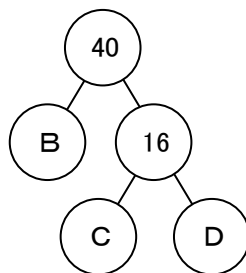


図 2 次の 2 分木

これらの操作を、すべての文字が葉になるまで繰り返し、左の枝に“1”右の枝に“0”を割り当てる(図 3)。

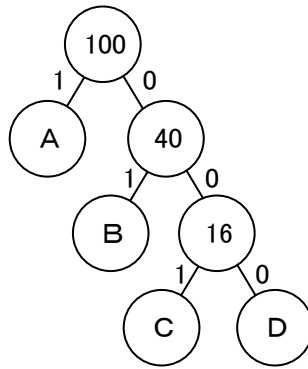


図3 完成した2分木

図3において、文字が格納された葉にたどりつくまでの枝の数字(ビット)を並べたものが、その文字に割り当てられたハフマン符号である。

表3 データの文字とハフマン符号

文字	ハフマン符号
A	<input type="text" value="(4)"/>
B	<input type="text" value="(5)"/>
C	<input type="text" value="(6)"/>
D	<input type="text" value="(7)"/>

(4) ~ (7) の解答群

- ア. 0                      イ. 1                      ウ. 00                      エ. 01  
 オ. 000                      カ. 001



問題を読みやすくするために、  
このページは空白にしてあります。

問題3 次のデータ構造に関する各設問に答えよ。

データがプログラムでどのように扱われるかを定義したものをデータ構造と呼ぶ。データ構造は、プログラムの処理効率に大きな影響を与えるため、様々な構造が考えられている。

<設問1> 次の2分木へのノードの追加に関する記述中の□に入れるべき適切な字句を解答群から選べ。なお、解答は重複して選んでもよい。

2分木とは、1つの親ノードから分岐する子ノードが2つ以下のものを指す。ここでは、ノードの値が必ず「左子ノード<親ノード<右子ノード」となる順序木を扱うものとする。また、ノードの値に同じものは無いものとする。

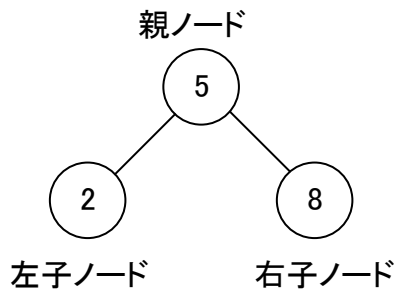


図1 2分木

2分木にノードを追加するには、まず親ノードと追加ノードの値を比較する。追加ノードの値が小さい場合は左子ノードへ、大きい場合は右子ノードへ進む。進んだ先にノードが存在する場合は値の比較を繰り返し、進んだ先にノードが存在しない場合は追加データをその方向の子データとして追加する。なお、ここではノードの値や位置を入れ替えることはしないものとする。

図2の状態にある2分木に対して、順序木になるようにノードを追加する。

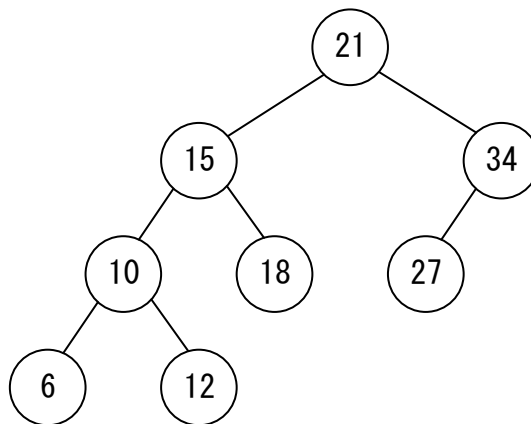


図2 追加前の2分木

「14」を2分木に追加する位置は、の値を持つノードのであり、「33」を2分木に追加する位置は、の値を持つノードのである。

(1) , (3) の解答群

ア. 6                      イ. 12                      ウ. 18                      エ. 27                      オ. 39

(2) , (4) の解答群

ア. 左子ノード                                      イ. 右子ノード

<設問2> 次の2分木からノードを削除する処理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。

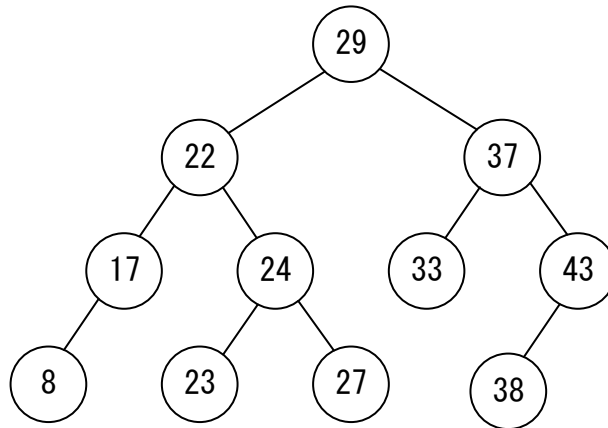


図3 削除前の2分木

子ノードを持つ親ノードを削除する場合、自身以下のノードから新しく親になるノードを選び移動する。図3の状態で「22」の値を持つノードを削除する場合、順序木になるようにするためには左部分木の最大値、または右部分木の最小値を移動しなければならないため、移動する候補はの値を持つノードとなる。また、図3の状態から「37」の値を持つノードを削除する場合は移動する候補はの値を持つノードとなる。

(5) , (6) の解答群

ア. 「8」または「27」                                      イ. 「17」または「23」  
 ウ. 「33」または「43」                                      エ. 「33」または「38」

<設問3> 次の単方向リストへの追加・削除に関する記述中の□に入れるべき適切な字句を解答群から選べ。

リストとは、リスト要素同士をポインタで結ぶ構造である。リストの先頭から順番にデータにアクセスし、後戻りできないものを単方向リストと呼ぶ。単方向リストの先頭の位置は root に格納し、最後の要素のポインタには null を格納する。

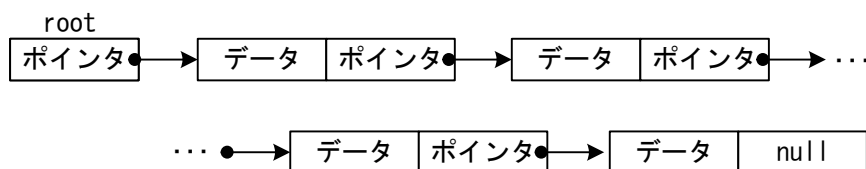


図4 単方向リストの構造

図5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「2000」なので、リストの先頭は2000番地から始まる。リストの先頭からたどると「100, 150, 220, 280, 350」の順番にデータが参照できる。

root		データ	ポインタ
2000	1000番地	150	3000
	2000番地	100	1000
	3000番地	220	4000
	4000番地	280	5000
	5000番地	350	null

図5 メモリに展開した単方向リスト

リストに対してデータを追加や削除する処理は、ポインタの値を入れ替えばよい。ここで、図5の状態にデータ「250」を6000番地に追加し、「220」と「280」の間に参照されるように挿入する。追加前は、リストを参照する順番は「…→3000番地→4000番地→5000番地」となっているので、この間に6000番地を参照するようにポインタを変更する。この場合、3000番地のポインタを□(7)、6000番地のポインタを□(8)に変更する。

また、図5の状態から1000番地に格納されているデータ「150」を削除するには、□(9)番地のポインタを□(10)に変更すればよい。

(7) ~ (10) の解答群

- |         |         |         |
|---------|---------|---------|
| ア. 1000 | イ. 2000 | ウ. 3000 |
| エ. 4000 | オ. 5000 | カ. 6000 |

問題4 次の論理演算に関する各設問に答えよ。

論理演算とは、真と偽の2種類で行う演算である。ここでは、真を1、偽を0とする。主な論理演算を表1に示す。

表1 主な論理演算

論理演算	内容
論理積 (AND)	入力する値が全て1であれば1を、それ以外は0を出力する。演算記号として、「 $\cdot$ 」を使用する。
論理和 (OR)	入力する値に一つでも1があれば1を、それ以外は0を出力する。演算記号として、「 $+$ 」を使用する。
否定 (NOT)	入力された値の否定(1なら0, 0なら1)の値を出力する。演算記号として、「 $\bar{\quad}$ 」を使用する。

また、真理値表は論理変数の取りうる値すべての組合せと、その演算結果を表にしたものである。表2に、論理式  $A \cdot B$  ,  $A+B$  と否定の真理値表を示す。

表2 論理式  $A \cdot B$  ,  $A+B$  と否定の真理値表

入力		出力	
A	B	AND	OR
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

入力	出力
A	NOT
0	1
1	0

論理式は、論理法則などを利用して簡素化できる。主な論理法則を表3に示す。なお、優先順位は()内、否定、論理積、論理和の順である。

表3 主な論理法則

論理法則	簡素化の例
同一の法則	$A+A=A$ , $A \cdot A=A$
恒等の法則	$A \cdot 0=0$ , $A+0=A$ , $A \cdot 1=A$ , $A+1=1$
補元の法則	$A \cdot \bar{A}=0$ , $A+\bar{A}=1$
分配の法則	$A \cdot (B+C)=A \cdot B+A \cdot C$ , $A+B \cdot C=(A+B) \cdot (A+C)$
吸収の法則	$A+A \cdot B=A$ , $A \cdot (A+B)=A$
ド・モルガンの法則	$\overline{A \cdot B}=\bar{A}+\bar{B}$ , $\overline{A+B}=\bar{A} \cdot \bar{B}$

CPU 内部での演算は，論理回路の組み合わせにより実現される。論理回路を表すにはミル記号が用いられる。図 1 にミル記号を示す。


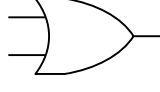
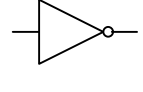
論理回路	AND	OR	NOT
ミル記号			

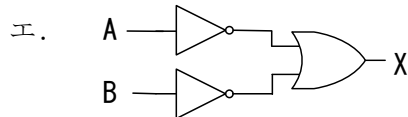
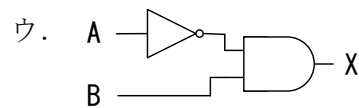
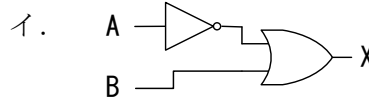
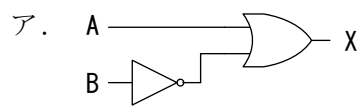
図 1 ミル記号

なお，AND 回路と NOT 回路を組み合わせで否定論理積演算 (NAND) を，OR 回路と NOT 回路を組み合わせで否定論理和演算 (NOR) を構成できる。また，排他的論理和 (XOR) は AND 回路，OR 回路と NOT 回路を組み合わせで実現できる。

<設問 1> 次の論理式と等価な真理値表を出力する論理回路を選べ。

- (1)  $\bar{A} \cdot B + \bar{A} + B$
- (2)  $\bar{A} + (\overline{A \cdot B})$
- (3)  $A \cdot B + \bar{B} + A \cdot \bar{B}$

(1) ~ (3) の解答群



<設問2> 次のビットの加算に関する記述中の  に入れるべき適切な字句を解答群から選べ。

同じ位置にあるビット同士の加算を考える。1ビットの加算は、図2のようになり、真理値表は表4となる。

$$\begin{array}{rcccc}
 A : & 0 & 0 & 1 & 1 \\
 B : & \underline{+ 0} & \underline{+ 1} & \underline{+ 0} & \underline{+ 1} \\
 & 00 & 01 & 01 & 10
 \end{array}$$

図2 1ビットの加算

表4 1ビットの加算結果の真理値表

入力		演算結果	
A	B	同桁の和	桁上がり
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

表4から、同桁の和は  (4) , 桁上がりは  (5) の論理回路で表現できることがわかる。このように、1桁の2進数を二つ加算して同桁の和と桁上がりを出力するものを  (6) と呼ぶ。ただし、下位からの桁上りを考慮していないため、最下位ビットの演算しか行えない。そこで、下位からの桁上りを含めた加算回路が必要である。これは、二つの  (6) と一つの  (7) 回路で構成されている。

(4) , (5) の解答群

- |              |        |         |
|--------------|--------|---------|
| ア. XOR (EOR) | イ. NOT | ウ. NAND |
| エ. NOR       | オ. AND | カ. OR   |

(6) の解答群

- |            |         |         |
|------------|---------|---------|
| ア. アキュムレータ | イ. 全加算器 | ウ. 半加算器 |
|------------|---------|---------|

(7) の解答群

- |         |        |
|---------|--------|
| ア. NAND | イ. NOR |
| ウ. AND  | エ. OR  |

問題5 次のシステムの利用形態に関する記述を読み、各設問に答えよ。

ネットワークを利用したシステムの利用形態には、集中処理システムと分散処理システムがある。

<設問1> 次のシステムの利用形態に関する記述中の項目に該当する処理の組み合わせを解答群から選べ。

表は、集中処理システムと分散処理システムを比較したものであり、他方に対しての特徴を表している。

表 特徴の記述

項目	特徴
A	コンピュータ(処理装置)の停止が全体の停止になる。
B	セキュリティの管理が容易である。
C	運用や保守の管理が複雑である。

(1) の解答群

ア.

A	集中処理
B	集中処理
C	集中処理

イ.

A	集中処理
B	集中処理
C	分散処理

ウ.

A	集中処理
B	分散処理
C	集中処理

エ.

A	分散処理
B	分散処理
C	分散処理

<設問2> 次の分散処理に関する記述中の□□□□に入れるべき適切な字句を解答群から選べ。

分散処理の接続形態は、複数のコンピュータを対等に接続する□□(2)型、処理を依頼するコンピュータと処理を実行するコンピュータに主従関係がある□□(3)型がある。

また、分散処理にはそれぞれ目的がある。処理の中断を最小限にとどめるため、1台のコンピュータが停止したときに他のコンピュータが補う□□(4)分散、処理をFEP(前処理)やBEP(後処理)のように分割し集中させない□□(5)分散、処理内容により利用するコンピュータを分ける□□(6)分散がある。



(2) ~ (6) の解答群

- ア. 機能                      イ. クライアント／サーバ                      ウ. ピアツーピア  
エ. プロキシサーバ        オ. 負荷    カ. リスク

<設問3> 次のネットワークの稼働率に関する記述中の  に入れるべき適切な字句を解答群から選べ。

図1に示すように、福岡と東京を結ぶ稼働率0.9の回線がある。福岡－東京間の回線の信頼性を高めるために、図2に示すように大阪経由の迂回回線を追加した。図2における福岡－東京間の稼働率は  (7) となる。ただし、追加した迂回回線の稼働率は福岡－大阪間、大阪－東京間ともに0.8である。

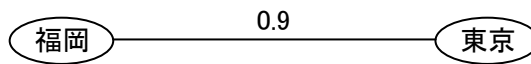


図1 福岡－東京間の回線

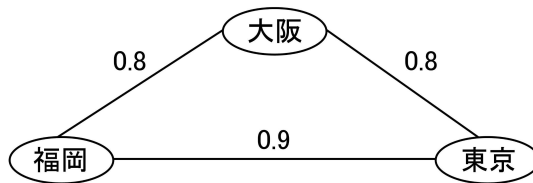


図2 福岡－東京間の回線(迂回回線あり)

(7) の解答群

- ア. 0.900                      イ. 0.964    ウ. 0.999

<メモ欄>

<メモ欄>

