

平成26年度前期 情報検定

＜実施 平成26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（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、電卓付腕時計等
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17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います。

問題 1 次のソフトウェア規模の見積もりに関する記述を読み、各設問に答えよ。

＜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 (FP) 法とは、ソフトウェアの規模を測定する手法の一つである。次の手順により、ソフトウェアの持つ機能の数や複雑度などをもとに、FP 値を算出する。

[FP 値算出手順]

① 測定の対象となるシステムについて、各機能の複雑度は表 1 のとおりである。

表 1 機能および複雑度別係数

機 能	複 雑 度 係 数		
	低	中	高
内部論理ファイル	7	10	15
外部インタフェース	5	7	10
外部入力	3	4	6
外部出力	4	5	7
外部照合	3	4	6

② 各機能の数と、表 1 の複雑度を評価した係数の積和を、未調整 FP 値とする。

未調整 FP 値 = (内部論理ファイルのファンクション数) × (内部論理の複雑度係数)
+ (外部インタフェースのファンクション数) × (外部インタフェースの複雑度係数)
+ (外部入力のファンクション数) × (外部入力の複雑度係数)
+ (外部出力のファンクション数) × (外部出力の複雑度係数)
+ (外部照合のファンクション数) × (外部照合の複雑度係数)

③ システム特性について、その複雑さを評価し、調整値とする。

④ 次式により、FP 値を求める。

FP 値 = 未調整 FP 値 × (0.65 + 調整値 × 0.01)

なお、計算結果の小数点以下は切り上げる。

いま、あるソフトウェアのシステム特性を測定し、表 2 の結果が得られた。

まず、②の計算を考えると、内部論理ファイルについては、ファンクション数が 20 で、複雑度が中であるから複雑度係数は 10 となる。よってその値は、200 となる。同様に外部インタフェースについては 50 となり、②の未調整 FP 値は (1) となる。

また、③の調整値を 30 とすると、④の FP 値は (2) となる。

表2 ソフトウェアの測定結果

機 能	ファンクション数	複雑度
内部論理ファイル	20	中
外部インタフェース	10	低
外部入力	15	高
外部出力	15	低
外部照合	10	中

また、FP 値を使うと、開発工数やコスト、開発期間を見積もることができる。

開発工数[人月] = ソフトウェアの FP 値 ÷ 1 人月で開発できる FP 値

コスト = 開発工数 × 1 人月あたりの費用

開発期間 = 開発工数 ÷ 作業人数

例えば、ソフトウェアの FP 値=800、1 人月で開発できる FP 値=10、1 人月あたりの費用=60 万円、作業人数=10 人とする、開発工数は (3) 人月、コストは (4) 万円、開発期間は (5) ヶ月となる。

(1) , (2) の解答群

ア. 360 イ. 380 ウ. 396 エ. 400
オ. 418 カ. 428 キ. 440 ク. 462

(3) の解答群

ア. 10 イ. 80 ウ. 100 エ. 800

(4) の解答群

ア. 2400 イ. 3600 ウ. 4800 エ. 6000

(5) の解答群

ア. 8 イ. 10 ウ. 16 エ. 80

＜設問２＞ 次のソフトウェア規模の見積もりに関する記述中の□□□□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以外にも、ソフトウェア規模を見積もる手法がある。各手法の特徴を表３に示す。

表３ ソフトウェア規模の見積もり手法

手法	特 徴
(6)	過去に制作した類似の事例から、その実績値をもとに今回の事例を見積もる。精度は担当者の知識や経験に大きく左右される。
(7)	最も古くから存在する手法の一つで、予想されるソースプログラムの行数で見積もる。システム要件がほぼ固まった段階でないと計算できず、プログラマの技量に左右されるため信頼性が低い。
(8)	予想されるソースプログラムの行数をもとに見積もるが、工数と規模の関係は単純な比例関係ではないという考え方に基づいている。プログラマの習熟度などによる補正係数を利用して見積もり、プログラム言語にも左右されず客観性を保てる。

(6) ～ (8) の解答群

ア．COCOMO

イ．LOC(Lines Of Code)法

ウ．RFP

エ．コストプラス法

オ．標準タスク法

カ．類推法

問題を読みやすくするために、
このページは空白にしています。

問題2 次の数値表現に関する記述を読み、各設問に答えよ。

コンピュータで扱う数値には、小数点以下の値を持たない整数型や小数点以下を扱える実数型がある。整数型を扱う場合に使用するのが固定小数点数であり、実数型を扱う場合に使用するのが浮動小数点数である。

＜設問1＞ 次の固定小数点数に関する記述中の に入れるべき適切な値を解答群から選べ。

固定小数点数とは、小数点を決められた場所に固定して表現するものである。最右端ビットの右側に小数点位置があると考えた場合、図1のようになる。

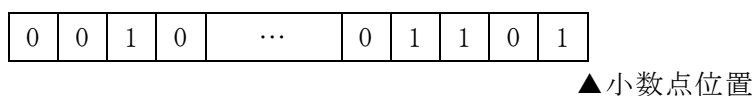


図1 固定小数点数

例えば、8ビットの固定小数点数で0以上の数を扱うとすれば、その最小値は2進数で00000000であり、最大値は11111111である。それぞれ10進数では0と255である。nビットを使って0以上の数を扱う場合、最小値は0、最大値は (1) である。

負数を扱う場合は、先頭ビットを符号ビットとした2の補数表現を使う。

例えば、10進数の-15を2の補数表現により8ビットの2進数に変換するには、次のような手順になる。

- ① -15の絶対値である15を8ビットの2進数に変換する(00001111)。
- ② 各ビットの0と1を反転する(11110000)。
- ③ 1を加える(11110001)。

この方法により8ビットで扱える最小値は2進数で (2) であり、最大値は (3) である。

nビットを使って、先頭ビットを符号ビットとした2の補数で負数を表現する場合の最小値は (4) , 最大値は (5) である。

(1) , (5) の解答群

- ア. $2^{n-1}-1$ イ. 2^n-1 ウ. 2^{n-1} エ. 2^n

(2) , (3) の解答群

- ア. 00000000 イ. 01111111 ウ. 10000000 エ. 11111111

(4) の解答群

- ア. -2^n イ. $-(2^n-1)$ ウ. -2^{n-1} エ. $-(2^{n-1}-1)$

<設問 2> 次の浮動小数点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。

浮動小数点数とは、数値を $(-1)^{\text{符号}} \times \text{仮数} \times \text{基数}^{\text{指数}}$ として表現するものである。
 ここでは 32 ビット（単精度）の IEEE754 形式で説明をする。

符号部 1 ビット	指数部 8 ビット	仮数部 23 ビット
--------------	--------------	---------------

図 2 IEEE754 形式（単精度）

- ・符号部は仮数部の符号を表し、非負の場合は 0、負の場合は 1 とする。
- ・指数部は 2 を基数とし、指数部を必ず正の値にするため、実際の値に 127 を加えたバイアス値とする。
- ・仮数部は絶対値を 2 進法で表し、小数点を左に動かして整数部に 1 だけ残すように桁移動した値の小数部分を保持する。例えば、 $1010.01 = 1.01001 \times 2^3$ のように桁移動した小数部分の 01001 である。このように調整して精度を保つようにすることを (6) と呼ぶ。

例えば、10 進数の 23 は、2 進数で 10111 となり、 1.0111×2^4 と調整し、符号部は 0、指数部は $4+127=131$ (2 進数で 10000011)、仮数部は 1.0111 の小数部である 0.01110...0 (小数部分は全部で 23 ビット) となる。

符号部 : 0
 指数部 : 10000011
 仮数部 : 0111000 00000000 00000000

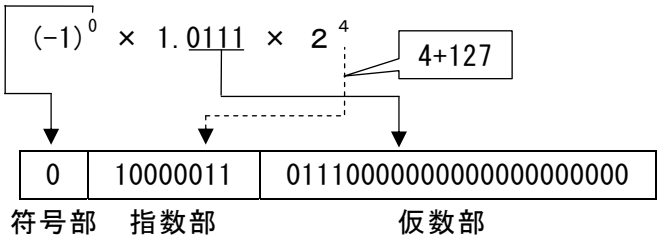


図 3 IEEE754 形式で 10 進数の 23 を表現した結果

同様に、10 進数の 2.5 を IEEE754 形式で表現すると、各部の値は 2 進数表現で次のようになる。

$2.5 = (-1)^0 \times 1.01 \times 2^1$
 符号部 : 0
 指数部 : 10000000
 仮数部 : (7)

なお、浮動小数点数で数値を扱う場合、誤差が生じる場合がある。代表的なものに、情報落ちと桁落ちがある。

情報落ちは した場合に発生し、桁落ちは した場合に発生する。

(6) の解答群

ア．詳略化 イ．細分化 ウ．正規化 エ．標準化

(7) の解答群

ア． 0100000 00000000 00000000 イ． 1000000 00000000 00000000
ウ． 1010000 00000000 00000000 エ． 1100000 00000000 00000000

(8) , (9) の解答群

ア． 仮数部に入りきれない値を格納
イ． 指数部に入りきれない値を格納
ウ． 絶対値の差がほとんどない数値で行う異符号の加算や同符号の減算
エ． 絶対値の差が非常に大きい数値で行う加算や減算

問題を読みやすくするために、
このページは空白にしています。

問題3 次の木構造に関する記述を読み、各設問に答えよ。

木構造とは、一つの要素（節：node）から枝のようにいくつかの子要素を持つデータ構造で、子要素はさらに子要素を持つことができるため、階層的なデータ構造として使われる。親の無い節を根（root）と呼び、子要素を持たない節を葉（leaf）と呼ぶ。なお、ここで木構造に格納する要素の値は0以上の整数とする。

<設問1> 次の2分木に関する記述中の に入るべき適切な字句を解答群から選べ。

一つの節が持つ子要素の数が二つ以下である木構造のことを2分木（binary tree）と呼び、次の条件が常に成立するような構造になっている。

[条件] (左側の子要素の値) ≤ (親の値) ≤ (右側の子要素の値)

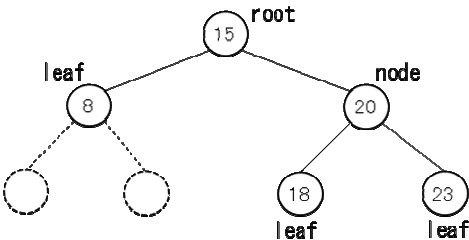


図1 2分木の例

図1の2分木の例を1次元配列BTで表現する。

添字 i	1	2	3	4	5	6	7	8	9	10	11	12	13	...
配列 BT	15	8	X	-1	-1	Y	23	-1	-1	-1	-1	-1	-1	...

図2 1次元配列BT

- ・根は、BT[1]に格納される。
- ・どの要素位置に対しても、添字 i に対して、左側の子要素は BT[2i] に、右側の子要素は BT[2i+1] に格納される。なお、未使用領域には-1を格納する。
- ・配列 BT は、処理に必要な十分な大きさを持っているものとする。

図1の2分木を上記の方法で1次元配列BTを用いて表現すると、Xには (1) が入り、Yには (2) が入る。

また、図2の配列BTに10を追加した場合、配列BT[(3)]に追加され、追加後の葉（leaf）の数は (4) 個となる。

(1) ～ (4) の解答群

- ア. -1 イ. 3 ウ. 4
エ. 5 オ. 18 カ. 20

<設問 2> 次のヒープに関する記述中の に入るべき適切な字句を解答群から選べ。

節の値はその節のどの子よりも小さい（または大きい）2分木をヒープと呼ぶ。なお、ヒープでは、葉は左詰めにし、子要素どうしの大小関係は問わない。ここで扱うヒープは、次の条件が常に成立するような構造になっている。

[条件] (親の値) \leq (全ての子の値)

次のデータをヒープ構造で表現すると図3のようになる。ここで、P, Q, R に入る組み合わせは (5) となる。

[データ] 8, 14, 15, 20, 23

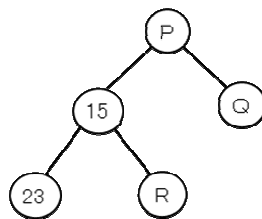


図3 ヒープの例

(5) の解答群

	P	Q	R
ア.	8	14	20
イ.	14	8	20
ウ.	14	20	8
エ.	20	14	8

図4のヒープ構造を持つデータから4を削除した場合、根 (root) の位置に格納されるデータは (6) となる。

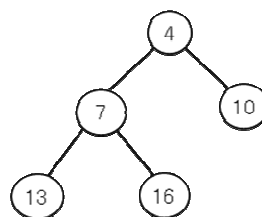


図4

(6) の解答群

ア. 7

イ. 10

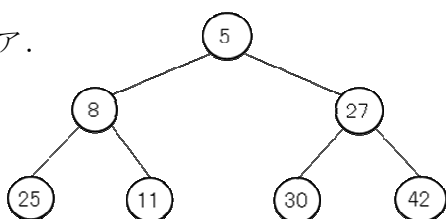
ウ. 13

エ. 16

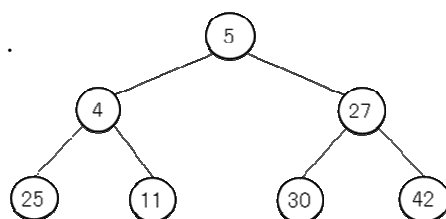
次の 2 分木データで、ヒープの条件を満たしているものは (7) である。

(7) の解答群

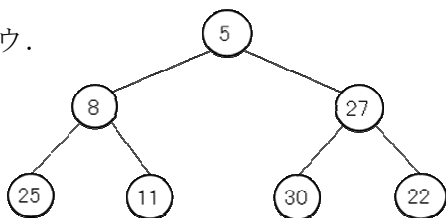
ア.



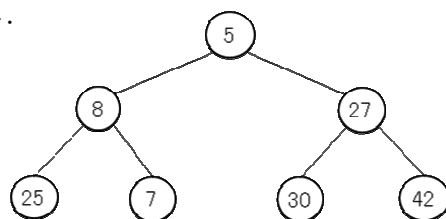
イ.



ウ.



エ.



問題を読みやすくするために、
このページは空白にしています。

問題4 次の補助記憶装置に関する各設問に答えよ。

＜設問1＞ 次の補助記憶装置と媒体の種類に関する記述中の□に入れるべき適切な字句を解答群から選べ。

コンピュータで処理されるプログラムやデータは、あらかじめ主記憶装置に記憶されていなければならない。しかし、主記憶装置は電源を切るとその記憶内容を失ってしまい、記憶できる容量も大きくない。これを補うための装置が補助記憶装置である。

補助記憶装置には、一般的にハードディスクが使われているが、最近では、より軽量で高速なフラッシュメモリをハードディスクの代わりとする□(1)が多くなってきている。

一般的に利用されている補助記憶媒体として光ディスクがある。光ディスクには、標準記憶容量が700Mバイト程度である□(2)、片面一層で4.7Gバイトおよび片面二層で8.5Gバイトの記憶容量を持つ□(3)、片面一層で25Gバイトおよび片面二層で50Gバイトの記憶容量を持つ□(4)などがある。

(1)～(4)の解答群

- | | | |
|-----------------|-----------|------------|
| ア. Blu-ray Disk | イ. CD | ウ. DVD |
| エ. SSD | オ. SD メモリ | カ. USB メモリ |

＜設問2＞ 次のハードディスクのアクセス時間に関する記述中の□に入れるべき適切な字句を解答群から選べ。

ハードディスクからデータをアクセスするには、まず磁気ヘッドが目的のトラックに移動し（シーク）、目的のセクタが回転して磁気ヘッドの下に来るまで待ち（サーチ）、その後読み書き（データ転送）を行う。これらの時間の合計がハードディスクのアクセス時間となるが、シークとサーチはタイミングにより掛かる時間が異なるので、一般には平均値を用いる。

ここで、平均シーク時間20ミリ秒、毎分6,000回転で、1トラック当たりの記憶容量40kバイトのハードディスク装置がある。1ブロック12kバイトのデータを転送する場合、回転速度より平均回転待ち時間は5ミリ秒、データ転送速度は4kバイト/ミリ秒となるため、平均アクセス時間は□(5)ミリ秒になる。

なお、シーク時間はアクセス時間に占める割合が大きい傾向がある。そのため、□(6)などが、平均アクセス時間の短縮に効果的である。

(5)の解答群

- | | | | |
|-------|-------|-------|-------|
| ア. 25 | イ. 28 | ウ. 30 | エ. 33 |
|-------|-------|-------|-------|

(6) の解答群

- ア．データを内側のシリンダと外側のシリンダと交互に記録する
- イ．データをできるだけ別シリンダに分けて記録する
- ウ．データを同一シリンダ内の連続した領域に記録する
- エ．データを小さい単位のブロックに分けて異なるシリンダに記録する

問題5 次のOSのタスク管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。

ユーザから見た仕事の単位をジョブと呼ぶ。

これに対して、OS から見た仕事の単位をタスクと呼ぶ。

サーバ系のコンピュータで動作するデータベースやトランザクション処理のソフトウェアは、一つのジョブが複数のタスクで構成されている。タスクは、CPU を割り当てることで、独自に処理を実行する単位でもある。

現在の 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列的に実行させる [(1)] 機能を備えている。

タスクの中をさらに独立した処理に分割し、処理を実行する単位を [(2)] という。 [(2)] は、OS のオーバーヘッドが非常に少ないので、現在では [(2)] による並列処理が中心になっている。

[(1)] 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、実行可能状態、実行状態、待ち状態の三つの状態で管理している。

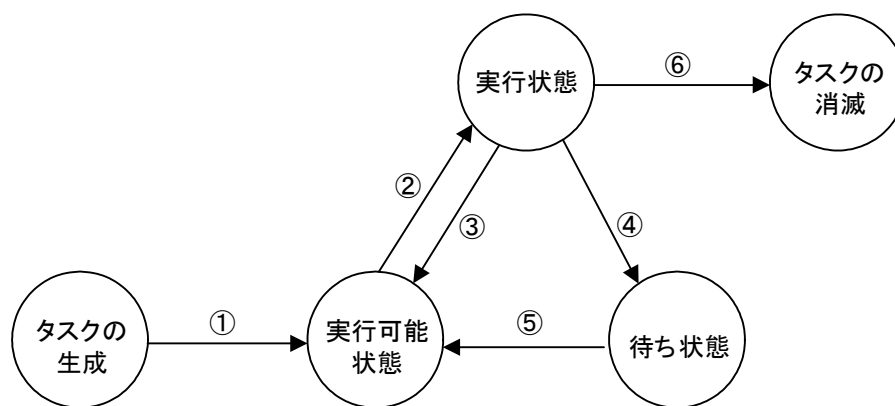


図 タスクの状態遷移

- ① 生成された直後のタスクは、実行可能状態になる。
- ② 実行可能状態のタスクの中から実行するタスクを選択し、そのタスクに CPU の使用权が割り当てられ実行状態となる。この CPU 割り当てを [(3)] と呼ぶ。
- ③ 実行状態中に、例えば、割り当てられた時間が経過したときに発生する [(4)] 割込みによって実行可能状態となり、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)] スケジューリングと呼ぶ。
- ④ 実行状態中に、入出力要求が発生すると、タスクは待ち状態となる。このように入出力など OS の機能を利用するため、スーパーバイザを呼び出すのが [(6)] である。
- ⑤ タスクは入出力終了によって、待ち状態から実行可能状態となる。このとき発生するのが [(7)] 割込みである。
- ⑥ CPU の割り当てを繰り返し、タスクは消滅する。

(1) の解答群

- | | |
|----------------|-----------|
| ア. マルチタスク | イ. マルチブート |
| ウ. マルチプラットフォーム | エ. マルチユーザ |

(2) の解答群

- | | |
|---------|----------|
| ア. イベント | イ. スレッド |
| ウ. セマフォ | エ. プログラム |

(3) の解答群

- | | |
|-------------|------------|
| ア. ディスパッチング | イ. デッドロック |
| ウ. プリエンプション | エ. リロケータブル |

(4) , (6) , (7) の解答群

- | | |
|----------|-----------|
| ア. SVC | イ. 機械チェック |
| ウ. コンソール | エ. タイマ |
| オ. 入出力 | カ. プログラム |

(5) の解答群

- | | |
|-------------|-------------|
| ア. プリエンプティブ | イ. マルチプロセッサ |
| ウ. ラウンドロビン | エ. リエントラント |

