

＼通勤・通学にも！／

科目A 過去問 16回 分^{※1}
^{※2}

スマホ
単語帳^{※1}
付き

令和5年度

基本情報技術者 予想問題集

【2023年】

かんたん合格

株式会社ノマド・ワークス 著

必勝
予想問題
3回分収録
(科目Bも
60問)



合格力UP
の
「カギ」

令和5年度開始

新試験は これでOK!

科目A 全問に出題予測率付き！

インプレス

問題 + 解説 +



合格力に差が付く
3ステップ構成



目 次

⌚ <基本情報技術者> 新試験対応 攻略ガイド	3
⌚ <ゼッタイ出る!> 大定番キーワードde点数UP↑↑	8

基本情報技術者試験

予想問題 ①

17

科目 A	18
科目 B	54

予想問題 ②

109

科目 A	110
科目 B	146

予想問題 ③

207

科目 A	208
科目 B	246
■ 擬似言語の記述形式	306
■ 索引	308

購入者限定特典!!

本書の特典は、下記サイトにアクセスすることでご利用いただけます。

<https://book.impress.co.jp/books/1122101100>

サイトにアクセスの上、画面の指示に従って操作してください。

※特典のご利用には、無料の読者会員システム「CLUB Impress」への登録が必要となります。

※特典のご利用には期限があります。

※本特典を印刷してご利用することはできません。あらかじめご了承ください。

●特典①：本書の電子版

本書の全文の電子版（PDF ファイル）を無料でダウンロードいただけます。

●特典②：（科目 A）過去問 16 回分

新試験の「科目 A 試験」に当たる、旧試験の「午前試験」の過去問 16 回分を無料でダウンロードいただけます（平成 23 年度春期（特別）から平成 30 年度秋期までの問題・解説解答、PDF ファイル）。

※ 新試験制度における「科目 A」とは 1 回分の問題数、試験時間が異なります。詳しくは、次ページからの「新試験対応 基本情報技術者 攻略ガイド」をご確認ください。なお、出題予測率は付いていません。

●特典③：スマホで学べる単語帳「てる語句 200」

基本情報技術者試験で出題頻度の高い 200 の語句がいつでもどこでも暗記できるデジタル単語帳「てる語句 200」を無料でご利用いただけます。

※ 本特典のご利用は、書籍をご購入いただいた方に限ります。

※ 本特典のダウンロード期間・ご利用期間は、いずれも本書発売より 1 年間です。

・ 本書は、基本情報技術者試験の受験対策用の教材です。著者、株式会社インプレスは、本書の使用による基本情報技術者試験への合格を保証するものではありません。

・ 本書の内容については正確な記述につとめましたが、著者、株式会社インプレスは本書の内容に基づくいかなる試験の結果にも一切責任を負いかねますので、あらかじめご了承ください。

・ 本書の試験問題は、独立行政法人 情報処理推進機構（IPA）が公開している情報に基づいて作成しています。

新試験
対応

基本情報技術者 攻略ガイド



基本情報技術者試験は、令和5年4月より、通年でいつでも受験できる方式になります。それにともない、出題形式・出題範囲も大きく変わります。新しくなった試験の概要をくわしく説明します。



基本情報技術者試験の概要

◆実施方式

新しくなった基本情報技術者試験は、「**科目A**」「**科目B**」の2科目で構成されます。各科目的概要は次のとおりです。

試験名	試験時間	問題数	出題形式
科目A	90分	60問（全問必須）	四肢択一式
科目B	100分	20問（全問必須）	多肢択一式

科目A、科目Bは、それぞれ旧試験の午前試験、午後試験に相当しますが、旧試験では合計5時間かかっていた試験時間が、新試験では3時間10分に短縮されます。これにともない、問題数や問題構成も大きく変わります。なお、科目Aと科目Bは、同じ試験会場で1日にまとめて実施されます。

◆CBT方式

試験は、科目A、科目BともにCBT（Computer Based Testing）方式で実施されます。受験者は、予約した日時に試験会場に出向き、パソコン上に表示された試験問題に対して、マウスを操作して解答します。



試験会場には複数の受験者が受験しますが、よーいドンでいっせいに試験をはじめるわけではなく、受付を済ませた順に試験会場に入り、指定された席に座って試験を開始します。試験会場によっては、別の試験を受けている人がいる場合もあります。

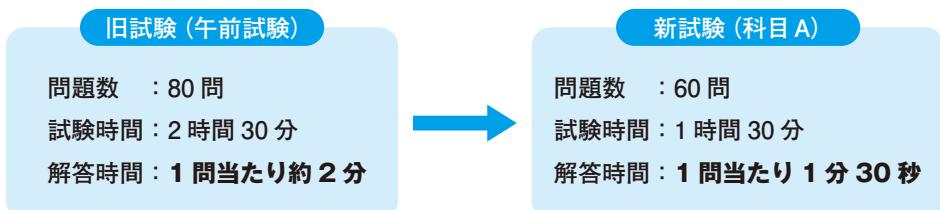
◆科目A試験の概要

科目A試験は、4つの選択肢から1つを選ぶ択一式の問題が、全部で**60問**出題されます。試験時間は**90分**（1時間30分）です。

科目Aの出題範囲

分野	大分類	問題数の内訳（予想）
テクノロジ系	基礎理論 コンピュータシステム 技術要素 開発技術	40～45問
マネジメント系	プロジェクトマネジメント サービスマネジメント	5～10問
ストラテジ系	システム戦略 経営戦略 企業と法務	10～15問

試験問題の形式や出題範囲は旧試験の午前試験に準じます。ただし、新試験では1問当たりの解答時間が短くなっているので注意が必要です。



◆科目B試験の概要

科目B試験は、「**アルゴリズムとプログラミング**」（擬似言語による出題）と「**情報セキュリティ**」の2つの分野で構成されます。

科目Bの出題範囲

分野	カテゴリ	問題数の内訳（予想）
アルゴリズムとプログラミング	①プログラムの基本要素 ②データ構造及びアルゴリズム ③プログラミングの諸分野への適用	16問（カテゴリごとに5～6問）
情報セキュリティ		4問

科目B試験は、出題形式、出題範囲ともに旧午後試験から大きく変更されています。旧試験では1問の問題文が長く、設問も複数ありましたが、新試験では問題文が大幅に短くなっています。設問も1問につき1つです。

選択問題は廃止され、**全20問が必須**になりました（試験時間は100分）。また、個別プログラ

ム言語（C, Java, Python, アセンブラー言語, 表計算ソフト）による出題は、処理の流れを擬似的なプログラム言語で記述した**擬似言語**による出題に統一されます。

◆合格基準

新試験は、科目A、科目Bともに1,000点満点で、それぞれの科目で600点以上得点すれば合格となります。ただし、採点はIRT（項目応答理論）という方式で行われるため、配点は正答率などに応じて変動します。単純に6割正答すれば合格というわけではないことに注意しましょう。

科目A 試験の攻略法

科目A 試験の出題内容は、旧試験の午前試験とほぼ同じなので、午前試験の過去問題の学習が効果的です。本書では、旧試験や応用情報技術者試験の午前試験の過去問題の中から出題が予想される問題を厳選し、新試験に合わせて再構成した予想問題を作成しています。

過去問題の学習による効果は、次のとおりです。

①過去に出題された問題が出題される

旧試験の午前試験は、過去問題が出題される頻度が比較的高いため、多くの過去問題に接しているほど、見覚えのある問題が本番で出題される確率が高くなりました。新試験でも、出題傾向は旧試験と変わらないため、過去問題の学習が効果的です。

②出題される範囲や学習のポイントを把握できる

合格基準は1,000点満点で600点以上なので、満点をとる必要はありません。滅多に出題されないような分野の問題より、ひんぱんに出題される分野に的をしぼって学習するのが効果的です。

③自分の苦手分野がわかる

基本情報技術者試験は、出題範囲をまとめて学習しようとすると非常に広範囲になってしまふため、学習量も膨大になります。過去問題によって、自分に足りている知識・足りていない知識を把握できるので、足りていない知識を重点的に学習できます。

④実践的な学習ができる

すでにひととおりの知識を身に着けている方は、参考書を使わなくても、過去問題を何回か解くだけで大きな学習効果が得られます。予想問題では、過去問題を分析してとくに出題が予想される問題を取り上げています。

なお、本書には付録として旧試験の午前問題16回分のPDFも付いているので、科目Aの試験対策として有効です。



科目B試験の攻略法

科目B試験は新しい試験なので、過去問題を使った学習はできません。そこで本書では公表されているサンプル問題や出題範囲などをもとに作成したオリジナル問題を用意しました。

①擬似言語に慣れよう

科目B試験の8割を占める「アルゴリズムとプログラミング」分野の問題では、情報処理試験の擬似言語によるプログラムが出題されます。本書の予想問題によって、擬似言語の仕様に慣れておいてください。

②データ構造について学習しよう

基礎的なデータ構造とアルゴリズムについて学習しておきましょう。とくに以下のデータ構造が重要です。

配列	二分探索、整列アルゴリズム、ハッシュテーブル
二次元配列	要素の参照
単方向リスト	リストの探索、挿入、削除
スタック	プッシュ、ポップ、スタックを用いたアルゴリズム
キュー	エンキュー、デキュー
二分木	二分木の走査、二分探索木、ヒープ
グラフ	有向グラフ、無向グラフ、隣接行列、トポロジカルソート

③情報セキュリティについても知っておこう

科目B試験では、情報セキュリティ分野の問題が全体の2割（4問程度）出題されます。情報セキュリティについては、科目Aでも出題比率が高いので、まず科目A試験の対策となる基礎知識を学習しましょう。その際、単なる用語の暗記ではなく、それらがどのように活用されているかを具体的な事例などによって把握するようにすれば、おのずと科目Bの試験対策になります。



おすすめ学習法

◆合格のカギ

自分の苦手な分野や、欠けている知識を洗い出して、必要に応じて整理しておきましょう。特に科目Aでは、用語の知識が多く問われます。本書の解説欄の「**合格のカギ**」には問題に関連する用語や問題を解くためのヒントが掲載されているので参考にしてください。

◆間違えた問題をそのままにしない

一度間違えた問題はそのままにせず、チェックボックスにレ印や×印をつけておいて、後で必ず復習しましょう。

◆繰返し問題を解く

今回初めて基本情報技術者試験に挑戦する人は、繰返し問題を解いて実力を養いましょう。すでに解答がわかっている問題でも、見直してみると案外新たな発見があるものです。

◆出題予測率の高い問題を重点的に学習

科目A試験の各問に、過去の情報処理試験の傾向から分析した出題予測率を掲載しています。出題されやすい問題を重点的に学習することで、出題範囲を効率的にカバーできます。

◆時間を計って解く

試験時間は、科目Aが90分・科目Bが100分です。本番で時間が足りなくなっていて、せっかくわかる問題なのに解けなかったということのないよう、少なくとも一度は時間を計って問題を解いてみることをおすすめします。特に、科目Bは時間が足りなくなる人が多いので気をつけましょう。

なお、通勤時間を勉強に充てている人など、まとまった時間がとれない場合は、科目A問題10問を15分とか、科目B問題1問を5分のように、小分けにするのもよいでしょう。

「チャレンジ！ 基本情報技術者」

<https://shikaku.impress.co.jp/fe/>

基本情報技術者の合格をサポート！
解説付き過去問題と頻出用語が学べる



問合せ先

●情報処理技術者試験に関するお問合せ

ホームページ：<https://www.jitec.ipa.go.jp/>

問合せ先：IPA IT人材育成センター国家試験・試験部

「お問い合わせ」に進む前に：https://www.jitec.ipa.go.jp/_jitecinquiry_faq.html

ゼッタイ出る！

大定番キーワード de 点数UP↑↑

科目 A で出題が予想される基本情報技術者試験の定番のキーワードをまとめました。確実に得点できるので、基礎知識として必ずチェックしておきましょう。

※出題予測率は、公表済みの過去問題等から独自に分析しています。

出題予測率
90%

稼働率

システムの運用時間全体のうち、実際にシステムが稼働している時間の割合を**稼働率**といいます。

$$\text{稼働率} = \frac{\text{稼働時間}}{\text{運用時間}}$$

たとえば、システムの運転時間が 100 時間で、そのうち稼働時間が 95 時間の場合、稼働率は $95 \div 100 = 0.95$ です。これは、利用者がこのシステムにアクセスしたとき、95% の確率でシステムを利用できることを意味します。

稼働率は、システムの**可用性**（アベイラビリティ）の指標になります。

● MTBF と MTTR から稼働率を求める

稼働率は、**MTBF**（平均故障間隔）と**MTTR**（平均修理時間）から、次のように求めることができます。

覚える MTBF と MTTR

$$\text{稼働率} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

MTBF (平均故障間隔)	システムが前回故障してから、次に故障するまでの間隔（稼働している時間）の平均。MTBF が長いほど信頼性が高い。
MTTR (平均修理時間)	システムの修理にかかる時間の平均。MTTR が短いほど保守性が高い。

● 複数の装置を接続した場合の稼働率を求める

2 台の装置を直列に接続したシステムでは、一方の装置が故障するとシステム全体が停止てしまいます。この場合のシステム全体の稼働率は、次のように求めます。

テクノロジ系

● 本書掲載
予想 1 問 16 / 予想 2 問 12
予想 3 問 14

覚える 直列システムの稼働率



$$\text{稼働率} = \text{稼働率 A} \times \text{稼働率 B}$$

一方、2 台の装置を並列に接続したシステムでは、1 台の装置が故障しても、もう一方の装置があればシステム全体は運転を継続できます。

システム全体が停止するのは、装置が 2 台とも故障した場合だけです。したがって、システム全体の稼働率は次のように求められます。

覚える 並列システムの稼働率



$$\text{稼働率} = 1 - (1 - \text{稼働率 A}) \times (1 - \text{稼働率 B})$$

2 台とも故障する確率

例題

MTBF が 45 時間で MTTR が 5 時間の装置がある。この装置を二つ直列に接続したシステムの稼働率は幾らか。

ア 0.81 イ 0.90 ワ 0.95 エ 0.99

【解法】装置1台分の稼働率は、MTBFとMTTRから求められます。

$$\frac{45 \text{ 時間}}{45 \text{ 時間} + 5 \text{ 時間}} = \frac{45}{50} = 0.9$$

稼働率0.9の装置2台を直列に接続するので、システム全体の稼働率は

$$0.9 \times 0.9 = 0.81$$

となります。

【解答】ア

出題予測率
90%

論理回路

テクノロジ系

●本書掲載
予想2問18／予想3問18

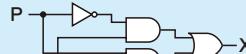
論理回路は、入力データ(0または1)に対して論理演算を行い、結果を0または1で出力する回路です。論理回路は、次のような部品(論理素子)で構成されています。図記号の形と演算との対応は覚えておきましょう。

覚える 論理回路

論理積(AND)	論理和(OR)																				
<table border="1"><thead><tr><th>入力</th><th>出力</th></tr></thead><tbody><tr><td>0 0</td><td>0</td></tr><tr><td>0 1</td><td>0</td></tr><tr><td>1 0</td><td>0</td></tr><tr><td>1 1</td><td>1</td></tr></tbody></table>	入力	出力	0 0	0	0 1	0	1 0	0	1 1	1	<table border="1"><thead><tr><th>入力</th><th>出力</th></tr></thead><tbody><tr><td>0 0</td><td>0</td></tr><tr><td>0 1</td><td>1</td></tr><tr><td>1 0</td><td>1</td></tr><tr><td>1 1</td><td>1</td></tr></tbody></table>	入力	出力	0 0	0	0 1	1	1 0	1	1 1	1
入力	出力																				
0 0	0																				
0 1	0																				
1 0	0																				
1 1	1																				
入力	出力																				
0 0	0																				
0 1	1																				
1 0	1																				
1 1	1																				
排他的論理和(XOR)	論理否定(NOT)																				
<table border="1"><thead><tr><th>入力</th><th>出力</th></tr></thead><tbody><tr><td>0 0</td><td>0</td></tr><tr><td>0 1</td><td>1</td></tr><tr><td>1 0</td><td>1</td></tr><tr><td>1 1</td><td>0</td></tr></tbody></table>	入力	出力	0 0	0	0 1	1	1 0	1	1 1	0	<table border="1"><thead><tr><th>入力</th><th>出力</th></tr></thead><tbody><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></tbody></table>	入力	出力	0	1	1	0				
入力	出力																				
0 0	0																				
0 1	1																				
1 0	1																				
1 1	0																				
入力	出力																				
0	1																				
1	0																				
否定論理積(NAND)	否定論理和(NOR)																				
<table border="1"><thead><tr><th>入力</th><th>出力</th></tr></thead><tbody><tr><td>0 0</td><td>1</td></tr><tr><td>0 1</td><td>1</td></tr><tr><td>1 0</td><td>1</td></tr><tr><td>1 1</td><td>0</td></tr></tbody></table>	入力	出力	0 0	1	0 1	1	1 0	1	1 1	0	<table border="1"><thead><tr><th>入力</th><th>出力</th></tr></thead><tbody><tr><td>0 0</td><td>1</td></tr><tr><td>0 1</td><td>0</td></tr><tr><td>1 0</td><td>0</td></tr><tr><td>1 1</td><td>0</td></tr></tbody></table>	入力	出力	0 0	1	0 1	0	1 0	0	1 1	0
入力	出力																				
0 0	1																				
0 1	1																				
1 0	1																				
1 1	0																				
入力	出力																				
0 0	1																				
0 1	0																				
1 0	0																				
1 1	0																				

例題

図の論理回路と等価な回路はどれか。



- ア P Q → X
イ P Q → X
ウ P Q → X
エ P Q → X

【解法】各素子の演算は、①「Pの否定とQとの論理積」、②「PとQの否定との論理積」、③「①と②の論理和」です。これらの結果を真理値表に書き出すと、次のようになります。

P	Q	\bar{P} AND Q → ①	P AND \bar{Q} → ②	① OR ② → ③
0	0	1 AND 0 → 0	0 AND 1 → 0	0 OR 0 → 0
0	1	1 AND 1 → 1	0 AND 0 → 0	1 OR 0 → 1
1	0	0 AND 0 → 0	1 AND 1 → 1	0 OR 1 → 1
1	1	0 AND 1 → 0	1 AND 0 → 0	0 OR 0 → 0

③の出力は、PとQが同じ値のとき0、異なる値のとき1になるので、**排他的論理和**と等価になります。

【解答】ウ

出題予測率
80%

RAMとROM

テクノロジ系

●本書掲載
予想1問20／予想2問17
予想3問17

電源を切ると記憶内容が消えてしまうメモリを**RAM**(Random Access Memory)といい、代表的なものに**DRAM**(Dynamic RAM)と**SRAM**(Static RAM)があります。

一方、電源を切っても内容が消えないメモリを**ROM**(Read Only Memory)といいます。ROMには多くの種類がありますが、試験では**フラッシュメモリ**がよく出題されます。

覚える 試験に出るメモリ

DRAM

- 構造が単純で大容量化しやすく、ピット当たりの単価が安い。
- コンデンサに蓄えた電荷の有無で情報を記憶するため、定期的にリフレッシュ(記憶内容を再書き込みすること)が必要。
- 主にコンピュータの**主記憶装置**に用いる。

SRAM

- DRAM より高速な読み書きが可能だが、ビット当たりの単価が高い。
- フリップフロップ回路で構成されており、リフレッシュ動作が不要。
- 主にキャッシュメモリとして用いられている。

フラッシュメモリ

- 電源を切った後も内容が消えないメモリ(ROM)の一種。
- 記憶内容を電気的に消去・書換えることができる。
- USB メモリや SD カードなどに用いられている。

例題

SRAM と比較した場合の DRAM の特徴はどれか。

- ア 主にキャッシュメモリとして使用される。
- イ データを保持するためのリフレッシュ又はアクセス動作が不要である。
- ウ メモリセル構成が単純なので、ビット当たりの単価が安くなる。
- エ メモリセルにフリップフロップを用いてデータを保持する。

【解法】ア、イ、エは SRAM の特徴です。

【解答】ウ

出題予測率

80%

LAN間接続装置

テクノロジ系

本書掲載
予想1問26／予想2問25
予想3問25

LAN間接続装置とは、装置と装置、ネットワークとネットワークをつなぐジョイントの役割を果たす装置です。基本情報技術者では、特に次の4種類の機能の違いを把握しておきましょう。下にいくほど機能が高度になることに注意。

覚える LAN間接続装置の種類

リピータ	信号を増幅して、ネットワークの伝送距離を延長する装置。
ブリッジ、スイッチングハブ (レイヤ2スイッチ)	特定の端末に宛てたデータ(=フレーム)を、MAC アドレスと呼ばれる識別番号を基にして中継する装置。接続された端末同士は、1つのネットワーク(セグメント)を構成する。
ルータ	特定の端末に宛てたデータ(=パケット)を、IP アドレスを基にして中継する装置。ネットワークとネットワークをつなぎ、複数のルータを経由してデータを別のネットワーク上にある端末に届けることができる。
ゲートウェイ	プロトコルの異なるネットワーク同士(例:電話回線網とインターネットなど)を接続する装置。

LAN間接続装置に関しては、OSI 基本参照モデルとの対応を覚えておくことも重要です。OSI 基本参照モデルとは、データ通信を7つの層に切り分け、それぞれの層の機能を規定したものです。下から上

の層にいくほど、機能はより抽象的・高度になっていきます。

覚える OSI 基本参照モデルとの対応

OSI 基本参照モデル 対応する LAN間接続装置

第7層 アプリケーション層	ゲートウェイ
第6層 プレゼンテーション層	
第5層 セッション層	
第4層 トランスポート層	
第3層 ネットワーク層	ルータ
第2層 データリンク層	ブリッジ、スイッチングハブ
第1層 物理層	リピータ

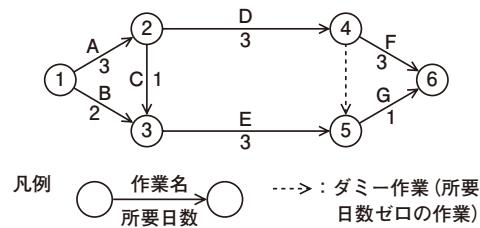
例題

LAN間をOSI基本参照モデルの物理層で相互に接続する装置はどれか。

- ア ゲートウェイ
- イ ブリッジ
- ウ リピータ
- エ ルータ

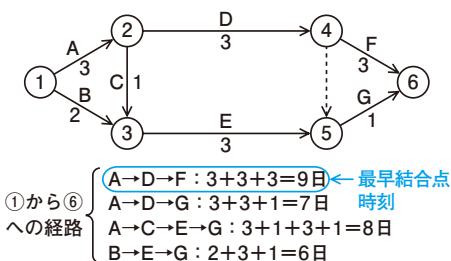
【解答】ウ

アローダイアグラムは、あるプロジェクトの開始から終了までに行う作業を矢印で表し、各作業の結合点を○印で示して、プロジェクト全体の作業経路を表した図です。



●プロジェクトの所要日数を求める

ある結合点において、最も早く次の作業を開始できる時刻を、**最早結合点時刻**といいます。最早結合点時刻を求めるには、スタートからその結合点までの作業経路の所要時間を合計します（ダミー作業の所要時間はゼロで計算します）。経路が複数ある場合は、最も大きい合計所要時間が、その結合点における最早結合点時刻になります。



作業終了時点の最早結合点時刻が、プロジェクト全体の所要時間になります。

●クリティカルパスを求める

プロジェクト開始から終了までの各作業のうち、時間的余裕のない一連の作業経路を**クリティカルパス**といいます。クリティカルパス上にある作業が遅れると、プロジェクト全体の所要時間が遅れることになります。

アローダイアグラムでは、開始から終了に至る複数の経路のうち、合計所要日数の最も多い経路がクリティカルパスになります。

●所要期間の短縮

所要時間の短縮方法として、クラッキングとファストトラッキングがあります。

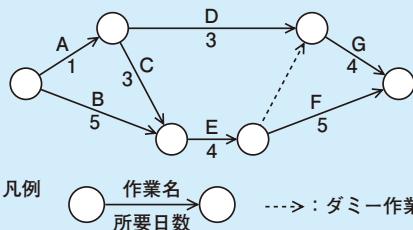
クラッキングとは、新たな人員や予算を投入して、作業期間を短縮する方法です。

ファストトラッキングとは、通常は順番にすすめる作業を同時並行ですすめることで、作業期間を短縮する方法です。

クラッキングもファストトラッキングも、クリティカルパス上の作業を短縮しなければ、プロジェクト全体の所要時間短縮にはつながりません。

例題

プロジェクトのタイムマネジメントのためには次のアローダイアグラムを作成した。クリティカルパスはどれか。



- | | |
|----------------------------|------------------------|
| ア
ウ | イ
エ |
| A → C → E → F
B → E → F | A → D → G
B → E → G |

【解法】 **ア**～**エ**の各作業経路のうち、所要時間の合計が最も多い経路がクリティカルパスになります。**ア**：13日、**イ**：8日、**ウ**：14日、**エ**：13日より、正解は**ウ**。

【解答】 **ウ**



出題予測率

70%

タスクスケジューリング

テクノロジ系

本書掲載
予想1問15／予想2問14

1個のCPUで複数のタスク(処理の実行単位)を同時に実行するには、1つのタスクがCPUを独占することのないように、システムが各タスクのCPU使用時間を細かく切り替えながら処理をすすめます。これをタスクスケジューリングといい、主な方式として、**ラウンドロビン方式**と**優先度順方式**があります。

覚える スケジューリング方式

ラウンドロビン方式

各タスクに一定のCPU使用時間を均等に割り当て、順番に切り替えていく方式。

優先度順方式

各タスクに優先度を設け、優先度の高いタスクから順にCPUを使用する方式。

ラウンドロビンや優先度順方式では、CPUを使用するタスクをOS側が管理し、CPUを使用中のタスクを強制的に切り替えます。このようなスケジューリングの方式を**プリエンプティブ**といいます。

一方、CPUを使用中のタスクが終了するか、自分から他のタスクに譲るまで処理を中断できない方式を**ノンプリエンプティブ**といいます。



出題予測率

60%

開発工数の計算

マネジメント系

工数とは

1人で行うと1日かかる作業量(工数)を**1人日**といいます。一般に、a人が行ってb日かかる作業量は、 $(a \times b)$ 人日になります。

ア 2 イ 7 ウ 10 エ 20

生産性とは

1人日の作業量が、ソースプログラム1,000ステップ(ステップはプログラムの行数のこと)に相当する場合の作業の生産性を、**1,000ステップ／人日**と表します。

一般に、生産性がpステップ／人日の場合、sステップのプログラミング作業に必要な工数は $(s \div p)$ 人日です。

例題

10人が0.5kステップ／人日の生産性で作業するとき、30日間を要するプログラミング作業がある。10日目が終了した時点で作業が終了したステップ数は、10人の合計で30kステップであった。予定の30日間でプログラミングを完了するためには、少なくとも何名の要員を追加すればよいか。ここで、追加する要員の生産性は、現在の要員と同じとする。

【解法】 10人で30日間かかる工数は $10 \times 30 = 300$ 人日です。生産性が0.5kステップ／人日の場合、全体の作業量は $0.5k \times 300 = 150k$ ステップになります。

一方、10人が10日間作業して30kステップ終了した場合、 $10 \times 10 = 100$ 人日で30kステップなので、生産性は $30k \div 100 = 0.3k$ ステップしかありません。

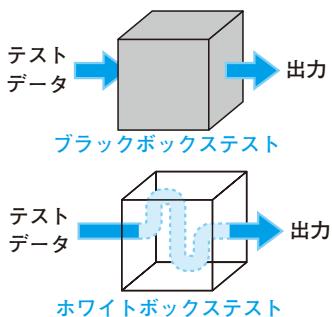
残りの作業量は $150k - 30k = 120k$ ステップです。生産性を0.3kステップ／人日とすると、工数は $120k \div 0.3k = 400$ 人日となるので、20日で完了するには $400 \div 20 = 20$ 人の要員が必要です。

以上から、追加する要因は $20 - 10 = 10$ 人となります。

【解答】 ウ

プログラムをテストするために用意する一連のデータをテストケースといいます。テストケースとしてどのようなデータを用意するかは、テストがブラックボックステストかホワイトボックステストによって異なります。

ブラックボックステストは、プログラムの内部構造には関知せず、与えられたデータに対して、あらかじめ決められたとおりに動作するかどうかを確認するテストです。これに対し**ホワイトボックステスト**は、プログラムの内部構造に着目し、ロジックが正しいかどうかを確認します。

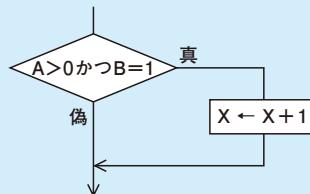


覚える ホワイトボックステストのテストケース

命令網羅	プログラム中のすべての命令を実行する
判定条件網羅	プログラム中のすべての分岐経路を実行する
条件網羅	個々の条件の真偽を少なくとも1回実行する（判定条件が複数の条件で構成される場合は、個々の条件の真偽を少しづつ実行）。
判定条件／条件網羅	条件網羅だけではすべての経路が実行されない場合があるので、条件網羅と判定条件網羅を組み合わせ、すべての経路とすべての条件の真偽を少なくとも1回実行する
複数条件網羅	判定条件のすべての真偽の組合せを網羅し、すべての命令を実行する

例題

図の構造をもつプログラムに対して、ホワイトボックステストのテストケースを設計するとき、少なくとも実施しなければならないテストケース数が最大になるテスト技法はどれか。



【解法】図のプログラムのテストケースとしては、次の4通りが考えられます。

- ① 「 $A > 0$ 」が真、「 $B = 1$ 」が真（例： $A = 1, B = 1$ ）
- ② 「 $A > 0$ 」が真、「 $B = 1$ 」が偽（例： $A = 1, B = 0$ ）
- ③ 「 $A > 0$ 」が偽、「 $B = 1$ 」が真（例： $A = 0, B = 1$ ）
- ④ 「 $A > 0$ 」が偽、「 $B = 1$ 」が偽（例： $A = 0, B = 0$ ）

最低限必要なテストケースは、それぞれ次のようにになります。

- ア 条件網羅：各条件の真偽を1回ずつ実行（②と③または①と④）
- イ 判定条件網羅：判定条件の真偽を1回ずつ実行（①と②、①と③、①と④）
- ウ 複数条件網羅：各条件の真偽の組合せをすべて実行（①～④すべて）
- エ 命令網羅：命令をすべて実行（①）

【解答】 ウ



出題予測率

60%

スタック／キュー

テクノロジ系

本書掲載
予想2問7／予想3秋問7

● スタック

「**スタック**」は、複数のデータを格納しておくデータの置き場所の一種です。スタックの特徴は、データを格納するときは下から上に積み上げていき、取り出すときは一番上にあるデータから取り出すということです。後に入れたデータほど先に取り出すので、この特徴を「**後入れ先出し**」といいます。

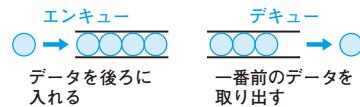
また、スタックにデータを格納することを**プッシュ**、スタックからデータを取り出すことを**ポップ**といいます。



● キュー

「**キュー**」もスタックと同様、複数のデータを格納しておくデータの置き場所です。キューは入口と出口が別の方通行のトンネルで、データを入れると一番後ろに格納され、取り出すときは先頭から取り出されます。後に入れたデータを先に取り出すので、この特徴を「**先入れ先出し**」といいます。

また、キューにデータを格納することを**エンキュー**、キューからデータを取り出すことを**デキュー**といいます。



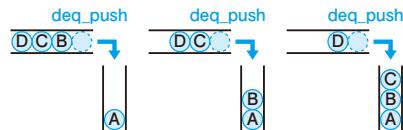
例題

四つのデータ A, B, C, D がこの順に入っているキューと空のスタックがある。手続

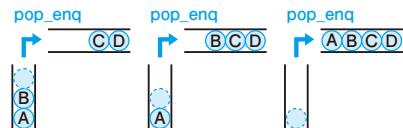
`pop_enq`, `deq_push` を使ってキューの中のデータを D, C, B, A の順に並べ替えるとき、`deq_push` の実行回数は最小で何回か。ここで、`pop_enq` はスタックから取り出したデータをキューに入れる操作であり、`deq_push` はキューから取り出したデータをスタックに放入する操作である。

ア 2 イ 3 ウ 4 エ 5

【解法】キューの末尾にある D が先頭にくるようにしたいので、とりあえず D が先頭にくるまで、キューからデータを取り出します。キューからデータを取り出すには、手続 `deq_push` を使います。



上のように、`deq_push` を 3 回実行すると、キューの先頭が D になります。この後、手続 `pop_enq` を 3 回実行すると、C, B, A の順にデータがキューに入ります。



以上から、手続 `deq_push` の実行回数は最小で 3 回です。

【解答】 イ

出題予測率

60%

ファンクションポイント法

マネジメント系

本書掲載
予想1問44／予想2問44

「**ファンクションポイント法**」は、システム開発のコストを見積もる手法のひとつです。システムの機能を入出力データやファイル、画面などの個数によって計測し、開発難易度や特性に応じて点数を付けます。算出した点数の合計によって、システムの規模を見積もります。開発規模を客観的な点数で表せるので、顧客にも説明しやすいメリットがあります。

覚える 主な見積もり手法

ファンクション ポイント法	プログラムの機能を個数や特性によって 点数化
プログラムス テップ法	プログラムごとのステップ数を積算
標準タスク法	必要な作業の WBS を作成
経験値による 見積もり	過去の類似例から開発規模を見積もる

例題

表の機能と特性をもったプログラムのファンクションポイント値はいくらか。ここで、複雑さの補正係数は 0.75 とする。

ユーザファンクションタイプ	個数	重み付け係数
外部入力	1	4
外部出力	2	5
内部論理ファイル	1	10
外部インターフェースファイル	0	7
外部照会	0	4

ア 18 イ 24 ウ 30 エ 32

【解法】各機能の個数に重み付け係数を掛けたものを合計すると、 $1 \times 4 + 2 \times 5 + 1 \times 10 = 24$ 。この点数に補正係数 0.75 を掛けると、 $24 \times 0.75 = 18$ となります。

【解答】ア



出題予測率
60%

再帰呼出し

テクノロジ系

◆本書掲載
予想1問9／予想2問9
予想3問8

処理の中で自分自身を呼び出すことを**再帰呼出し**といいます。

例題

$n!$ の値を、次の関数 $F(n)$ によって計算する。
乗算の回数を表す式はどれか。

$$F(n) = \begin{cases} 1 & (n = 0) \\ n \times F(n - 1) & (n > 0) \end{cases}$$

ア $n - 1$
ウ n^2

イ n
エ $n!$

【解法】 $n = 0$ のとき、 $F(0) = 1$ なので、乗算の回数は 0 回です。また、 $n > 0$ のときは次のようになります。

$$\begin{aligned} F(1) &= 1 \times F(0) = 1 \times 1 && \leftarrow \text{乗算回数 1 回} \\ F(2) &= 2 \times F(1) = 2 \times 1 \times F(0) \\ &= 2 \times 1 \times 1 && \leftarrow \text{乗算回数 2 回} \\ &\vdots \\ F(n) &= n \times F(n - 1) = n \times (n - 1) \times F(n - 2) \\ &= n \times (n - 1) \times (n - 2) \times \cdots \times 1 \times 1 && \leftarrow \text{乗算回数 } n \text{ 回} \end{aligned}$$

【解答】イ

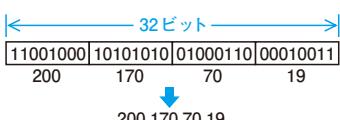
出題予測率
50%

IPアドレスの仕組み

テクノロジ系

◆本書掲載
予想1問27／予想2問26

IP アドレスは、ネットワークに接続するすべての端末に割り当てられる識別番号です。現在一般的な IP アドレスは 32 ビット（2 進数 32 桁）の長さがありますが、表記するときは 8 ビットずつ区切って、4 個の 10 進数で表します。

IP アドレスの例：


200.170.70.19

●グローバルIPアドレスとプライベートIPアドレス
インターネットに接続する端末に割り当てられる IP アドレスを、**グローバル IP アドレス**といいます。グローバル IP アドレスは、世界で一意に割り当てられます。

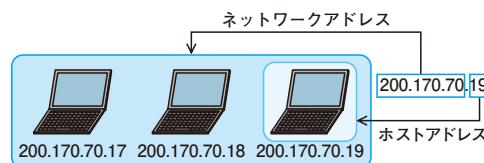
これに対し、社内ネットワークなどで使う IP アドレスを**プライベート IP アドレス**といいます。プライベート IP アドレスを割り当てられた端末は、直接インターネットに接続することはできま

せん。社内ネットワークとインターネットを接続するルータを介して、プライベートIPアドレスをグローバルIPアドレスに変換する必要があります。このようなルータの機能を、NAT (Network Address Translation) またはIPマスカレードといいます。

なお、プライベートIPアドレスとして使用できる範囲は、IPアドレスのクラスごとに定められています。

● IPアドレスのクラス

IPアドレスは、ネットワークアドレスとホストアドレスに区分されます。ネットワークアドレスは、端末が所属するネットワークを示すアドレスで、ホストアドレスは、そのネットワーク内の各端末に一意に割り当てられるアドレスです。



IPアドレスは、ネットワーク部の長さによって、次のようにクラスA～クラスDに分類されます。

クラス	範囲*
クラスA	ネットワークアドレス8ビット+ホストアドレス24ビット 0.0.0～127.255.255.255
クラスB	ネットワークアドレス16ビット+ホストアドレス16ビット 128.0.0.0～191.255.255.255
クラスC	ネットワークアドレス24ビット+ホストアドレス8ビット 192.0.0.0～223.255.255.255
クラスD	ネットワークアドレス32ビット 224.0.0.0～239.255.255.255

*色文字の部分がネットワークアドレスを示す

1つのネットワークに接続できる端末の台数は、ホストアドレスの長さによって決まります。たとえば、クラスCのホストアドレスは8ビットなので、1つのネットワークにつき $2^8 = 256$ 通りのホストアドレスがあります。ただし、ホストアドレスの先頭（ビットがすべて0のアドレス）と末尾（ビットがすべて1のアドレス）は、端末に割り当てできない決まりなので、割り当てる可能な端末は最大254台になります。

$$\text{割り当てる可能な台数} = 2^H - 2 \text{ 台}$$

*H: ホストアドレスのビット数

● サブネットマスク

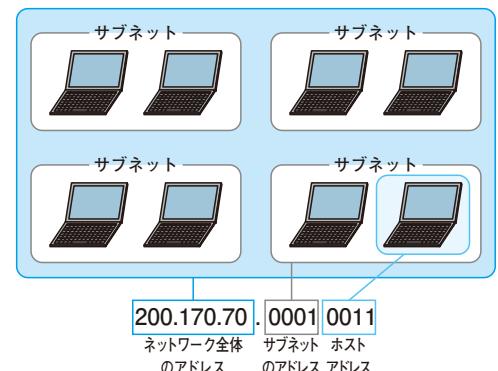
IPアドレスのうち、どこまでをネットワークアドレスとして扱うかは、IPアドレスと組になっているサブネットマスクを使って設定します。サブネットマスクを2進数で表すと、上位ビットが1の連続になります。この1の部分に対応する部分が、IPアドレスのネットワークアドレスになります。



サブネットマスクを調整すると、1つのネットワークを複数のサブネットに分割できます。たとえば、クラスCのネットワークアドレスは24ビットなので、サブネットマスクは通常「255.255.255.0」です。これを「255.255.255.240」とすると、ネットワークアドレスが28ビット、ホストアドレスが4ビットになります。



追加でネットワークアドレスに割り当てた4ビットは、サブネットのアドレスとして使用できます。



基本情報技術者

予想問題①

• 科目 A 18

(全 60 問 試験時間 : 90 分)

• 科目 B 54

(全 20 問 試験時間 : 100 分)



基本情報技術者試験 予想問題①

科目A



問 01

16進数の小数 0.248 を 10進数の分数で表したものはどれか。

ア $\frac{31}{32}$

イ $\frac{31}{125}$

ウ $\frac{31}{512}$

エ $\frac{73}{512}$



問 02

桁落ちの説明として、適切なものはどれか。

- ア 値がほぼ等しい浮動小数点数同士の減算において、有効桁数が大幅に減ってしまうことである。
- イ 演算結果が、扱える数値の最大値を超えることによって生じるエラーのことである。
- ウ 浮動小数点数の演算結果について、最小の桁よりも小さい部分の四捨五入、切上げ又は切捨てを行うことによって生じる誤差のことである。
- エ 浮動小数点数の加算において、一方の数値の下位の桁が結果に反映されないことがある。



問 03

ある工場では、同じ製品を独立した二つのライン A, B で製造している。ライン A では製品全体の 60%を製造し、ライン B では 40%を製造している。ライン A で製造された製品の 2%が不良品であり、ライン B で製造された製品の 1%が不良品であることが分かっている。いま、この工場で製造された製品の一つを無作為に抽出して調べたところ、それは不良品であった。その製品がライン A で製造された確率は何%か。

ア 40

イ 50

ウ 60

エ 75



問 04

サンプリング周波数 40kHz、量子化ビット数 16 ビットで A/D 変換したモノラル音声の 1 秒間のデータ量は、何 kB イトとなるか。ここで、1kB は 1,000 バイトとする。

ア 20

イ 40

ウ 80

エ 640

解説

問01 16進数の小数

■■■ 40%

10進数の小数は、0.1が $\frac{1}{10}$ 、0.01が $\frac{1}{100}$ 、0.001が $\frac{1}{1000}$ を表しています。これと同様に、16進数の小数は、0.1が $\frac{1}{16}$ 、0.01が $\frac{1}{16^2}$ 、0.001が $\frac{1}{16^3}$ を表します。

したがって16進数0.248は、10進数では次のようにになります。

$$\begin{aligned} 2 \times \frac{1}{16} + 4 \times \frac{1}{16^2} + 8 \times \frac{1}{16^3} &= \frac{2}{16} + \frac{4}{16^2} + \frac{8}{16^3 \times 16} = \frac{2}{16} + \frac{4}{16^2} + \frac{1}{16^2 \times 2} \\ &= \frac{2 \times 16 \times 2 + 4 \times 2 + 1}{16^2 \times 2} = \frac{64 + 8 + 1}{512} = \frac{73}{512} \end{aligned}$$

正解は**工**です。

問02 衍落ち キホン!

■■■ 20%

値がほぼ等しい浮動小数点数同士を引き算すると、演算結果が小さすぎるために、有効桁数が減ってしまいます。このために生じる誤差を**衍落ち**といいます。

- **ア** 正解です。
- × **イ** オーバフローの説明です。
- × **ウ** 丸め誤差の説明です。
- × **エ** 情報落ちの説明です。

有効数字	1.41421356237309…
-	1.4142134
	0.0000001 → 0.1×10^{-6}
	衍落ち

問03 ベイズの定理

■■■ 20%

ラインAで製造した製品60%のうちの2%と、ラインBで製造した製品40%のうちの1%が不良品となります。

ラインAの不良品： $60\% \times 2\% = 1.2\%$

ラインBの不良品： $40\% \times 1\% = 0.4\%$

ラインAの不良品+ラインBの不良品＝1.6%のうち、1.2%がラインAで製造したものですから、不良品がラインAで製造された確率は、 $1.2 \div 1.6 = 0.75 \rightarrow 75\%$ になります。正解は**工**です。

$$\text{ラインAである確率} \times \text{ラインAの製品のうち不良品である確率} = \frac{60\% \times 2\%}{\text{製品全体のうち不良品である確率}} = \frac{60\% \times 2\%}{1.2\% + 0.4\%} = 0.75$$

問04 A/D変換

■■■ 20%

サンプリング周波数は、アナログデータを1秒間に何回サンプリング（標本化）するかを表します。40kHzなら1秒間に40,000回サンプリングするので、40,000個のデータが得られます。

量子化ビット数は、サンプリングデータ1個当たりのサイズです。16ビットなら、1個当たりのサイズは $16 \div 8 = 2$ バイトになります。

以上から、1秒間のデータ量は、 $40,000 \times 2 = 80,000$ バイト＝ $80 \times 1,000$ バイト＝80kバイトになります。正解は**ウ**です。

合格のカギ

覚えよう！ 問02

衍落ちといえば

- ほぼ同じ数同士の減算で有効桁数が減ってしまうことによる誤差

オーバフローといえば

- 演算結果が、処理できる数値の最大値を超ってしまうこと

丸め誤差といえば

- 四捨五入や切捨て、切上げによる誤差

情報落ちといえば

- 大きな数と小さな数を加算したとき、小さな数が演算結果に反映されないことにによる誤差

ベイズの定理 問03

製品が不良品である確率をP(E)、ラインAで製造されたものである確率をP(A)、ラインAで製造されたもののうち、不良品である確率をP_A(E)とする。不良品のうち、ラインAで製造されたものである確率P_E(A)は、次のようなベイズの定理で求めることができる。

$$P_E(A) = \frac{P(A) P_A(E)}{P(E)}$$

覚えよう！ 問04

サンプリング周波数 といえば

- アナログデータを1秒間に何回サンプリングするか
- 量子化ビットといえば
サンプリングデータ1個当たりのサイズ

○ 解答 ○

- | | |
|-------|-------|
| 問01 工 | 問02 ア |
| 問03 エ | 問04 ウ |

問 05

表は、文字列を検査するための状態遷移表である。検査では、初期状態を a とし、文字列の検査中に状態が e になれば不合格とする。

解答群で示される文字列のうち、不合格となるものはどれか。ここで、文字列は左端から検査し、解答群中の△は空白を表す。

		文字				
		空白	数字	符号	小数点	その他
現在の状態	a	a	b	c	d	e
	b	a	b	e	d	e
	c	e	b	e	d	e
	d	a	e	e	e	e

ア + 0010

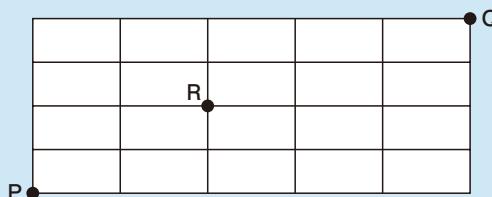
イ - 1

ウ 12.2

エ 9.△

問 06

図の線上を、点 P から点 R を通って、点 Q に至る最短経路は何通りあるか。



ア 16

イ 24

ウ 32

エ 60

問 07

AI におけるディープラーニングの特徴はどれか。

- ア “A ならば B である” というルールを人間があらかじめ設定して、新しい知識を論理式で表現したルールに基づく推論の結果として、解を求めるものである。
- イ 厳密な解でなくてもなるべく正解に近い解を得るようにする方法であり、特定分野に特化せずに、広範囲で汎用的な問題解決ができるようにするものである。
- ウ 人間の脳神経回路を模倣して、認識などの知能を実現する方法であり、ニューラルネットワークを用いて、人間と同じような認識ができるようにするものである。
- エ 判断ルールを作成できる医療診断などの分野に限定されるが、症状から特定の病気にはまり込むといった、確率的に高い判断ができる。

解説

問 05 状態遷移表 キホン!

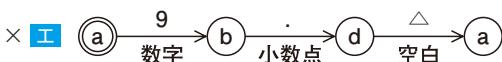
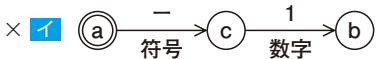
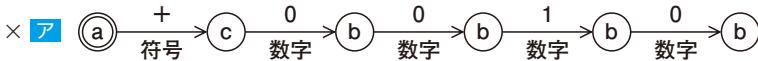
合格のカギ

■ 20%

初期状態 a から開始します。たとえば「+ 0010」の場合、1 文字目は符号な


合格のカギ

ので、現在の状態 a の“符号”の列をみます。すると c があるので、状態 c に遷移します。2 文字目は数字なので、現在の状態 c の“数字”的列をみます。すると b があるので、状態 b に遷移します。以下同様にして、状態 e に遷移するものを探します。



問06 最短経路の組合せ

 20%

点 P から点 Rまでの最短経路は、かならず横に 2 回、縦に 2 回の移動になります。横にすすむ順番と縦にすすむ順番は任意なので、

横横縦縦、横縦横縦、横縦縦横、縦横横縦、縦横縦横、縦縦横横

の 6 通りです。これは「4 回の移動のうち、どの 2 回を横（または縦）にするか」という組合せなので、次のような計算式で表せます。

$${}_4C_2 = \frac{4 \times 3}{2 \times 1} = 6 \text{ 通り}$$

点 R から点 Q への移動も同様に考えます。最短距離は 5 回の移動のうち、横に 3 回、縦に 2 回の移動になります。横にすすむ順番と縦にすすむ順番は任意なので、「5 回の移動のうち、どの 2 回を縦にするか」という組合せです。

$${}_5C_2 = \frac{5 \times 4}{2 \times 1} = 10 \text{ 通り}$$

点 P から点 Q までの経路が 6 通り、点 Q から点 R までが 10 通りなので、全体では $6 \times 10 = 60$ 通りとなります。正解は エ です。

問07 ディープラーニング

 20%

ディープラーニング（深層学習）とは、ニューラルネットワークを用いた機械学習によって、コンピュータに人間と同じような認識ができるようにする技術です。ニューラルネットワークの階層を深くすることにより、識別するための特徴をデータから自動的に学習でき、認識精度を高めることができます。最近では音声認識や画像認識などへの応用がすすんでいます。

× ア ルールベースシステムの説明です。

× イ メタヒューリスティックの説明です。

○ ウ 正解です。

× エ エキスパートシステムの説明です。

 覚えよう！ 間06

順列といえば

• n 個から r 個選んで順番に並べる並べ方

$${}_nPr = \frac{n!}{(n-r)!}$$

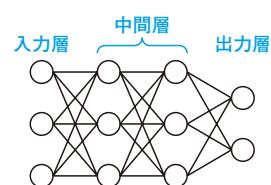
組合せといえば

• n 個から並び順を考慮しないで r 個選ぶ組合せ

$${}_nCr = \frac{{}_nPr}{r!} = \frac{n!}{r!(n-r)!}$$

 ニューラルネットワーク 間07

人間の脳神経のネットワークをモデル化したもの。



一般に、中間層を 2 層以上に多層化したモデルをディープラーニングという。

 機械学習 間07

テキスト、画像、音声などのデータから、コンピュータが知識やルールを自動的に学習する技術のこと。

○ 解答

問05 ウ 問06 エ
問07 ウ

問 08 リストを二つの1次元配列で実現する。配列要素 $box[i]$ と $next[i]$ の対がリストの一つの要素に対応し、 $box[i]$ に要素の値が入り、 $next[i]$ に次の要素の番号が入る。配列が図の状態の場合、リストの3番目と4番目との間に値がHである要素を挿入したときの $next[8]$ の値はどれか。ここで、 $next[0]$ がリストの先頭(1番目)の要素を指し、 $next[i]$ の値が0である要素はリストの最後を示し、 $next[i]$ の値が空白である要素はリストに連結されていない。

box	0	1	2	3	4	5	6	7	8	9
		A	B	C	D	E	F	G	H	I

next	0	1	2	3	4	5	6	7	8	9
	1	5	0	7		3		2		

ア 3

イ 5

ウ 7

エ 8

問 09

次の関数 $f(n, k)$ がある。 $f(4, 2)$ の値は幾らか。

$$f(n, k) = \begin{cases} 1 & (k=0), \\ f(n-1, k-1) + f(n-1, k) & (0 < k < n), \\ 1 & (k=n). \end{cases}$$

ア 3

イ 4

ウ 5

エ 6

問 10

動作クロック周波数が 700MHz の CPU で、命令実行に必要なクロック数及びその命令の出現率が表に示す値である場合、この CPU の性能は約何 MIPS か。

命令の種別	命令実行に必要なクロック数	出現率 (%)
レジスタ間演算	4	30
メモリ・レジスタ間演算	8	60
無条件分岐	10	10

ア 10

イ 50

ウ 70

エ 100

解説

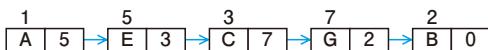
問 08 リスト構造

リストは、要素の値と次の要素へのポインタを一組として、各要素を連結するものです。問題文では、要素の値を $box[i]$ 、次の要素へのポインタを $next[i]$ に格納し、 $box[i]$ と $next[i]$ を一組の要素とみなします。先

合 格 の カ キ

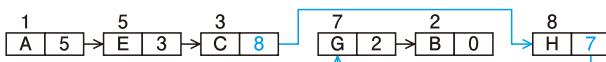
■ 30%

頭の要素を指す `next[0]` の値は 1, `next[1]` の値は 5, `next[5]` の値は 3, `next[3]` の値は 7, `next[7]` の値は 2, `next[2]` の値は 0 です。`next[i]` の値が 0 である要素はリストの最後を示すので、このリストは $1 \rightarrow 5 \rightarrow 3 \rightarrow 7 \rightarrow 2$ の順番に連結されています。



3番目の要素であるCと、4番目の要素であるGの間にHを挿入します。Hはbox[8]に格納されているので、まずnext[8]にGの要素番号7を設定します。これで、Hの次の要素がGになります。次に、Cの次の要素をHにするため、next[3]の値を8に書き換えます。以上で、CとGの間にHが挿入されます。

次の図のように、next[8] の値は 7 になるので、正解は **ウ** です。



合格のカギ

リスト

問 08

各要素は値とポインタの組で構成され、各要素のポインタに次の要素のアドレスが格納されているデータの集合（線形リストともいう）。ポインタをたどっていくことで各要素に順番にアクセスできる。要素の挿入や削除はポインタを書き換えるだけができるが、要素を検索するときは要素を先頭から順にたどっていく必要があるため効率が悪い。



問09 再帰的関数 三キホン!

60%

関数 $f(n, k)$ は、 $k = 0$ または $k = n$ のとき値 1 を返し、それ以外のときは $f(n - 1, k - 1) + f(n - 1, k)$ を再帰的に呼び出す **再帰的関数** です。 $f(4, 2)$ の計算式を展開すると、次のようになります。

$$\begin{aligned}
 f(4, 2) &= f(4-1, 2-1) + f(4-1, 2) \\
 &= f(3, 1) + f(3, 2) \\
 &= f(2, 0) + f(2, 1) + f(2, 1) + f(2, 2) \\
 &= 1 + f(1, 0) + f(1, 1) + f(1, 0) + f(1, 1) + 1 \\
 &= 1 + 1 + 1 + 1 + 1 + 1 + 1 \\
 &= 6
 \end{aligned}$$

以上から、正解は **エ** です。

問10 MIPS 計算 キホン!

40%

MIPS は、コンピュータが 1 秒間に何百万個の命令を実行できるかを表します。問題文の表から、1 命令に必要なクロック数の平均は

$$4 \times 0.3 + 8 \times 0.6 + 10 \times 0.1 = 1.2 + 4.8 + 1 = 7 \text{ クロック}$$

レジスタ間演算 無条件分岐
 ループ・レジスタ間演算

となります。動作クロック周波数が 700MHz の場合、1 秒間のクロック数は 700×10^6 クロックになるので、1 秒間に実行できる命令は

$$700 \times 10^6 \div 7 = 100 \times 10^6 \text{ 約 } \leftarrow 10^6 = 100 \text{ 萬}$$

となります。以上から、CPUの性能は100MIPSです。正解はエーです。

 MIPS

問 10

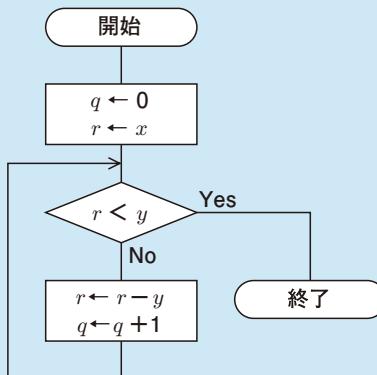
Million Instructions Per Second の略。プロセッサが 1 秒間に実行する命令数を百万単位で表したもの。

○ 解答

問08 ウ 問09 イ
問10 イ

問 11

x と y を自然数とするとき、流れ図で表される手続を実行した結果として、適切なものはどれか。



	q の値	r の値
ア	$x \div y$ の余り	$x \div y$ の商
イ	$x \div y$ の商	$x \div y$ の余り
ウ	$y \div x$ の余り	$y \div x$ の商
エ	$y \div x$ の商	$y \div x$ の余り

問 12

ディープラーニングの学習に GPU を用いる利点として、適切なものはどれか。

- ア 各プロセッサコアが独立して異なるプログラムを実行し、異なるデータを処理できる。
- イ 汎用の行列演算ユニットを用いて、行列演算を高速に実行できる。
- ウ 浮動小数点演算ユニットをコプロセッサとして用い、浮動小数点演算ができる。
- エ 分岐予測を行い、パイプラインの利用効率を高めた処理を実行できる。

問 13

内部割込みに分類されるものはどれか。

- ア 商用電源の瞬時停電などの電源異常による割込み
- イ ゼロで除算を実行したことによる割込み
- ウ 入出力が完了したことによる割込み
- エ メモリパリティエラーが発生したことによる割込み

解説

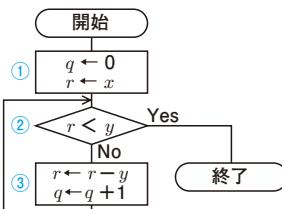
問11 流れ図 キホン!

40%

例として、 $x = 10$, $y = 3$ として流れ図をトレースし、 q と r の値がどのように変化するか見てみましょう。

手続きが終了すると、 q には3 ($10 \div 3$ の商) が、 r には1 ($10 \div 3$ の余り) が入ります。以上から、正解はイです。

位置	q	r	備考
開始	-	-	$x=10$, $y=3$
①	0	10	$q \leftarrow 0$, $r \leftarrow x$
②	0	10	$r < y \Rightarrow 10 < 3$ は No
③	1	7	$q \leftarrow 0+1$, $r \leftarrow 10-3$
②	1	7	$r < y \Rightarrow 7 < 3$ は No
③	2	4	$q \leftarrow 1+1$, $r \leftarrow 7-3$
②	2	4	$r < y \Rightarrow 4 < 3$ は No
③	3	1	$q \leftarrow 2+1$, $r \leftarrow 4-3$
②	3	1	$r < y \Rightarrow 1 < 3$ は Yes
終了	3	1	



問12 GPGPU

20%

GPU (Graphics Processing Unit) は、画像処理のための演算を高速に行うためのプロセッサです。GPUは、画像を構成する複数のピクセル（画素）を高速に処理するために、**行列演算ユニット**を備えています。ディープラーニングも、入力された信号を並列に処理するために大量の行列演算が必要となるため、GPUを利用します。正解はイです。

GPUの行列演算ユニットを汎用目的に用いる技術を**GPGPU** (General-Purpose computing on Graphics Processing Unit)といいます。

問13 内部割込み

40%

実行中のプログラムを中断し、CPUに強制的に別の処理を実行させることを**割込み**といいます。割込みには**内部割込み**と**外部割込み**があります。外部割込みが周辺機器などのハードウェアによって発生するのに対し、内部割込みはプログラムの実行によって発生します。

- × ア 外部割込みの一種（機械チェック割込み）です。
- イ 正解です。
- × ウ 外部割込みの一種（入出力割込み）です。
- × エ 外部割込みの一種（機械チェック割込み）です。

合格のカギ

問11

参考 変数に実際の値をあてはめて流れを追ってみよう。



覚えよう！

問13

内部割込みといえば

- プログラムによって発生

外部割込みといえば

- ハードウェアによって発生

入出力割込み	入出力処理の完了やエラーで発生
タイマ割込み	タイマによって発生
機械チェック割込み	ハードウェアの障害などによって発生
コンソール割込み	割込みスイッチの操作で発生

○ 解答 ○

問11 イ 問12 イ
問13 イ



問 14

コンピュータを2台用意しておき、現用系が故障したときは、現用系と同一のオンライン処理プログラムをあらかじめ起動して待機している待機系のコンピュータに速やかに切り替えて、処理を続行するシステムはどれか。

- ア コールドスタンバイシステム
- イ ホットスタンバイシステム
- ウ マルチプロセッサシステム
- エ マルチユーザシステム



問 15

三つのタスクの優先度と、各タスクを単独で実行した場合のCPUと入出力(I/O)装置の動作順序と処理時間は、表のとおりである。優先度方式のタスクスケジューリングを行うOSの下で、三つのタスクが同時に実行可能状態になってから、全てのタスクの実行が終了するまでの、CPUの遊休時間は何ミリ秒か。ここで、CPUは1個であり、1CPUは1コアで構成され、I/Oは競合せず、OSのオーバヘッドは考慮しないものとする。また、表中の()内の数字は処理時間を示すものとする。

優先度	単独実行時の動作順序と処理時間(ミリ秒)
高	CPU(3) → I/O(5) → CPU(2)
中	CPU(2) → I/O(6) → CPU(2)
低	CPU(1) → I/O(5) → CPU(1)

ア 2

イ 3

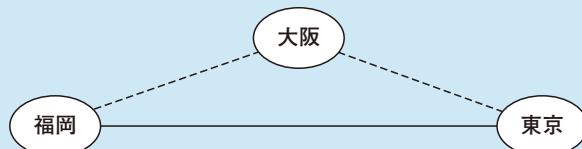
ウ 4

エ 5



問 16

東京と福岡を結ぶ実線の回線がある。東京と福岡の間の信頼性を向上させるために、大阪を経由する破線の迂回回線を追加した。迂回回線追加後における、東京と福岡の間の稼働率は幾らか。ここで、回線の稼働率は、東京と福岡、東京と大阪、大阪と福岡の全てが0.9とする。



ア 0.729

イ 0.810

ウ 0.981

エ 0.999

解説

問14 デュプレックスシステム

30%



現用系と待機系の2系統を用意したシステムを**デュプレックスシステム**といいます。このうち、待機系のコンピュータにふだんから現用系と同じプログラムを起動させておく方式を**ホットスタンバイ**、待機系にふだんは別の処理をさせておく方式を**コールドスタンバイ**といいます。

- × **A** **コールドスタンバイシステム**では、待機系はふだんは現用系と別の処理を実行します。
- **イ** 正解です。
- × **ウ** **マルチプロセッサシステム**は、1台のコンピュータに複数のCPUを搭載したシステムです。
- × **エ** **マルチユーザシステム**は、1台のコンピュータを複数のユーザが同時に利用できるシステムです。



問14

ホットスタンバイシステムといえれば

- 待機系が現用系と一緒に稼働
- **コールドスタンバイシステム**といえれば
- 待機系はふだん現用系と違う処理を担当

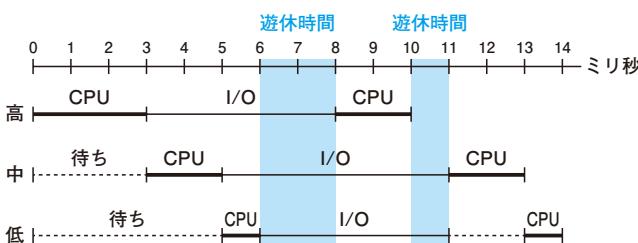
問15 タスクスケジューリング キポン!

70%

問15

優先度「高」のタスクは、必要なときはいつでも優先的にCPUを使用できます。優先度「中」のタスクは、優先度「高」のタスクが使用していないときに限って、CPUを使用できます。優先度「低」のタスクは、他のタスクが使用していないときだけ、CPUを使用できます。

3つのタスクのCPUとI/Oのスケジュールは次のようになります。なお、各タスクのI/Oは競合しないので、各タスクが重複してもかまいません。



CPUが使用されていない遊休時間は、図から6~8ミリ秒と、10~11秒の合計3ミリ秒になります。正解は**イ**です。

問15

システムに投入されたタスクやジョブの実行順序を、その特性や優先順位に応じて決定すること。ラウンドロビン、優先度順など的方式がある。

問16 回線の稼働率 キポン!

90%

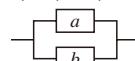
問16

直列システムの稼働率

$$a \times b = a - a \times b$$

並列システムの稼働率

$$1 - (1 - a) \times (1 - b)$$



問16

ある期間中に、装置やシステムが正常に稼働している割合のこと。稼働時間 ÷ 全運用時間で求められる。

○ 解答 ○

問14 イ 問15 イ
問16 ウ

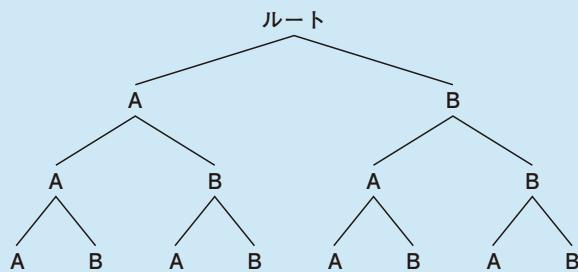


問 17

A, B という名の複数のディレクトリが、図に示す構造で管理されている。“¥B¥A¥B”がカレントディレクトリになるのは、カレントディレクトリをどのように移動した場合か。ここで、ディレクトリの指定は次の方法によるものとし、→は移動の順序を示す。

[ディレクトリ指定方法]

- (1) ディレクトリは、“ディレクトリ名¥…¥ディレクトリ名”のように、経路上のディレクトリを順に“¥”で区切って並べた後に、“¥”とディレクトリ名を指定する。
- (2) カレントディレクトリは、“.”で表す。
- (3) 1階層上のディレクトリは、“..”で表す。
- (4) 始まりが“¥”のときは、左端にルートディレクトリが省略されているものとする。
- (5) 始まりが“¥”, “.”, “..”のいずれでもないときは、左端に“.¥”が省略されているものとする。



ア ¥A → ..¥B → .¥A¥B
ウ ¥B → ¥A → ¥B

イ ¥B → .¥B¥A → ..¥B
エ ¥B¥A → ..¥B



問 18

リンクの機能として、適切なものはどれか。

- ア** 作成したプログラムをライブラリに登録する。
- イ** 実行に先立ってロードモジュールを主記憶にロードする。
- ウ** 相互参照の解決などを行い、複数の目的モジュールなどから一つのロードモジュールを生成する。
- エ** プログラムの実行を監視し、ステップごとに実行結果を記録する。

解説

問17 ディレクトリの指定

■ 20

ディレクトリとは、ディスク上に記録されているファイルの登録簿のことですが、一般には「ファイルの入れ物」と考えるとイメージしやすいでしょう。Windows などでは、ディレクトリをフォルダのアイコンで表示します。

ファイルを階層構造で管理するシステムでは、ディレクトリの中に、さらに

合格のカギ

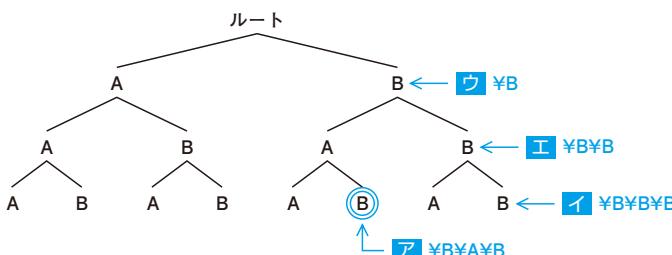
ディレクトリを作成できます。このような構造を階層ディレクトリ構造といいます。また、いちばん上位のディレクトリをルートディレクトリ、現在開いているディレクトリをカレントディレクトリといいます。

始まりが“¥”記号のときは、左端にルートディレクトリが省略されているとみなします。したがってディレクトリ“¥B¥A¥B”は、

ルートディレクトリ→B→A→B

の順にディレクトリをたどることを表します。下図の○のついたディレクトリが、“¥B¥A¥B”です。

- **ア** ¥A → ..¥B → .¥A¥B は、「ルートの下の A → 1つ上に戻り(..), その下の B → その下の A の下の B」を表します。カレントディレクトリは“¥B¥A¥B”になります。
- × **イ** ¥B → .¥B¥A → ..¥B は、「ルートの下の B → その下の B の下の A → 1つ上に戻り(..), その下の B」を表します。カレントディレクトリは“¥B¥B¥B”になります。
- × **ウ** ¥B → ¥A → ¥B は、「ルートの下の B → ルートの下の A → ルートの下の B」を表します。カレントディレクトリは“¥B”になります。
- × **エ** ¥B¥A → ..¥B は、「ルートの下の B の下の A → 1つ上に戻り(..), その下の B」を表します。カレントディレクトリは“¥B¥B”になります。

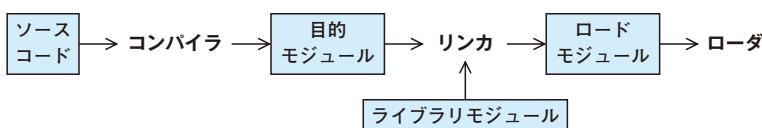


問18 リンカの機能 キホン!

20%

コンパイラによって生成された目的モジュール（オブジェクトコード）は、プログラムから呼び出す他のモジュールが含まれていなかったり、参照先アドレスが未解決のままだったりするため、そのままでは実行できません。リンカ（リンクエディタ）は、複数の目的モジュールを結合（リンク）して、実行可能なロードモジュールを生成するツールです。

- × **ア** ライブラリは、汎用性の高い目的モジュールを集めて他のプログラムから利用できるようにしたモジュール集です。プログラムをライブラリに登録するには、アーカイバなどのツールを利用します。
- × **イ** ロードモジュールを主記憶にロードするツールはローダです。
- **ウ** 正解です。
- × **エ** トレーサの説明です。



合格のカギ

問17

対策 “¥A¥B”が「ルートディレクトリ→A→B」を表すのに対し、“¥A¥B”は「カレントディレクトリ→A→B」を表す。違いに注意しよう。



○ 解答
問17 ア 問18 ウ

問 19 分解能が 8 ビットの D/A 変換器に、デジタル値 0 を入力したときの出力電圧が 0V となり、デジタル値 128 を入力したときの出力電圧が 2.5V となるとき、最下位の 1 ビットの変化によるこの D/A 変換器の出力電圧の変化は何 V か。

- ア 2.5 / 128 イ 2.5 / 255 ウ 2.5 / 256 エ 2.5 / 512

問 20 DRAM の説明として、適切なものはどれか。

- ア 1 バイト単位でデータの消去及び書き込みが可能な不揮発性のメモリであり、電源遮断時もデータ保持が必要な用途に用いられる。
イ 不揮発性のメモリで NAND 型又は NOR 型があり、SSD に用いられる。
ウ メモリセルはフリップフロップで構成され、キャッシュメモリに用いられる。
エ リフレッシュ動作が必要なメモリであり、PC の主記憶として用いられる。

問 21 関係 X と Y を自然結合した後、関係 Z を得る関係代数演算はどれか。

X		
学生番号	氏名	学部コード
1	山田太郎	A
2	情報一郎	B
3	鈴木花子	A
4	技術五郎	B
5	小林次郎	A
6	試験桃子	A

Y	
学部コード	学部名
A	工学部
B	情報学部
C	文学部

学部名	学生番号	氏名
情報学部	2	情報一郎
情報学部	4	技術五郎

- ア 射影と和 イ 選択 ウ 選択と射影 エ 選択と和



解説

問19 D/A変換器

20%

D/A変換器は、デジタル信号(D)をアナログ信号(A)に変換する装置です。分解能が8ビットなので、 $2^8 = 256$ 段階の変化に対応しています。デジタル値128のときの出力が2.5Vですから、出力電圧の変化は1段階につき $2.5 / 128V$ となることがわかります。正解はアです。

問20 メモリの種類 キホン!

80%

DRAM (Dynamic RAM) は、電源を切ると内容が消えてしまう揮発性メモリの一種です。構造が単純なので大容量化・高集積化が簡単にでき、ビット当たりの単価を安くできます。そのため、主に主記憶装置に用いられています。DRAMは、電源が入っている間は、内容を保持するために定期的な再書き込みが必要です。この動作をリフレッシュといいます。

- × ア EEPROM の説明です。
- × イ フラッシュメモリの説明です。
- ✓ ウ SRAM (Static RAM) の説明です。SRAMはDRAMと同じく電源を切ると内容が消えてしまう揮発性メモリですが、リフレッシュ動作が不要で高速なのが特徴です。ただし、フリップフロップ回路は構造が複雑なので価格が高く、大容量には向きません。そのため主にPCのキャッシングメモリに用いられています。

○ エ 正解です。

問21 関係代数演算 キホン!

50%

関係 X と Y を結合した結果は、次のようにになります。

X と Y を結合した表

学生番号	氏名	学部コード	学部名
1	山田太郎	A	工学部
2	情報一郎	B	情報学部
3	鈴木花子	A	工学部
4	技術五郎	B	情報学部
5	小林次郎	A	工学部
6	試験桃子	A	工学部

関係 Z は、上の表から「学部名」「学生番号」「氏名」の列だけを取り出しています。このように、表から特定の列だけを取り出す関係演算を射影といいます。

また、関係 Z は、上の表から学部名が“情報学部”的行だけを取り出しています。このように、表から特定の行だけを取り出す関係演算を選択といいます。

以上から、関係 Z を得る関係代数演算はウの選択と射影です。

合格のカギ

覚えよう！ 問20

DRAM といえば

- 構造が単純で安価
- リフレッシュ動作が必要
- 主記憶装置に用いる

SRAM といえば

- リフレッシュ動作が不要
- 高速だが構造が複雑で高価
- キャッシュメモリに用いる

EEPROM 問20

電気的に書換え・消去が可能な不揮発性メモリ。主に電源遮断時でも保持しておく必要がある設定情報などの記録に用いられている。

フラッシュメモリ 問20

ブロック単位でデータの書換え・消去が可能な不揮発性メモリ。NAND型とNOR型があり、デジタルカメラの記憶媒体やUSBメモリ、SSDなどに用いられている。

SSD 問20

ソリッドステートドライブ。半導体メモリを用いた記憶装置のこと。読み書きの速度が速く、省電力、小型・軽量の上、衝撃にも強い。ただし、現時点では、HDD(ハードディスクドライブ)と比べ、容量当たりの単価が高い。

覚えよう！ 問21

関係演算といえば

- 射影：特定の列を取り出す
- 選択：特定の行を取り出す
- 結合：複数の表を共通の項目をキーにして結合する

解答

問19 ア 問20 エ
問21 ウ

問 22

関係データベースの説明として、適切なものはどれか。

- ア 属性単位に、属性値とその値をもつレコード格納位置を組にして表現する。索引として利用される。
- イ データを表として表現する。表間は相互の表中の列の値を用いて関連付けられる。
- ウ レコード間の関係を、ポインタを用いたデータ構造で表現する。木構造の表現に制限される。
- エ レコード間の関係を、リンクを用いたデータ構造で表現する。木構造や網構造も表現できる。

問 23

データベースの更新前や更新後の値を書き出して、データベースの更新記録として保存するファイルはどれか。

- ア ダンプファイル
- イ チェックポイントファイル
- ウ バックアップファイル
- エ ログファイル

問 24

1.5M ビット／秒の伝送路を用いて 12M バイトのデータを転送するのに必要な伝送時間は何秒か。ここで、伝送路の伝送効率を 50% とする。

- ア 16
- イ 32
- ウ 64
- エ 128

問 25

プライベート IP アドレスをもつ複数の端末が、一つのグローバル IP アドレスを使ってインターネット接続を利用する仕組みを実現するものはどれか。

- ア DHCP
- イ DNS
- ウ NAPT
- エ RADIUS

解説

問 22 関係データベース

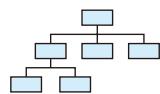
関係データベースは、複数の属性をひと組にして 1 件のデータとし、各属性を列、各データを行とする表の形式にまとめたものです。

表同士は、共通する列の値をキーにして関連付けることができます。

行	列	品名	NO	共通する列をキーに関連付ける	品名	単価
001		A	10		A	1000
002		B	15		B	1500
003		A	20		C	2000
004		C	30			

合格のカギ

②階層型データベース 問22



③ネットワーク型データベース 問22




合格のカギ

- A** インデックス（索引）ファイルの説明です。
- I** 正解です。
- W** 階層型データベースの説明です。
- E** ネットワーク型データベースの説明です。

問23 データベースの更新記録 キホン!
 20%

データベースの更新履歴を記録したファイルを、**ログファイル**または**ジャーナルファイル**といいます。

- A** **ダンプファイル**は、データベースの現在の内容を書き出したものです。
- I** **チェックポイントファイル**は、メモリ上に記憶されている更新内容を、一定期間ごとにディスク上に書き出したものです。システム障害によってメモリ上の内容が失われた場合は、チェックポイントファイルを使って直前のチェックポイントの状態に復元し、さらにログファイルによって障害前の状態に復元します。
- W** **バックアップファイル**は、データベースの内容を別の媒体に保管したものです。
- E** 正解です。

問24 伝送時間の計算 キホン!
 30%

伝送路のスピードは 1.5M ビット／秒ですが、伝送効率が 50% ので、実効スピードは $1.5 \times 0.5 = 0.75$ M ビット／秒になります。12M バイト = 12×8 M ビットなので（1 バイト = 8 ビット）、データを転送するのに必要な時間は、次のように求められます。正解は **E** です。

$$12 \times 8 \div 0.75 = 128 \text{ 秒}$$

問25 プライベート IP アドレスの変換
 30%

LAN に接続されている端末からインターネットを利用するには、各端末のプライベート IP アドレスを、グローバル IP アドレスに変換しなければなりません。この機能を **NAT** (Network Address Translation) といいます。さらに、IP アドレスをポート番号を組み合わせることで、1 つのグローバル IP アドレスだけで、LAN 上の各端末が同時にインターネットを利用できるようにする仕組みを **NAPT** (Network Address Port Translation : IP マスカレード) といいます。

- A** **DHCP** (Dynamic Host Configuration Protocol) は、LAN に接続した端末に IP アドレスを自動的に割り当てるためのプロトコルです。
- I** **DNS** (Domain Name System) は、インターネットのドメイン名と IP アドレスを変換するための仕組みです。
- W** 正解です。
- E** **RADIUS** (Remote Authentication Dial In User Service) は、ネットワークを介してアクセスするサーバ利用者を認証するためのプロトコルです。


チェックポイント 問23

メモリ上に記憶されている更新内容を、ディスク上に記録されているデータベースに反映する時点のこと。チェックポイント後に更新された内容はディスク上に反映されていないため、システム障害によって失われてしまう。最近のチェックポイントから、データベースを最新の状態に復元するために、ログファイル（ジャーナルファイル）を利用する。


プライベートIPアドレス 問25

LAN に接続されている各端末に割り当てる、その LAN 内部で利用するための IP アドレス。


グローバルIPアドレス 問25

インターネット上で一意に割り当てる IP アドレス。


覚えよう！ 問25

NAT といえば

- プライベート IP アドレスとグローバル IP アドレスを相互に変換
- グローバル IP アドレスを LAN 上の複数の端末が共有することで、インターネットの IP アドレス不足を解消できる
- グローバル IP アドレスは同時に接続する数だけ必要

NAPT (IPマスカレード) といえば

- 1 つのグローバル IP アドレスで、複数の端末が同時にインターネットを利用できるように NAT 機能を拡張したもの

○ 解答

- | | | | |
|------------|----------|------------|----------|
| 問22 | I | 問23 | E |
| 問24 | E | 問25 | W |

問 26 LAN 間接続装置に関する記述のうち、適切なものはどれか。

- ア ゲートウェイは、OSI 基本参照モデルにおける第1～3層だけのプロトコルを変換する。
- イ ブリッジは、IP アドレスを基にしてフレームを中継する。
- ウ リピータは、同種のセグメント間で信号を增幅することによって伝送距離を延長する。
- エ ルータは、MAC アドレスを基にしてフレームを中継する。

問 27 次の IP アドレスとサブネットマスクをもつ PC がある。この PC のネットワークアドレスとして、適切なものはどれか。

IP アドレス : 10.170.70.19
サブネットマスク : 255.255.255.240

- ア 10.170.70.0
- イ 10.170.70.16
- ウ 10.170.70.31
- エ 10.170.70.255

問 28 インターネットにおける電子メールの規約で、ヘッダフィールドの拡張を行い、テキストだけでなく、音声、画像なども扱えるようにしたものはどれか。

- ア HTML
- イ MHS
- ウ MIME
- エ SMTP

問 29 SQL インジェクション攻撃の説明はどれか。

- ア Web アプリケーションに問題があるとき、悪意のある問合せや操作を行う命令文を Web サイトに入力して、データベースのデータを不正に取得したり改ざんしたりする攻撃
- イ 悪意のあるスクリプトを埋め込んだ Web ページを訪問者に閲覧させて、別の Web サイトで、その訪問者が意図しない操作を行わせる攻撃
- ウ 市販されている DBMS の脆弱性を悪用することによって、宿主となるデータベースサーバを探して感染を繰り返し、インターネットのトラフィックを急増させる攻撃
- エ 訪問者の入力データをそのまま画面に表示する Web サイトを悪用して、悪意のあるスクリプトを訪問者の Web ブラウザで実行させる攻撃

解説

問 26 LAN 間接続装置 キホン！



OSI 基本参照モデルは、データ通信を 7 つの層に切り分けて、それぞれの層での機能を規定しています。LAN 間接続装置も、OSI 基本参照モデルのどの層の通信を中継するかによって役割が異なります。

- × ア ゲートウェイは、OSI 基本参照モデルの第 1 ～ 7 層すべてのプロトコルを変換します。
- × イ ブリッジは、ネットワークインターフェースカード (NIC) ごとに一意

に割り振られている MAC アドレスを基に、データをフレーム単位で中継する装置です。

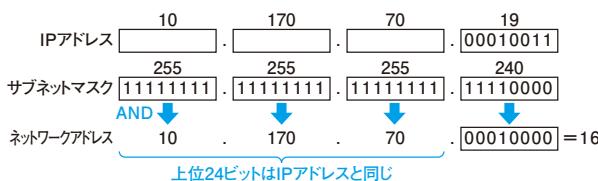
- ウ 正解です。リピータは OSI 基本参照モデルの第 1 層で通信を中継する装置で、データ通信の信号を物理的に増幅して中継します。
- × エ ルータは、IP アドレスを基に、データをパケット単位で中継します。

問27 ネットワークアドレス キポン!

50%

ネットワークアドレスを求めるには、IP アドレスとサブネットマスクを 2 進数に変換し、両者の論理積（AND）を求めます。

なお、サブネットマスク「255.255.255.240」の上位 24 ビット（255.255.255 の部分）は、2 進数ではすべて 1 なので、ネットワークアドレスの上位 24 ビットは IP アドレスと同じになります。したがってこの問題では、下位 8 ビットだけを 2 進数に変換すれば解答できます。



以上から、ネットワークアドレスは「10.170.70.16」です。正解は イ です。

問28 電子メールの規約 キポン!

20%

インターネットの電子メールで、テキスト以外にもさまざまなデータを取り扱うための規格を **MIME** (Multipurpose Internet Mail Extensions) といいます。

- × ア HTML (HyperText Markup Language) は、インターネットの Web ページを記述するためのマークアップ言語です。
- × イ MHS (Message Handling System) は、OSI 基本参照モデルに基づく電子メールサービスの国際規格です。
- ウ 正解です。
- × エ SMTP (Simple Mail Transfer Protocol) は、インターネットの電子メールの送信プロトコルです

問29 SQL インジェクション攻撃

50%

SQL インジェクションとは、Web アプリケーションの入力データに、データベースへの悪意ある命令文を埋め込んで実行させ、データベースを改ざんしたり、情報を不正入手したりする攻撃です。

- ア 正解です。
- × イ クロスサイトリクエストフォージェリ攻撃の説明です。
- × ウ SQL Slammer と呼ばれるコンピュータウイルスの説明です。
- × エ クロスサイトスクリプティング攻撃の説明です。



覚えよう！ 問26

OSI基本参照モデルとLAN間接続装置

OSI基本 参照モデル	LAN間 接続装置
第4～7層 その他	ゲートウェイ
第3層 ネットワーク層	ルータ
第2層 データリンク層	ブリッジ、 レイヤ2スイッチ
第1層 物理層	リピータハブ、 リピータ

ネットワークアドレス 問27

IP アドレスをネットワーク部とホスト部に分け、ホスト部のビットをすべて 0 にしたアドレス。同一のネットワーク内にある IP アドレスは、すべて共通のネットワークアドレスをもつ。

サブネットマスク 問27

IP アドレスのネットワーク部とホスト部の境界を指定するもの。サブネットマスクを 2 進数に変換したとき、1 のビットの範囲がネットワーク部、0 のビットの範囲がホスト部になる。

論理積 (AND) 問27

入力が両方とも 1 のとき 1、それ以外は 0 になる論理演算。

○ 解答 ○

- | | |
|-------|-------|
| 問26 ウ | 問27 イ |
| 問28 ウ | 問29 ア |

 **問 30** SEO ポイズニングの説明はどれか。

- ア** Web 検索サイトの順位付けアルゴリズムを悪用して、検索結果の上位に、悪意のある Web サイトを意図的に表示させる。
- イ** 車などで移動しながら、無線 LAN のアクセスポイントを探して出して、ネットワークに侵入する。
- ウ** ネットワークを流れるパケットから、侵入のパターンに合致するものを検出して、管理者への通知や、検出した内容の記録を行う。
- エ** マルウェア対策ソフトのセキュリティ上の脆弱性を悪用して、システム権限で不正な処理を実行させる。

 **問 31** 暗号方式に関する記述のうち、適切なものはどれか。

- ア** AES は公開鍵暗号方式、RSA は共通鍵暗号方式の一種である。
- イ** 共通鍵暗号方式では、暗号化及び復号に同一の鍵を使用する。
- ウ** 公開鍵暗号方式を通信内容の秘匿に使用する場合は、暗号化に使用する鍵を秘密にして、復号に使用する鍵を公開する。
- エ** デジタル署名に公開鍵暗号方式が使用されることではなく、共通鍵暗号方式が使用される。

 **問 32** 送信者 A からの文書ファイルと、その文書ファイルのデジタル署名を受信者 B が受信したとき、受信者 B ができることはどれか。ここで、受信者 B は送信者 A の署名検証鍵 X を保有しており、受信者 B と第三者は送信者 A の署名生成鍵 Y を知らないものとする。

- ア** デジタル署名、文書ファイル及び署名検証鍵 X を比較することによって、文書ファイルに改ざんがあった場合、その部分を判別できる。
- イ** 文書ファイルが改ざんされていないこと、及びデジタル署名が署名生成鍵 Y によって生成されたことを確認できる。
- ウ** 文書ファイルがマルウェアに感染していないことを認証局に問い合わせて確認できる。
- エ** 文書ファイルとデジタル署名のどちらかが改ざんされた場合、どちらが改ざんされたかを判別できる。

問30 SEO ポイズニング

■ 20%

特定のサイトが、検索サイトの検索結果の上位に表示されるようにサイトの内容を調整することをSEO（検索エンジン最適化）といいます。SEO ポイズニングは、SEO の手法を使って、悪意のあるサイトを検索結果の上位に表示させる攻撃手法です。

- ア 正解です。
- イ ウォードライビングと呼ばれる攻撃手法の説明です。
- ウ IDS（侵入検知システム）の説明です。
- エ 2020年に発見されたマルウェア対策ソフトの脆弱性についての説明です。SEO ポイズニングとは関係ありません。

問31 暗号方式

■ 20%

共通鍵暗号方式は、暗号化と復号に同一の鍵を使用する暗号方式です。共通鍵暗号方式では、第三者に鍵が漏えいすると暗号が解読されてしまうため、鍵を厳重に秘密にしなければなりません。これに対し公開鍵暗号方式は、暗号化用の鍵と復号用の鍵とを分離して、暗号化用の鍵を公開できるようにした暗号方式です。

- ア AESは米国政府によって規格化された標準の共通鍵暗号方式、RSAは大きな数の素因数分解が困難なことを利用した公開鍵暗号方式の一種です。
- イ 正解です。
- ウ 公開鍵暗号方式では、暗号化に使用する鍵を公開して、復号に使用する鍵を秘密にします。
- エ デジタル署名では、公開鍵暗号方式の秘密鍵を署名用の鍵とし、公開鍵を署名検証用の鍵として使用します。

問32 デジタル署名

■ 20%

送信者 A は、文書ファイルから生成したメッセージダイジェスト（ハッシュ値）を、送信者 A の秘密鍵（署名生成鍵 Y）によって暗号化します。これが受信者 B に送信するデジタル署名になります。

受信者 B は、受け取った署名を送信者 A の公開鍵（署名検証鍵 X）によって復号します。復号に成功すれば、その署名が送信者 A の署名生成鍵 Y によって生成されたものであると確認できます。また、復号した署名と文書ファイルから生成したメッセージダイジェストとを照合することで、文書ファイルが改ざんされていないことが確認できます。

- ア 改ざんされた部分を特定することはできません。
- イ 正解です。
- ウ 認証局はマルウェアに感染しているかどうかは確認できません。
- エ 署名が改ざんされた場合は、文書が改ざんされたかどうかを判別できません。

合格のカギ

マルウェア対策ソフトの脆弱性

問30

マルウェア対策ソフトが不正なファイルを削除しようとした瞬間に狙い、そのファイルをシステムファイルへのリンクに置き換えて削除される脆弱性のこと。

覚えよう！

問31

公開鍵暗号方式といえば

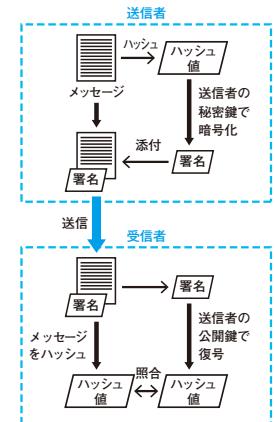
- 受信者の公開鍵で暗号化し、受信者の秘密鍵で復号
 - RSA、楕円曲線暗号など
- 共通鍵暗号方式といえば
- 暗号化用と復号用に共通の鍵を使用
 - DES、AES など

覚えよう！

問32

デジタル署名といえば

- 送信者：メッセージのハッシュ値を送信者の秘密鍵で暗号化
- 受信者：送信者の公開鍵で署名を復号



○ 解答

問30 ア 問31 イ
問32 イ

問 33 ボットネットにおける C&C サーバの役割として、適切なものはどれか。

- ア Web サイトのコンテンツをキャッシュし、本来のサーバに代わってコンテンツを利用者に配信することによって、ネットワークやサーバの負荷を軽減する。
- イ 外部からインターネットを経由して社内ネットワークにアクセスする際に、CHAP などのプロトコルを用いることによって、利用者認証時のパスワードの盗聴を防止する。
- ウ 外部からインターネットを経由して社内ネットワークにアクセスする際に、チャレンジレスポンス方式を採用したワンタイムパスワードを用いることによって、利用者認証時のパスワードの盗聴を防止する。
- エ 侵入して乗っ取ったコンピュータに対して、他のコンピュータへの攻撃などの不正な操作をするよう、外部から命令を出したり応答を受け取ったりする。

問 34 JIS Q 27000:2014（情報セキュリティマネジメントシステム－用語）において、“エンティティは、それが主張するとおりのものであるという特性”と定義されているものはどれか。

- ア 真正性
- イ 信頼性
- ウ 責任追跡性
- エ 否認防止

問 35 WAF の説明はどれか。

- ア Web サイトに対するアクセス内容を監視し、攻撃とみなされるパターンを検知したときに当該アクセスを遮断する。
- イ Wi-Fi アライアンスが認定した無線 LAN の暗号化方式の規格であり、AES 暗号に対応している。
- ウ 様々なシステムの動作ログを一元的に蓄積、管理し、セキュリティ上の脅威となる事象をいち早く検知、分析する。
- エ ファイアウォール機能を有し、ウイルス対策、侵入検知などを連携させ、複数のセキュリティ機能を統合的に管理する。

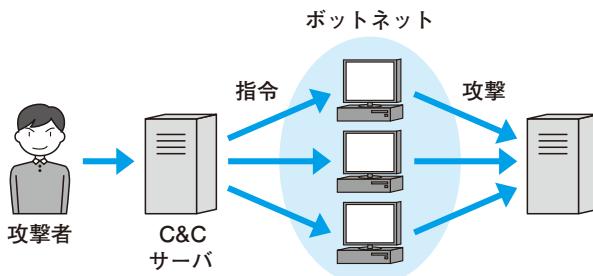
解説

問33 C&C サーバの役割

ウイルスなどによって外部から不正に遠隔操作されたコンピュータで構成されるネットワークを、ボットネットといいます。C&C (Command and



Control) サーバは、支配下にあるコンピュータに対して命令を出したり、応答を受け取ったりするサーバのことです。



- ア リバースプロキシの説明です。
- イ, ウ 社内ネットワークの利用者を識別する認証サーバの説明です。
- エ 正解です。

問34 情報セキュリティの拡張要素

30%

一般に情報セキュリティとは、情報の**機密性**、**完全性**、**可用性**を維持することと定義されますが、JIS Q 27000ではこの3要素に加えて、**真正性**、**責任追跡性**、**否認防止**、**信頼性**の4つ特性の維持を情報セキュリティに含めることもできると定義しています。

このうち、サーバ、利用者、文書などのセキュリティの管理対象（エンティティ）がなりすましや偽物ではなく、「私は〇〇です」と主張しているとおりの本物であることを**真正性**といいます。

- ア 正解です。
- イ **信頼性**とは、システムが正常に稼働して、期待した処理を確実に行っている特性です。
- ウ **責任追跡性**とは、情報に対して操作を行った利用者を特定でき、過去にさかのぼって追跡できる特性です。
- エ **否認防止**とは、情報に対して操作が行われた事実を、後になって否認されないようにする特性です。

問35 WAF

20%

WAF(Webアプリケーションファイアウォール)は、クライアントとWebサイトとの間に設置して、Webサイトに対するアクセス内容を監視し、攻撃とみなされるパターンを検知すると、そのアクセスを遮断するファイアウォールです。

- ア 正解です。
- イ **WPA**(Wi-Fi Protected Access)の説明です。
- ウ **SIEM**(Security Information and Event Management)の説明です。
- エ **UTM**(Unified Threat Management)の説明です。



CHAP

問33

ユーザーIDとパスワードを使う認証方式のひとつ。ダイヤルアップ接続などのシステムで使われていたもので、ユーザーIDとパスワードは暗号化されるため安全性が高いのが特徴。

チャレンジレスポンス

問33

チャレンジとレスポンスと呼ばれる値を使った認証方式。サーバは利用者に毎回異なる値(チャレンジ)を送信し、利用者はチャレンジに自分のパスワードをつなげてハッシュ化した値(レスポンス)を送り返す。レスポンスは毎回異なった値になるので、一種のワンタイムパスワードとなる。

覚えよう！

問34

情報セキュリティの要素 といえば

- **機密性**：機密が保たれること
- **完全性**：情報が正確なこと
- **可用性**：必要なときに利用できること

覚えよう！

問35

WAFといえば

- Webアプリケーションへの通信内容を監視
- SQLインジェクションなどの攻撃を検知して遮断

○ 解答 ○

問33 エ 問34 ア
問35 ア

問 36

社内ネットワークとインターネットの接続点に、ステートフルインスペクション機能をもたない、静的なパケットフィルタリング型のファイアウォールを設置している。このネットワーク構成において、社内の PC からインターネット上の SMTP サーバに電子メールを送信できるようにするとき、ファイアウォールで通過を許可する TCP パケットのポート番号の組合せはどれか。ここで、SMTP 通信には、デフォルトのポート番号を使うものとする。

	送信元	宛先	送信元 ポート番号	宛先 ポート番号
ア	PC	SMTP サーバ	25	1024 以上
	SMTP サーバ	PC	1024 以上	25
イ	PC	SMTP サーバ	110	1024 以上
	SMTP サーバ	PC	1024 以上	110
ウ	PC	SMTP サーバ	1024 以上	25
	SMTP サーバ	PC	25	1024 以上
エ	PC	SMTP サーバ	1024 以上	110
	SMTP サーバ	PC	110	1024 以上

問 37

電子メールをスマートフォンで受信する際のメールサーバとスマートフォン間の通信をメール本文を含めて暗号化するプロトコルはどれか。

ア APOP

イ IMAPS

ウ POP3

エ SMTP Submission

問 38

E-R 図の説明はどれか。

ア オブジェクト指向モデルを表現する図である。

イ 時間や行動などに応じて、状態が変化する状況を表現する図である。

ウ 対象とする世界を実体と関連の二つの概念で表現する図である。

エ データの流れを視覚的に分かりやすく表現する図である。

解説

問36 パケットフィルタリング キホン!

■ 30%

ファイアウォールを通過するパケットの IP アドレスやポート番号を調べて、

合格のカギ

通信の許可／不許可を判断することを**パケットフィルタリング**といいます。

社内のPCからSMTPサーバにアクセスするには、**PCからSMTPサーバへの通信**と、**SMTPサーバからPCへの通信**の両方を許可する必要があります。

SMTPサーバは、標準のポート番号としては25番ポートを使用します。したがって、PCからSMTPサーバへの送信は、宛先ポート番号に**25**を指定します。また、PC側の送信元ポート番号は、電子メールソフトが**1024以上**の番号を適宜割り当てます。

送信元	宛先	送信元 ポート番号	宛先 ポート番号
PC	SMTPサーバ	1024以上	25

一方、SMTPサーバからPCへの送信では、SMTPサーバが送信元になるので、送信元ポート番号に**25**を指定します。

送信元	宛先	送信元 ポート番号	宛先 ポート番号
SMTPサーバ	PC	25	1024以上

以上から、ポート番号の組合せは、**ウ**です。

問37 電子メールの暗号化プロトコル

■ 20%

メールアプリとメールサーバ間の通信プロトコルは、メール送信用のSMTP、メール受信用のPOP3またはIMAP4が標準的です。これらのプロトコルに伝送路を暗号化するSSL/TLSを組み合わせ、通信を暗号化したものを作成します。それでは**SMTPS**、**POPS**、**IMAPS**といいます。

- × **A** **APOP**は、POP3でメールサーバとやり取りするパスワードを暗号化するプロトコルです。
- **イ** 正解です。
- × **ウ** **POP3**はメールサーバからメールを受信するためのプロトコルです。
- × **エ** **SMTP Submission**は、利用者がSMTPでメール送信する際に、一般的な25番ポートではなくサブミッションポートと呼ばれるポート番号を使い、利用者認証を行う仕組みです。

合格のカギ

ステートフルインスペクション

問36

ファイアウォールがパケットの内容を検査して、パケットを通過させるかどうかを動的に判断する機能。サーバとクライアントのやり取りで、問合せに対する応答を自動的に通過させるといった機能がある。

問38 E-R図 キホン!

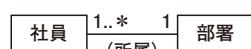
■ 20%

E-R図は、対象とする世界を**実体**（エンティティ）と**実体間の関連**（リレーションシップ）によってモデル化した図です。データベースの構造設計などに用いられています。

- × **A** UML（統一モデリング言語）などの説明です。
- × **イ** 状態遷移図の説明です。
- **ウ** 正解です。
- × **エ** DFD（データフローダイアグラム）の説明です。

E-R図の例

問38



1つの部署には複数の社員が所属し、各社員は1つの部署に所属することを示す。

○ 解答

- | | | | |
|-----|---|-----|---|
| 問36 | ウ | 問37 | イ |
| 問38 | ウ | | |

問 39

オブジェクト指向におけるクラスとインスタンスとの関係のうち、適切なものはどれか。

- ア** インスタンスはクラスの仕様を定義したものである。
- イ** クラスの定義に基づいてインスタンスが生成される。
- ウ** 一つのインスタンスに対して、複数のクラスが対応する。
- エ** 一つのクラスに対して、インスタンスはただ一つ存在する。

問 40

モジュール間の情報の受渡しがパラメタだけで行われる、結合度が最も弱いモジュール結合はどれか。

- ア** 共通結合
- イ** 制御結合
- ウ** データ結合
- エ** 内容結合

問 41

ブラックボックステストに関する記述のうち、適切なものはどれか。

- ア** テストデータの作成基準として、命令や分岐の網羅率を使用する。
- イ** 被テストプログラムに冗長なコードがあっても検出できない。
- ウ** プログラムの内部構造に着目し、必要な部分が実行されたかどうかを検証する。
- エ** 分岐命令やモジュールの数が増えると、テストデータが急増する。

問 42

ソフトウェア開発の活動のうち、アジャイル開発においても重視されているリファクタリングはどれか。

- ア** ソフトウェアの品質を高めるために、2人のプログラマが協力して、一つのプログラムをコーディングする。
- イ** ソフトウェアの保守性を高めるために、外部仕様を変更することなく、プログラムの内部構造を変更する。
- ウ** 動作するソフトウェアを迅速に開発するために、テストケースを先に設定してから、プログラムをコーディングする。
- エ** 利用者からのフィードバックを得るために、提供予定のソフトウェアの試作品を早期に作成する。

解説

問39 クラスとインスタンスの関係 キホン!



オブジェクトが備える属性や振る舞い（メソッド）を定義したものが**クラス**です。また、クラスの定義に基づいて生成されるオブジェクトを**インスタンス**



といいます。クラスを設計図とすれば、インスタンスは設計図に基づいて作られた製品に当たります。

- ア インスタンスの仕様を定義したものがクラスです。
- イ 正解です。
- ウ 原則として、1つのインスタンスには1つのクラスが対応します。
- エ 1つのクラスには複数のインスタンスが存在できます。

問40 モジュール結合度 キホン!

■■■ 40%

モジュール同士の関連性の度合いをモジュール結合度といいます。一般に、モジュール結合度が弱いほど個々のモジュールの独立性が高く、保守しやすくなる特徴があります。

モジュール結合度の種類

強	
内容結合	一方のモジュールが、他のモジュール内部を直接参照したり、他のモジュール中に分岐する。
共通結合	複数のモジュールが共通のデータ領域を共有している。
外部結合	モジュール同士が同じ大域的な変数を参照している。
制御結合	一方のモジュールが、他のモジュールの内部ロジックをパラメタで制御する。
スタンプ結合	モジュール同士が同じ非大域的な変数を参照している。
データ結合	一方のモジュールが、必要なデータ項目だけをパラメタとして他のモジュールに渡す。
弱	

以上から、モジュール結合度が最も弱いのは、ウのデータ結合です。

問41 ブラックボックステスト

■■■ 30%

ブラックボックステストは、プログラムが仕様どおりに動作するかどうかを、入力に対する出力に着目してテストします。出力が正しければ内部構造については考慮しないので、冗長なコードかどうかは検出できません。以上から、正解はイです。

ブラックボックステストに対して、プログラム内部のロジックが正しいかどうかを検証するテストを、ホワイトボックステストといいます。ア、ウ、エはいずれもホワイトボックステストに関する記述です。

問42 リファクタリング

■■■ 20%

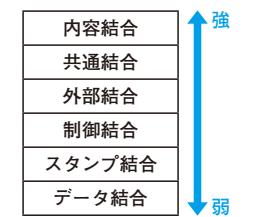
リファクタリングとは、外側から見える仕様は変更せずに、プログラムの内部構造を変更するソフトウェア開発手法です。主に、処理速度を高速化したり、ソフトウェアの保守性を高めるために行います。

- ア ペアプログラミングの説明です。
- イ 正解です。
- ウ テスト駆動開発の説明です。
- エ プロトタイピングの説明です。

合格のカギ

オブジェクト指向 問39
オブジェクトと呼ばれる構成部品を基に、プログラムを開発していく開発手法のこと。

モジュール結合度 問40
モジュール同士の関係の度合いを表す。結合度が弱いほどモジュールの独立性が高い。



覚えよう！ 問41

ブラックボックステスト といえば

- プログラムの外部仕様に基づいてテスト
- 同値分割法、限界値分析などの方式がある

ホワイトボックステスト といえば

- プログラムの内部構造に基づいてテスト
- 条件網羅、条件判定網羅、分岐網羅などの方式がある

アジャイル開発 問42

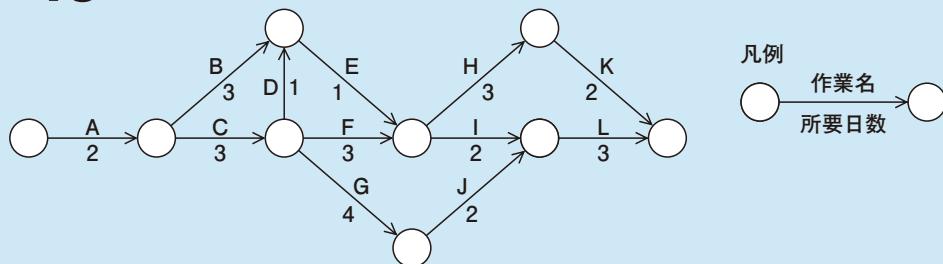
開発対象を小さな機能に分割し、短期間に1つの機能を開発する工程を反復しながら、全体を開発していく開発手法。エクストリームプログラミング(XP)、スクラムなどの種類がある。

○ 解答 ○

- | | | | |
|-----|---|-----|---|
| 問39 | イ | 問40 | ウ |
| 問41 | イ | 問42 | イ |

問 43

図に示すプロジェクト活動のクリティカルパスはどれか。



- ア A → B → E → I → L
ウ A → C → F → I → L

- イ A → C → D → E → H → K
エ A → C → G → J → L

問 44

あるソフトウェアにおいて、機能の個数と機能の複雑度に対する重み付け係数は表のとおりである。このソフトウェアのファンクションポイント値は幾らか。ここで、ソフトウェアの全体的な複雑さの補正係数は 0.75 とする。

ユーザファンクションタイプ	個数	重み付け係数
外部入力	1	4
外部出力	2	5
内部論理ファイル	1	10

ア 18

イ 24

ウ 30

エ 32

問 45

次の条件で IT サービスを提供している。SLA を満たすことができる、1か月のサービス時間帯中の停止時間は最大何時間か。ここで、1か月の営業日数は 30 日とし、サービス時間帯中は、保守などのサービス計画停止は行わないものとする。

[SLA の条件]

- ・サービス時間帯は、営業日の午前 8 時から午後 10 時までとする。
- ・可用性を 99.5%以上とする。

ア 0.3

イ 2.1

ウ 3.0

エ 3.6



解説

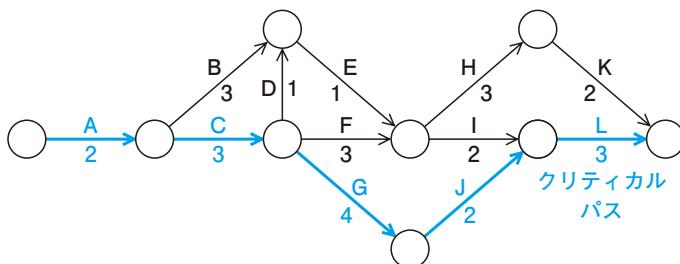
問43 クリティカルパス

■■■ 80%

開始から終了までの各作業のうち、時間的余裕のない一連の作業経路を、**クリティカルパス**といいます。クリティカルパス上にある作業が1日でも遅れるとき、プロジェクト全体の遅れにつながります。

アローダイアグラムでは、開始から終了に至る複数の作業経路のうち、合計の所要日数が最も多くなる経路がクリティカルパスになります。したがって、**A～E**のうち、所要日数の合計が最も多い経路が正解です。

- × **A** 2日 + 3日 + 1日 + 2日 + 3日 = 11日
- × **イ** 2日 + 3日 + 1日 + 1日 + 3日 + 2日 = 12日
- × **ウ** 2日 + 3日 + 3日 + 2日 + 3日 = 13日
- **エ** 2日 + 3日 + 4日 + 2日 + 3日 = 14日



問44 ファンクションポイント法 キポン!

■■■ 60%

システムの機能の個数に、難易度による重み付けをして点数を求め、システム開発のコストや規模を見積もる手法を**ファンクションポイント法**といいます。

問題文の表から、点数の合計は次のように求められます。

	個数	重み	点数	
外部入力	1	×	4	= 4
外部出力	2	×	5	= 10
内部論理ファイル	1	×	10	= 10 } 合計：24

ファンクションポイント値は、合計点数（未調整FP）に、複雑さの補正係数0.75を掛け、 $24 \times 0.75 = 18$ になります。正解は**ア**です。

問45 SLA キポン!

■■■ 20%

可用性（アベイラビリティ）とは、サービスがいつでも利用できる状態にあることをいいます。可用性の大きさは稼働率で表します。稼働率は、全サービス時間中、サービスが停止せずに利用可能な時間の割合です。

1日のサービス時間は午前8時から午後10時までの14時間、1か月の営業日数は30日なので、1か月のサービス時間は $14 \times 30 = 420$ 時間です。可用性を99.5%以上とするには、停止時間を全サービス時間の $100 - 99.5 = 0.5\%$ 未満におさえなければなりません。したがって、停止時間は最大で $420 \times 0.005 = 2.1$ 時間となります。正解は**イ**です。



アローダイアグラム 問43

プロジェクトを達成するのに必要な作業を矢線で、作業の結合点を○で表した図。所要日数を示して日程計画を立てるために用いる。

覚えよう！ 問44

ファンクションポイント法
といえば

- 入出力、画面、ファイルなどの機能の個数によって開発規模を見積もる
- 個々の機能を開発の難易度に応じて点数化する

SLA 問45

Service Level Agreement : サービス品質保証契約。ITサービスの利用者と提供者とが、サービスの品質について取り交わす合意のこと。

○ 解答 ○

- 問43 **エ** 問44 **ア**
問45 **イ**

問 46 システムの開発部門と運用部門が別々に組織化されているとき、システム開発を伴う新規サービスの設計及び移行を円滑かつ効果的に進めるための方法のうち、適切なものはどれか。

- ア** 運用テストの完了後に、開発部門がシステム仕様と運用方法を運用部門に説明する。
- イ** 運用テストは、開発部門の支援を受けずに、運用部門だけで実施する。
- ウ** 運用部門からもシステムの運用に関わる要件の抽出に積極的に参加する。
- エ** 開発部門は運用テストを実施して、運用マニュアルを作成し、運用部門に引き渡す。

問 47 システム監査実施体制のうち、システム監査人の独立性の観点から最も避けるべきものはどれか。

- ア** 監査チームメンバーに任命された総務部の A さんが、他のメンバーと一緒に、総務部の入退室管理の状況を監査する。
- イ** 監査部の B さんが、個人情報を取り扱う業務を委託している外部企業の個人情報管理状況を監査する。
- ウ** 情報システム部の開発管理者から 5 年前に監査部に異動した C さんが、マーケティング部におけるインターネットの利用状況を監査する。
- エ** 法務部の D さんが、監査部からの依頼によって、外部委託契約の妥当性の監査において、監査人に協力する。

問 48 事業継続計画（BCP）について監査を実施した結果、適切な状況と判断されるものはどれか。

- ア** 従業員の緊急連絡先リストを作成し、最新版に更新している。
- イ** 重要書類は複製せずに 1 か所で集中保管している。
- ウ** 全ての業務について、優先順位なしに同一水準の BCP を策定している。
- エ** 平時には BCP を従業員に非公開としている。

問 49 エンタープライズアーキテクチャの“四つの分類体系”に含まれるアーキテクチャは、ビジネスアーキテクチャ、テクノロジアーキテクチャ、アプリケーションアーキテクチャともう一つはどれか。

- | | |
|----------------------|------------------------|
| ア システムアーキテクチャ | イ ソフトウェアアーキテクチャ |
| ウ データアーキテクチャ | エ バスアーキテクチャ |

解説

問46 開発部門と運用部門

■ 20%



システム化する業務の内容に精通し、実際にシステムの運用を担当するのは運用部門です。したがって、運用部門がシステムの要件抽出の段階から参加することは、移行を円滑にし、効果的なシステムを開発するために重要です。

- × **A** 運用テストは運用部門が主体となって実施するので、仕様などは運用テスト前に説明すべきです。
- × **I** 運用テストの主体は運用部門ですが、開発部門の支援も必要です。
- **ウ** 正解です。
- × **エ** 運用テストは運用部門が主体となって実施します。

問47 システム監査人の独立性

■ 30%

システム監査人は、独立・公正な立場で監査を実施するため、被監査主体（監査される側）と身分上密接な利害関係をもたないようにしなければなりません。これをシステム監査人の独立性といいます。

- **A** 総務部に所属する人が総務部を監査するので、問題があります。
- × **イ** 監査部は委託先企業から独立しているので、問題ありません。
- × **ウ** 監査対象はマーケティング部であり、Cさんが以前に所属していた部署ではないので、問題ありません。
- × **エ** 監査に協力するだけなので、監査人の独立性とは関係ありません。

問48 事業継続計画（BCP）の監査

■ 20%

事業継続計画（BCP）とは、災害など不測の事態が発生した場合に、必要な業務を早期に復旧・再開できるように立てておく計画のことです。

- **ア** 正解です。緊急連絡先リストは緊急時に利用できるよう、常に最新のものに更新しておかなければなりません。
- × **イ** 重要書類が複製も分散管理もせずに保管されていると、災害時にすべて失われ、復元できない危険があります。
- × **ウ** 災害時にすべての業務を継続するのは困難な場合もあるので、重要度に応じて優先順位をつけるべきです。
- × **エ** BCPは普段から従業員に周知しておかなければ、いざというとき計画通りに行動できません。

問49 エンタープライズアーキテクチャ

■ 40%

エンタープライズアーキテクチャ（EA）とは、組織全体の業務と情報システムを統一的にモデル化し、組織の最適化を図る手法です。エンタープライズアーキテクチャのモデル化には、①**ビジネスアーキテクチャ**（業務機能）、②**データアーキテクチャ**（業務機能に使われる情報）、③**アプリケーションアーキテクチャ**（業務機能を実現するサービス）、④**テクノロジアーキテクチャ**（各サービスを実現するための技術）の4つの分類体系があります。正解は**ウ**です。



問47

監査人の独立性といえば

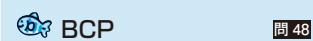
- 監査人が、監査される側と利害関係をもたないこと



問47

監査人の独立性の観点から避けるべきことといえば

- ①自分が所属している部署の業務を監査する。
 - ②監査人の上司を監査する。
 - ③納入したシステムを納入業者自身が監査する。
- といったケースは、システム監査人の独立性の観点から避けるべきである。



問48

Business Continuity Planの略。



問49

エンタープライズアーキテクチャといえば

ビジネスアーキテクチャ
データアーキテクチャ
アプリケーションアーキテクチャ
テクノロジアーキテクチャ

○ 解答 ○

- | | | | |
|-----|---|-----|---|
| 問46 | ウ | 問47 | ア |
| 問48 | ア | 問49 | ウ |

問 50 BPO を説明したものはどれか。

- ア** 自社ではサーバを所有せずに、通信事業者などが保有するサーバの処理能力や記憶容量の一部を借りてシステムを運用することである。
- イ** 自社ではソフトウェアを所有せずに、外部の専門業者が提供するソフトウェアの機能をネットワーク経由で活用することである。
- ウ** 自社の管理部門やコールセンタなど特定部門の業務プロセス全般を、業務システムの運用などと一緒に外部の専門業者に委託することである。
- エ** 自社よりも人件費が安い派遣会社の社員を活用することによって、ソフトウェア開発の費用を低減させることである。

問 51 SaaS を説明したものはどれか。

- ア** インターネット経由でアプリケーションソフトウェアの機能を、利用者が必要なときだけ利用するサービスのこと
- イ** 企業の経営資源を有効に活用するために、基幹業務を統合的に管理するためのソフトウェアパッケージのこと
- ウ** 既存の組織やビジネスプロセスを抜本的に見直し、職務、業務フロー、管理機構、情報システムを再設計すること
- エ** 発注者とサービス提供者との間で、サービスの品質の内容について合意した文書のこと

問 52 コアコンピタンスの説明はどれか。

- ア** 企業の活動分野において、競合他社にはまねのできない卓越した能力
- イ** 経営を行う上で法令や各種規制、社会的規範などを遵守する企業活動
- ウ** 市場・技術・商品（サービス）の観点から設定した、事業の展開領域
- エ** 組織活動の目的を達成するために行う、業務とシステムの全体最適化手法

問 53 プロダクトライフサイクルにおける成長期の特徴はどれか。

- ア** 市場が製品の価値を理解し始める。製品ラインもチャネルも拡大しなければならない。この時期は売上も伸びるが、投資も必要である。
- イ** 需要が大きくなり、製品の差別化や市場の細分化が明確になってくる。競争者間の競争も激化し、新品種の追加やコストダウンが重要となる。
- ウ** 需要が減ってきて、撤退する企業も出てくる。この時期の強者になれるかどうかを判断し、代替市場への進出なども考える。
- エ** 需要は部分的で、新規需要開拓が勝負である。特定ターゲットに対する信念に満ちた説得が必要である。

解説

問50 BPO

■■■ 20%

BPO (Business Process Outsourcing) とは、自社の業務プロセスを、情報システムの運用業務ごと外部業者に委託（アウトソーシング）することです。

- ア ホスティングサービス（レンタルサーバ）の説明です。
- イ ASP（Application Service Provider）の説明です。
- ウ 正解です。
- エ 人材派遣サービスの説明です。

合格のカギ

④ アウトソーシング 問50
自社の業務の一部を、外部の専門業者に委託すること。

⑤ ASP 問50
汎用的なアプリケーションシステムの機能をネットワーク経由で複数の顧客に提供するサービス。

問51 SaaS キホン!

■■■ 20%

SaaS (Software as a Service) とは、サービスプロバイダが提供するアプリケーションソフトウェアの機能を、利用者が必要なときにインターネット経由で利用するサービスです。必要な機能を利用したいとき、従来のようにソフトウェアをPCにインストールするのではなく、オンラインサービスとして利用できるのが特徴です。

- ア 正解です。
- イ ERPパッケージの説明です。
- ウ BPR（Business Process Reengineering）の説明です。
- エ SLA（Service Level Agreement）の説明です。

覚えよう！ 問51

SaaSといえば

- ・ソフトウェアの機能をサービスとして提供
- ・利用者はインターネットを介して必要な機能だけを利用

覚えよう！ 問52

コアコンピタンス

といえば

- ・他社にまねのできない経営資源（技術やノウハウ）

問52 コアコンピタンス キホン!

■■■ 20%

コアコンピタンスとは、競合する他社がまねできない、自社独自の技術やノウハウのことです。コアコンピタンスを業務の中核に置いた経営を、コアコンピタンス経営といいます。

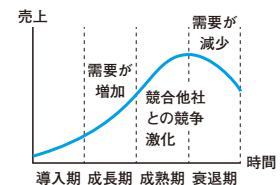
- ア 正解です。
- イ コンプライアンス（法令遵守）の説明です。
- ウ 事業ドメインの説明です。
- エ エンタープライズアーキテクチャの説明です。

問52

参考 「コア」は核、「コンピタンス」は能力の意味だね。

覚えよう！ 問53プロダクトライフサイクル
といえば

- ・導入期→成長期→成熟期→衰退期



問53 プロダクトライフサイクル

■■■ 30%

プロダクトライフサイクルとは、製品の売上の変遷を、①導入期→②成長期→③成熟期→④衰退期という4つの段階にモデル化したものです。成長期は製品が市場に認知され、売上が伸びていく段階です。

- ア 正解です。
- イ 成熟期の説明です。
- ウ 衰退期の説明です。
- エ 導入期の説明です。

○ 解答 ○

- | | | | |
|-----|---|-----|---|
| 問50 | ウ | 問51 | ア |
| 問52 | ア | 問53 | ア |

問 54 政府は、IoTを始めとする様々なICTが最大限に活用され、サイバー空間とフィジカル空間とが融合された“超スマート社会”的実現を推進してきた。必要なものやサービスが人々に過不足なく提供され、年齢や性別などの違いにかかわらず、誰もが快適に生活することができるとされる“超スマート社会”実現への取組は何と呼ばれているか。

ア e-Gov イ Society 5.0 ウ Web 2.0 エ ダイバーシティ社会

問 55 セル生産方式の利点が生かせる対象はどれか。

- ア 生産性を上げるために、大量生産が必要なもの
- イ 製品の仕様が長期間変わらないもの
- ウ 多種類かつフレキシブルな生産が求められるもの
- エ 標準化、単純化による分業が必要なもの

問 56 マトリックス組織を説明したものはどれか。

- ア 業務遂行に必要な機能と利益責任を、製品別、顧客別又は地域別にもつことによって、自己完結的な経営活動が展開できる組織である。
- イ 構成員が、自己の専門とする職能部門と特定の事業を遂行する部門の両方に所属する組織である。
- ウ 購買・生産・販売・財務など、仕事の専門性によって機能分化された部門をもつ組織である。
- エ 特定の課題の下に各部門から専門家を集めて編成し、期間と目標を定めて活動する一時的かつ柔軟な組織である。

問 57 不良品の個数を製品別に集計すると表のようになつた。ABC分析を行つて、まずA群の製品に対策を講じることにした。A群の製品は何種類か。ここで、A群は70%以上とする。

製品	P	Q	R	S	T	U	V	W	X	合計
個数	182	136	120	98	91	83	70	60	35	875

ア 3 イ 4 ウ 5 エ 6

解説

問54 超スマート社会の実現

サイバー空間（仮想空間）とフィジカル空間（現実空間）を高度に融合させたシステムにより、経済発展と社会的課題の解決を両立させた未来の社会とし

合格のカギ

20%

て、日本政府が提唱する社会は **Society 5.0** です。

- ア **e-Gov** (イーガブ) とは、各府省のオンライン申請・届出などの窓口サービスや、行政情報の検索・案内サービスを提供するポータルサイトです。
- イ 正解です。
- ウ **Web 2.0** とは、利用者が情報を受け取るだけでなく、情報発信も可能なウェブのことです。代表的なサービスにブログや SNS があります。
- エ **ダイバーシティ社会** とは、多様な背景や価値観をもつ人々を包含し受容する社会のことです。

合格のカギ

Society 5.0

問54

IoTなどの技術を駆使してサイバー空間とフィジカル空間とが融合した社会。狩猟社会(1.0)、農耕社会(2.0)、工業社会(3.0)、情報社会(4.0)に続く新たな社会なので Society 5.0 という。

ライン生産方式

問55

单一の工程を担当する作業員をライン上に配置して、流れ作業で1種類の製品を生産する方式。

問55 セル生産方式

■ 20%

セル生産方式 は、セルと呼ばれる区分けされた作業台の上で、1人ないし数人のチームがすべての工程を担当する生産方式です。大量生産には向いていませんが、カスタマイズにきめ細かく対応できるので、多品種少量生産や仕様が短期間で変わる製品の生産に向いています。

- ア 大量生産には **ライン生産方式** が向いています。
- イ 仕様が長期間変化しないものは **ライン生産方式** が向いています。
- ウ 正解です。
- エ 分業が必要な製品の生産には、**ライン生産方式** が向いています。

問56 マトリックス組織

キホン!

■ 20%

マトリックス組織 とは、構成員が職能別の部門と特定の事業を遂行する部門の両方に所属する組織形態です。縦割りの組織と横断的な組織が格子（マトリックス）状に交差することから、マトリックス組織といいます。

- ア **事業部制組織** の説明です。
- イ 正解です。
- ウ **職能別組織** の説明です。
- エ **プロジェクト組織** の説明です。

問57 ABC 分析

キホン!

■ 20%

ABC 分析 は、製品群を売上などの大きい順に A, B, C の 3 グループに分類し、重点的に管理する製品などを決定する分析手法です。

不良品の合計は 875 個です。このうちの 70% 以上を A 群とするので、A 群の個数は $875 \times 0.7 = 612.5$ 個となります。不良品の個数が多い順に個数を足していくと、P, Q, R, S の 4 製品の合計が $182 + 136 + 120 + 98 = 536$ 個、P, Q, R, S, T の 5 製品の合計が $536 + 91 = 627$ 個となり、5 製品で 612.5 個を超え、70% 以上となることがわかります。

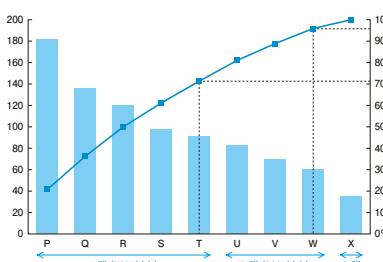
以上から、A 群の製品は 5 種類です。正解は **ウ** です。

覚えよう！

問57

ABC 分析といえ

- 商品全体を、全売上高の 70% を占める A グループ、残り 20% を占める B グループ ($A + B = 90\%$)、残り 10% の C グループに分類
- 分析にはパレート図が使われる
- 主力商品を重点的に管理できる



○ 解答

- | | | | |
|-----|---|-----|---|
| 問54 | イ | 問55 | ウ |
| 問56 | イ | 問57 | ウ |

問 58

表から、期末在庫品を先入先出法で評価した場合の期末の在庫評価額は何千円か。

摘要		数量(個)	単価(千円)
期首在庫		10	10
仕入	4月	1	11
	6月	2	12
	7月	3	13
	9月	4	14
期末在庫		12	

ア 132

イ 138

ウ 150

エ 168

問 59

キャッシュフロー計算書において、営業活動によるキャッシュフローに該当するものはどれか。

- ア** 株式の発行による収入
- イ** 商品の仕入による支出
- ウ** 短期借入金の返済による支出
- エ** 有形固定資産の売却による収入

問 60

A社は、B社と著作物の権利に関する特段の取決めをせず、A社の要求仕様に基づいて、販売管理システムのプログラム作成をB社に委託した。この場合のプログラム著作権の原始的帰属はどれか。

- ア** A社とB社が話し合って決定する。
- イ** A社とB社の共有となる。
- ウ** A社に帰属する。
- エ** B社に帰属する。



解説

問58 先入先出法 キホン!

■ 40%



先入先出法は、「先に仕入れたものから先に出庫する」と考えて在庫評価額を計算する方法です。

期首在庫は10個、期間中の仕入は $1+2+3+4=10$ 個なので、入庫数量は全部で20個です。一方、期末在庫は12個なので、期間中に $20-12=8$ 個を出庫したことになります。この8個は、仕入れが最も早い期首在庫10個の中から出庫されたとみなします（残りの期首在庫は $10-8=2$ 個）。

期末在庫の構成は次のようにになります。

	数量	単価	数量	在庫評価額
期首在庫	2個	10 × 2 =	20	千円
4月仕入分	1個	11 × 1 =	11	千円
6月仕入分	2個	12 × 2 =	24	千円
7月仕入分	3個	13 × 3 =	39	千円
9月仕入分	4個	14 × 4 =	56	千円
合計	12個		150	千円

以上から、在庫評価額は150千円です。正解はウです。

問59 キャッシュフロー計算書

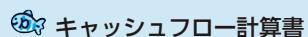
■ 20%



問59

キャッシュフロー計算書は、ある会計期間におけるキャッシュ（現金・預金）の増減を、①営業活動によるキャッシュフロー、②投資活動によるキャッシュフロー、③財務活動によるキャッシュフローの3項目に分類して表したもので、このうち、営業活動によるキャッシュフローは、本業で得た収入から支出を差し引いたもので、選択肢の中では「商品の仕入による支出」がこれに該当します。

- × ア 株式発行による収入は、財務活動によるキャッシュフローに該当します。
- イ 正解です。
- × ウ 短期借入金の返済による支出は、財務活動によるキャッシュフローに該当します。
- × エ 有形固定資産の売却による収入は、投資活動によるキャッシュフローに該当します。



問60

問60 プログラム著作権 キホン!

■ 20%



問60

コンピュータのプログラムは、**著作権法**で保護される著作物です。著作権の帰属について特段の取り決めをせず、A社がB社にプログラム作成を委託した場合、プログラムの著作権はA社ではなく、開発を行ったB社に帰属します。正解はエです。

著作権法で保護される著作物

- プログラム、マニュアル、データベース→保護される
- プログラム言語、規約、アルゴリズム→保護されない

○ 解答 ○

問58 ウ 問59 イ
問60 エ

基本情報技術者試験 予想問題①

科目B



問

01

次のプログラム中の [] に入る正しい答えを、解答群の中から選べ。

ある施設の入場料は、0歳から3歳までは100円、4歳から9歳までは300円、10歳以上は500円である。関数 fee は、年齢を表す0以上の整数を引数として受け取り、入場料を返す。

[プログラム]

○整数型 : fee(整数型 : age)

整数型 : ret

if (age が 3 以下)

 ret ← 100

elseif ([])

 ret ← 300

else

 ret ← 500

endif

return ret

解答群

ア (age が 4 以上) and (age が 9 より小さい)

イ (age が 4 と等しい) or (age が 9 と等しい)

ウ (age が 4 より大きい) and (age が 9 以下)

エ age が 4 以上

オ age が 4 より大きい

カ age が 9 以下

キ age が 9 より小さい

※サンプル問題より

プログラムの解説

```

01 ○整数型 : fee( 整数型 : age ) <-- 関数の宣言
02   整数型 : ret <-- 变数の宣言
03   if (age が 3 以下)
04     ret ← 100
05   elseif ( [ ] )
06     ret ← 300
07   else
08     ret ← 500
09   endif
10   return ret <-- return文

```

if ~ elseif ~ else 構文
については合格のカギを参照。

01 | 関数の宣言

先頭に○記号のついた行は、**関数**や**手続**の宣言です。情報処理試験の擬似言語では、関数を次のように定義します。

○ 戻り値の型名 : 関数名 (引数の型名 : 引数名, …)
関数の処理

キーワード	説明
戻り値の型名	関数の戻り値のデータ型を指定（本問では整数型）。
関数名	関数の名前（本問では fee）。
引数の型名	関数に指定する引数の型名（本問では整数型）。
引数名	関数に指定する引数（本問では age）。引数は複数指定できる。また、引数がない場合は省略できる。

02 | 变数の宣言

関数の処理の中で使用する**变数**は、

型名 : 变数名

の形式で宣言します。本問では ret という整数型の変数を 1 つだけ宣言しています。

10 | return文

return 文は戻り値を関数の呼び出し元のプログラムに返し、関数の処理を終了します。本問では、変数 ret の値を戻り値として返します。

return 戻り値

このプログラムで定義された関数 fee を使用するには、別のプログラムで

sum ← fee(8) + fee(12)

のように関数 fee を呼び出します（8 歳の入場料と 12 歳の入場料の合計を、変数 sum に代入する例）。

問題解説

行番号 03, 04 : 条件式「age が 3 以下」が真のとき、変数 ret に数値 100 を代入します。この処理は、施設の入場料が「0歳から3歳までは100円」であることに対応します。

```
03 if (age が 3 以下)
04   ret ← 100
```

行番号 05, 06 : 空欄の条件式が真のとき、変数 ret に数値 300 を代入します。この処理は、施設の入場料が「4歳から9歳までは300円」であることに対応します。したがって空欄に入る条件式は、年齢が4歳以上9歳以下であることを表すものになるはずです。たとえば、次のような条件式が考えられます。

(age が 4 以上) and (age が 9 以下) ← この条件式は解答群のアに近いですが、よくみると2つ目の条件式が違います。

ただし、この条件式が評価されるのは、行番号 03 の条件式「age が 3 以下」が真でなかったとき (=age が 4 以上のとき) に限られます。つまり、「age が 4 以上」であることは明らかなので、条件式としては「age が 9 以下」のみでじゅうぶんです。

```
05 elseif (age が 9 以下) ← 空欄
06   ret ← 300
```

以上から、本問は力が正解となります。

行番号 07, 08 : 3行目、5行目の条件式がいずれも真でなかったとき、すなわち age の値が 10 以上のとき、変数 ret に数値 500 を代入します。この処理は、施設の入場料が「10歳以上は500円」であることに対応します。

```
07 else
08   ret ← 500
```

○ 解答 ○
問01 力



合格のカギ

if～elseif～else 構文

if～elseif～else この構文では、条件式を上から順番に評価し、最初に真になった条件式に対応する処理を実行します。真になった条件式以降の条件式は評価せず、対応する処理も実行しません。条件式がどれも真にならなかったときは、else に対応する処理 n+1 を実行します。

elseif と処理の組は複数記述することができ、省略することもできます。また、else と処理の組は最後に 1 つだけ記述することができ、省略することもできます。

```
if [条件式 1]
  処理 1
elseif [条件式 2]
  処理 2
  :
elseif [条件式 n]
  処理 n
else
  処理 n+1
endif
```

問 02

次のプログラム中の **a** と **b** に入る正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素は 1 から始まる。

次のプログラムは、整数型の配列 array の要素を 1 つずつ左にずらし、先頭の要素を配列の末尾に移動するプログラムである。

[プログラム]

```
整数型の配列 : array ← {1, 2, 3, 4, 5}  
整数型 : i, tmp  
  
tmp ← array[1]  
for (i を 1 から a まで 1 ずつ増やす)  
    b  
endfor  
array[array の要素数] ← tmp
```

解答群

	a	b
ア	array の要素数	array[i] ← array[i + 1]
イ	array の要素数	array[i + 1] ← array[i]
ウ	(array の要素数 -1)	array[i] ← array[i + 1]
エ	(array の要素数 -1)	array[i + 1] ← array[i]

※オリジナル問題

プログラムの解説

```
01 整数型の配列 : array ← {1, 2, 3, 4, 5} ← 配列の宣言と初期化
02 整数型 : i, tmp

03 tmp ← array[1] ← 配列要素の参照
04 for (i を 1 から  まで 1 ずつ増やす) ← for 構文
05     
06 endfor
07 array[array の要素数] ← tmp ← 配列への代入
```

01 | 配列の宣言と初期化

配列とは、複数の値をまとめて格納しておく値の入れ物です。1つの配列には、同じデータ型の値しか格納できません。

配列の宣言では、格納する値のデータ型と配列名を次のように記述します。

データ型 の配列 : **配列名**

行番号 01 では、整数型の値を格納する **array** という名前の配列を宣言しています。
また、行番号 01 では、配列の宣言と同時に値の代入も行っています。配列の内容は、

{ 値 1, 値 2, 値 3, ...}

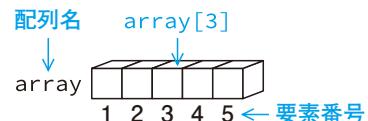
のように、{ } 内に値をカンマで区切って記述します。

03 | 配列要素の参照

行番号 03 では、配列 **array** の先頭の要素 **array[1]** の内容を、変数 **tmp** に格納しています。
配列の中の要素を指定するには、

配列名 [要素番号]

のように、配列名の後に要素番号を [] で囲んで記述します。



要素番号は、実際のプログラム言語では 0 から始まるのが一般的ですが、**擬似言語では基本的に 1 から始まります**。問題文に断り書きがあるのでかならず確認しましょう。

04 | for 構文

for 構文は、**for ~ endfor** の間の処理を、指定した条件にしたがって繰り返します。行番号 04 では次のようにになります。

繰り返しごとに値を変化させる変数

for (i を 1 から まで 1 ずつ増やす)

↑ 初期値 ↑ 終値 ↑ 増分値

繰り返し 1 回ごとに、変数 **i** の値を 1, 2, 3…のように 1 ずつ増やしていきます。**i** の値が の値になったときが最後の繰り返しになります。つまり、繰り返し回数は全部で 回になります。

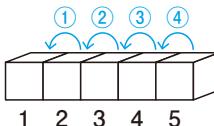
07 | 配列への代入

配列要素に値を代入するには、配列名の後に代入先の要素番号を [] で囲んで指定します。ここでは要素番号として「array の要素数」を指定しています。

本問では配列の要素番号が 1 から始まるため、array[array の要素数] は配列 array の末尾の要素を意味します。配列の要素番号が 0 から始まる場合、末尾の要素は array[array の要素数 - 1] になることに注意してください。

問題解説

行番号 04：図のように、配列の要素数が 5 個の場合、値を左に移す処理は全部で（要素数 - 1 =）4 回必要です。



行番号 04 は、

```
04  for (i を 1 から [a] まで 1 ずつ増やす)
05      [b]
06  endfor
```

のように、i が 1 から [a] になるまで [a] 回繰り返すので、空欄 a には「array の要素数 - 1」が入ります。

行番号 05：たとえば 1 回目の繰り返しでは、array の要素番号 2 の値を、array の要素番号 1 に移動します。変数 i = 1 を使って表すと、i 回目に array[i + 1] の値を array[i] に移動するので、空欄 b の処理は次のようにになります。

```
05  array[i] ← array[i + 1]
```

以上から、本問は空欄 a が「array の要素数 - 1」、空欄 b が「array[i] ← array[i + 1]」の組合せ **ウ** が正解となります。

array[i] の値を array[i + 1] に移動すると、配列の全要素が array[1] の値になってしまふよ。



解答

問02 **ウ**



合格の力ギ

配列の要素をずらす

配列の要素を 1 つずつ右または左にずらす処理は、実際のプログラムでもよくあります。左にずらす場合は配列の先頭から、右にずらす場合は配列の末尾から順にずらすことに注意します。

左にずらす場合



```
for(i を 1 から n-1 まで 1 ずつ増やす)
    array[i] ← array[i+1]
endfor
```

右にずらす場合



```
for(i を n から 2 まで 1 ずつ減らす)
    array[i] ← array[i-1]
endfor
```



03

次のプログラム中の **a** と **b** に入る正しい答えの組合せを、解答群の中から選べ。

関数 calcX と関数 calcY は、引数 inData を用いて計算を行い、その結果を戻り値とする。関数 calcX を calcX(1) として呼び出すと、関数 calcX の変数 num の値が、 $1 \rightarrow 3 \rightarrow 7 \rightarrow 13$ と変化し、戻り値は 13 となった。関数 calcY を calcY(1) として呼び出すと、関数 calcY の変数 num の値が、 $1 \rightarrow 5 \rightarrow 13 \rightarrow 25$ と変化し、戻り値は 25 となった。プログラム中の **a**、**b** に入る字句の適切な組合せはどれか。

[プログラム 1]

```
○ 整数型 : calcX( 整数型 : inData )
整数型 : num, i
num ← inData
for (i を 1 から 3 まで 1 ずつ増やす)
    num ← a
endfor
return num
```

[プログラム 2]

```
○ 整数型 : calcY( 整数型 : inData )
整数型 : num, i
num ← inData
for (b)
    num ← a
endfor
return num
```

解答群

	a	b
ア	$2 \times \text{num} + i$	i を 1 から 7 まで 3 ずつ増やす
イ	$2 \times \text{num} + i$	i を 2 から 6 まで 2 ずつ増やす
ウ	$\text{num} + 2 \times i$	i を 1 から 7 まで 3 ずつ増やす
エ	$\text{num} + 2 \times i$	i を 2 から 6 まで 2 ずつ増やす

※出典：令和4年度 ITパスポート
試験公表問題

プログラムの解説

[プログラム 1]

```

01 ○整数型 : calcX( 整数型 : inData)
02   整数型 : num, i
03   num ← inData
04   for (iを1から3まで1ずつ増やす)
05     num ← [a]
06   endfor
07   return num

```

[プログラム 2]

```

08 ○整数型 : calcY( 整数型 : inData)
09   整数型 : num, i
10   num ← inData
11   for ([b])
12     num ← [a]
13   endfor
14   return num

```

関数の呼び出し

数学で習う関数は、 $y = f(x)$ のように、 x の値に応じて、対応する y の値が決まるというものでした。関数 $f(x)$ が「 $f(x) = x + 2$ 」なら、 $x = 1$ のとき、対応する y の値は $y = f(1) = 1 + 2 = 3$ となります。

プログラムで使う関数も、基本的には数学の関数と同じ働きをします。プログラムでは、 x を引数、 y を戻り値といいます。引数には複数の値を指定できますが、戻り値は 1 つしかありません。

本問のプログラムでは、calcX, calcY という 2 つの関数が定義されています。問題文に、

「関数 calcX の引数に 1 を指定すると（中略）戻り値は 13 となった」

「関数 calcY の引数に 1 を指定すると（中略）戻り値は 25 となった」

とあるので、たとえば、

```

x ← calcX(1)
y ← calcY(1)

```

のようなプログラムを実行すると、変数 x には 13、変数 y には 25 が格納されます。

問題解説

行番号 05：プログラム 1 は、行番号 05 の処理を 3 回繰り返します。

```

04   for (i を 1 から 3 まで 1 ずつ増やす)
05     num ← [a]
06   endfor

```

1 + 2 → 3
 3 + 4 → 7
 7 + 6 → 13
 のように、増分が 2 ずつ増えています。



繰り返しごとに、変数 i の値は 1 ずつ増えます。また、 num の値は $1 \rightarrow 3 \rightarrow 7 \rightarrow 13$ と変化します。 num の増分は 2, 4, 6 のように 2 の倍数になっているので、次のような処理が行われていると考えられます。



合格のカギ

プログラムのトレース

プログラムの処理に応じて、変数の値がどのように変化するか追跡することをトレースといいます。変数の変化は、繰り返し 1 回ごとに表にするとわかりやすくなります。たとえば、プログラム 1 の行番号 05 の変数 num と変数 i の変化をトレースすると、右図のようになります。

	i	num	計算後の num
1	1	1 → 3	
2	3 → 7		
3	7 → 13		

num i

↓ ↓

繰返し 1 回目: $1 + 2 \times 1 \rightarrow 3$

繰返し 2 回目: $3 + 2 \times 2 \rightarrow 7$

繰返し 3 回目: $7 + 2 \times 3 \rightarrow 13$

以上から、空欄 a には「`num + 2 * i`」が入ります。

```

04   for (i を 1 から 3 まで 1 ずつ増やす)
05     num ← num + 2 * i
06   endfor
    ↑ 空欄 a
  
```

行番号 11 : プログラム 2 の空欄 a には、プログラム 1 と同じものが入ります。

```

11   for (b)
12     num ← num + 2 * i
13   endfor
    ↑ 空欄 a
  
```

プログラム 2 では、変数 `num` の値は $1 \rightarrow 5 \rightarrow 13 \rightarrow 25$ のように変化します。この変化をトレースしてみましょう。

①行番号 12 の右辺 「`num + 2 * i`」 に `num = 1` を代入します。

`num ← 1 + 2 * i`

このとき左辺の `num` は 5 になるので、`i` の値は 2 とわかります。 $1 + 2 \times i = 5$ より,
②「`num + 2 * i`」 に `num = 5` を代入すると, $i = (5 - 1) \div 2 = 2$

`num ← 5 + 2 * i`

このとき左辺の `num` は 13 になるので、`i` の値は 4 とわかります。 $5 + 2 \times i = 13$ より,
③「`num + 2 * i`」 に `num = 13` を代入すると, $i = (13 - 5) \div 2 = 4$

`num ← 13 + 2 * i`

このとき左辺の `num` は 25 になるので、`i` の値は 6 とわかります。 $13 + 2 \times i = 25$ より,
以上の結果をまとめると、次のようにになります。

num i

↓ ↓

繰返し 1 回目: $1 + 2 \times 2 \rightarrow 5$

繰返し 2 回目: $5 + 2 \times 4 \rightarrow 13$

繰返し 3 回目: $13 + 2 \times 6 \rightarrow 25$

`i` の値は 2 から 6 まで 2 ずつ増えているので、空欄 b は次のようにになります。

`for (i を 2 から 6 まで 2 ずつ増やす)` ← 空欄 b

以上から、正解の組合せは **エ** となります。

解答

問03 エ

 **問 04**

次のプログラム中の に入る正しい答えを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まるものとする。

関数 `multinomial` は、 $n + 2$ 個 ($n \geq 1$) の数値 $x, a_1, a_2, \dots, a_{n+1}$ を引数として受け取り、 n 次の多項式

$$a_1 x^n + a_2 x^{n-1} + \cdots + a_n x + a_{n+1}$$

を計算する。 a_1, a_2, \dots, a_{n+1} は、実数型の配列で表現する。

〔プログラム〕

○ 実数型 : `multinomial(実数型 : x, 実数型の配列 : a)`

```
実数型 : ret
整数型 : n
ret ← a[1]
for (n を 1 から (a の要素数 -1) まで 1 ずつ増やす)
    ret ← 
endfor
return ret
```

解答群

- | | | | |
|---|-----------------------------------|---|-----------------------------------|
| ア | <code>ret + (a[n + 1] × x)</code> | イ | <code>ret × (a[n + 1] + x)</code> |
| ウ | <code>(ret + x) × a[n + 1]</code> | エ | <code>(ret × x) + a[n + 1]</code> |
| オ | <code>(ret + a[n + 1]) × x</code> | カ | <code>(ret × a[n + 1]) + x</code> |

※オリジナル問題

プログラムの解説

```
01 ○ 実数型 : multinomial( 実数型 : x, 実数型の配列 : a )
02   実数型 : ret
03   整数型 : n
04   ret ← a[1]
05   for (n を 1 から (a の要素数 -1) まで 1 ずつ増やす)
06       ret ← 
07   endfor
08   return ret
```

ホーナー法

たとえば、4次の多項式

$$a_1x^4 + a_2x^3 + a_3x^2 + a_4x + a_5$$

を計算することを考えてみましょう。各項をそれぞれ計算して足し合わせると、

$$a_1 \times x \times x \times x \times x + a_2 \times x \times x \times x \times x + a_3 \times x \times x \times x + a_4 \times x + a_5$$

のように、掛け算が10回、足し算が3回必要になります。この式を次のように変形します。

$$\begin{aligned} a_1x^4 + a_2x^3 + a_3x^2 + a_4x + a_5 &= (a_1x^3 + a_2x^2 + a_3x + a_4)x + a_5 \\ &= ((a_1x^2 + a_2x + a_3)x + a_4)x + a_5 \\ &= (((a_1x + a_2)x + a_3)x + a_4)x + a_5 \end{aligned}$$

変形後の式では、掛け算が4回、足し算が4回で済みます。このような多項式の計算方法を**ホーナー法**といいます。

問題解説

行番号 06：本問のプログラムでは、行番号 06 の繰返しごとに

1回目： $a_1x + a_2$

2回目： $(a_1x + a_2)x + a_3$

3回目： $((a_1x + a_2)x + a_3)x + a_4$

4回目： $(((a_1x + a_2)x + a_3)x + a_4)x + a_5$

⋮

カッコの中に、前回の繰返し処理の計算結果が入るよ。



を計算します。n回目の繰返しごとに、n次の多項式の計算結果が変数 `ret` に格納されます。上記のように、n次の多項式は、

$$(n-1\text{次の多項式}) \times x + a_{n+1}$$

で求めることができますので、行番号 06 は次のようにになります。

$$\text{ret} \leftarrow (\text{ret} \times x) + a[n+1]$$

以上から、正解は **工** です。

解答

問04 **工**



合格のカギ

単純な例を当てはめて考える

本問ではホーナー法を使ったアルゴリズムをとりあげていますが、ホーナー法を知らないと問題が解けないということはありません。たとえば、1次式 $a_1x + a_2$ の計算をこのプログラムに当てはめると、行番号 04 で変数 `ret` に a_1 の値が格納されるので、行番号 06 では「`ret × x + a2`」を求めればよいことがわかります。このように、単純な実例をプログラムに当てはめてみて、必要な処理を考えるのが問題を解くコツになります。

 **問 05**

次のプログラム中の と に入る正しい答えの組合せを、解答群の中から選べ。

関数 gcd は、引数に指定した 2 つの正の整数 a, b ($a > b$) の最大公約数を返す。プログラム 1 とプログラム 2 は、関数 gcd を異なる方法で定義したものである。

[プログラム 1]

○ 整数型 : gcd(整数型 : a, 整数型 : b)

整数型 : m, n, r

$m \leftarrow a$

$n \leftarrow b$

do

$r \leftarrow m \bmod n$

$m \leftarrow n$

$n \leftarrow r$

while (r が 0 でない)

return

[プログラム 2]

○ 整数型 : gcd(整数型 : a, 整数型 : b)

整数型 : m, n, r

$m \leftarrow a$

$n \leftarrow b$

$r \leftarrow m \bmod n$

while (r が 0 でない)

$m \leftarrow n$

$n \leftarrow r$

$r \leftarrow m \bmod n$

endwhile

return

解答群

	a	b
ア	m	m
イ	m	n
ウ	n	m
エ	n	n

※オリジナル問題

プログラムの解説

[プログラム 1]

```

01 ○ 整数型 : gcd( 整数型 :a, 整数型 :b)
02 整数型 : m, n, r
03 m ← a
04 n ← b
05 do
06   r ← m mod n
07   m ← n
08   n ← r
09   while (r が 0 でない)
10   return a

```

[プログラム 2]

```

01 ○ 整数型 : gcd( 整数型 :a, 整数型 :b)
02 整数型 : m, n, r
03 m ← a
04 n ← b
05 r ← m mod n
06 while (r が 0 でない)
07   m ← n
08   n ← r
09   r ← m mod n
10 endwhile
11 return b

```

ユークリッドの互除法

本問のプログラムでは、**ユークリッドの互除法**と呼ばれるアルゴリズムを用いて、2つの整数 a, b ($a > b$) の**最大公約数**を求めます。このアルゴリズムは、

- ① $a \div b$ の余り r を求める
- ② a を b の値に、 b を r の値に置き換える

という手順を繰り返すと、 r が 0 になったときの b が最大公約数になるというものです。たとえば $a = 176$, $b = 99$ の場合は、次のようにになります。

$$\begin{array}{r}
 \begin{array}{ccc} a & b & r \\ 176 & \div & 99 \\ \downarrow & \downarrow & \downarrow \\ 99 & \div & 77 \\ \downarrow & \downarrow & \downarrow \\ 77 & \div & 22 \\ \downarrow & \downarrow & \downarrow \\ 22 & \div & 11 \\ \downarrow & \downarrow & \downarrow \\ \text{最大公約数} & & 0
 \end{array}
 \end{array}$$

問題解説

空欄には、この関数の戻り値、すなわち最大公約数が格納されている変数が入ります。繰返し処理を抜けたとき、最大公約数はどの変数に格納されているかが、この問題のポイントです。

ユークリッドの互除法は、 $a \div b$ の余り r が 0 になったときの b が最大公約数です。プログラム 1 では、

```

05 do
06   r ← m mod n
07   m ← n
08   n ← r
09   while (r が 0 でない)

```

mod は、 $m \div n$ の余りを求める演算子だよ。



のように、行番号 06 で $m \div n$ の余り r を求めているので、最大公約数は n に格納されます。その後、行番号 07 と 08 で、 n の値は m に、 r の値は n にそれぞれ代入されるので、最大公約数は変数 m に移ります。

以上から、繰返し処理を抜けたとき、最大公約数は変数 m に格納されています。

10 `return m ← 最大公約数`

以上から、空欄 a には「m」が入ります。

一方、プログラム 2 では、

06 `while (r が 0 でない)`
 07 `m ← n`
 08 `n ← r`
 09 `r ← m mod n`
 10 `endwhile`

のように行番号 09 で $m \mod n$ を実行しているので、最大公約数は n に格納されます。その後、プログラムは行番号 06 に戻って、 r が 0 なら繰返し処理を終了します。したがって、最大公約数は変数 n に格納されたままです。

11 `return n ← 最大公約数`

したがって、空欄 b には「n」が入ります。以上から、空欄 a が「m」、空欄 b が「n」のイが正解となります。

○ 解答 ○

問05 イ

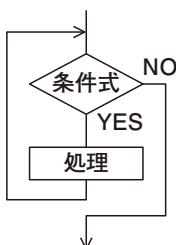


合格の力ギ

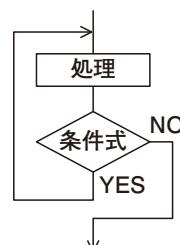
while ~ endwhile 構文と do ~ while 構文

while ~ endwhile と do ~ while は、どちらも条件式が真の間、処理を繰り返し実行します。

while (条件式)
 処理
endwhile



do
 処理
while(条件式)



2つの構文の違いは、while ~ endwhile が繰返し処理の入口で条件式を評価するのに対し、do ~ while は、繰返し処理の出口で条件式を評価することです。そのため、while ~ endwhile では繰返し処理が 1 度も実行されないことがあるのに対し、do ~ while では最低 1 回はかならず繰返し処理が実行されます。

一般に、do ~ while は while ~ endwhile に書き替えることができます。そのためプログラム言語によっては、do ~ while に当たる構文がないものもあります（Python など）。



06

次のプログラム中の と に入れる正しい答えの組合せを、
解答群の中から選べ。

関数 fact は、0 以上の整数 n を引数として受け取り、n の階乗を返す。なお、0 の階乗は 1 とする。

[プログラム]

○ 整数型 : fact(整数型 :n)

```
if (n が 0 より大きい)
    return  a
else
    return  b
```

解答群

	a	b
ア	fact(n + 1)	1
イ	fact(n - 1)	1
ウ	$n \times \text{fact}(n + 1)$	1
エ	$n \times \text{fact}(n - 1)$	1
オ	fact(n + 1)	0
カ	fact(n - 1)	0
キ	$n \times \text{fact}(n + 1)$	0
ク	$n \times \text{fact}(n - 1)$	0

※オリジナル問題

プログラムの解説

```

01 ○ 整数型 : fact( 整数型 : n )
02   if (n が 0 より大きい )
03     return [ ] a
04   else
05     return [ ] b ← n が 0 のとき戻り値

```

| n の階乗

n の階乗とは、1 から n までの整数を順番に掛けたもので、記号では n の階乗を $n!$ のように書きます。

$$n! = n \times (n - 1) \times (n - 2) \times \cdots \times 3 \times 2 \times 1$$

なお、0 の階乗 $0!$ は 1 とします。

例 : $3! = 3 \times 2 \times 1 = 6$ $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

問題解説

行番号 03 : n の階乗 $n!$ は、「 $n \times (n - 1)!$ 」のように書き換えることができます。 $(n - 1)!$ は、関数 `fact` を使って「`fact(n - 1)`」で求められますから、行番号 03 は次のようにになります。

```
03 return n × fact(n - 1) ← 空欄 a
```

このように、関数の処理の中で自分自身を呼び出すプログラムを再帰的プログラムといいます。

行番号 05 : $0! = 1$ なので、 $n = 0$ の場合は 1 を返します。

```
05 return 1 ← 空欄 b
```

以上から、空欄 a が「 $n \times fact(n - 1)$ 」、空欄 b が「1」の **□** が正解となります。

解答

問06 □



合格のカギ 再帰的プログラム

関数や手続の処理の中で、自分自身を呼び出すプログラムを**再帰的プログラム**といいます。たとえば関数 a の処理の中でさらに関数 a が呼び出され、その処理の中でさらに関数 a が呼び出され…のように、何段にも入れ子になって関数が呼び出されるプログラムです。

再帰的プログラムでは、「これ以上は自分自身を呼び出さない」という条件がかならず設定されます。この条件がないと、無限に自分自身を呼び出し続けることになってしまうからです。

再帰的プログラムは、クイックソート、マージソート、木構造の探索など、様々なアルゴリズムで使われるよ。





07

次のプログラムの説明中の [] に入る正しい答えを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まる。

関数 `calc` は、後置記法（逆ポーランド記法）で表現された数式を引数で受け取り、計算結果を返す。使用する演算子は、+（足し算）、-（引き算）、×（掛け算）、÷（割り算）の 4 種類とする。数式は文字列型の配列で表す。

数式の解析にスタックを用いる。スタックはクラス `Stack` を用いて表現する。クラス `Stack` の説明を図に示す。`Stack` 型の変数は、クラス `Stack` のインスタンスへの参照を格納する。

メソッド	説明
<code>push(整数型 : num)</code>	整数 <code>num</code> をスタックに格納する。
整数型 : <code>pop()</code>	スタックから整数を 1 つ取り出し、その整数を返す。

コンストラクタ	説明
<code>Stack()</code>	スタックを生成する。スタックの大きさは十分あるものとする。

図 クラス `Stack` の説明

関数 `calc` に次のような引数を指定して呼び出したとき、プログラムの α の行が 1 回目に実行した直後のスタックの内容は [] である。

`calc(["12", "3", "4", "×", "+", "8", "÷"])`

〔プログラム〕

```
○ 整数型 : calc( 文字列型の配列 : array )
整数型 : i, num1, num2, ans
文字列型 : tmp
Stack: stack ← Stack()

for (i を 1 から array の要素数 まで 1 ずつ増やす)
    tmp ← array[i]
    if (tmp が数字のみからなる文字列)
        stack.push(tmp を整数值に変換した値)
    else
        num2 ← stack.pop()
```

```
num1 ← stack.pop()
if (tmp が " + ")
    ans ← num1 + num2
elseif (tmp が " - ")
    ans ← num1 - num2
elseif (tmp が " × ")
    ans ← num1 × num2
elseif (tmp が " ÷ ")
    ans ← num1 ÷ num2 の商
endif
stack.push(ans) ← α
endif
endfor
return stack.pop()
```

解答群

- ア 12
- イ 12, 3
- ウ 12, 3, 4
- エ 12, 12
- オ 24
- カ 空

※オリジナル問題

プログラムの解説

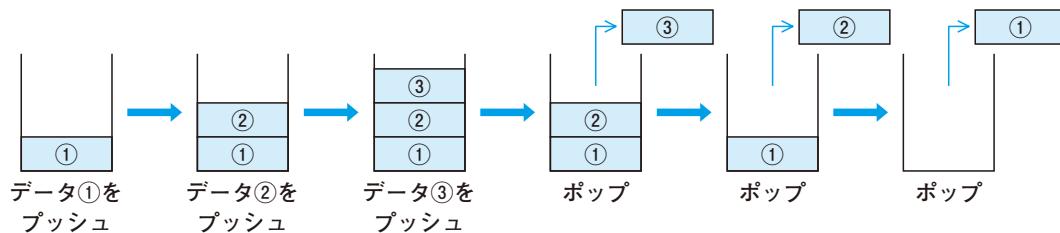
```
01 ○ 整数型 : calc( 文字列型の配列 : array )
02 整数型 : i, num1, num2, ans
03 文字列型 : tmp
04 Stack: stack ← Stack()

05 for (i を 1 から array の要素数 まで 1 ずつ増やす )
06     tmp ← array[i]
07     if (tmp が数字のみからなる文字列 )
08         stack.push(tmp を整数値に変換した値 ) ← 数値はスタックにpush
09     else
10         num2 ← stack.pop() } ← + - × ÷ だった場合は、スタックから数値を
11         num1 ← stack.pop() } 2個取り出す
12         if (tmp が " + ")
13             ans ← num1 + num2
14         elseif (tmp が " - ")
15             ans ← num1 - num2
16         elseif (tmp が " × ")
17             ans ← num1 × num2
18         elseif (tmp が " ÷ ")
19             ans ← num1 ÷ num2 の商
20         endif
21         stack.push(ans) ← α ← 計算結果をスタックにpush
22     endif
23 endfor
24 return stack.pop() ← 最後に計算結果を取り出す
```

【スタック】

「**スタック**」とは、**プッシュ**と**ポップ**という、2つの基本操作をもったデータ構造です。プッシュはスタックにデータを格納する操作、ポップはスタックからデータを取り出す操作です。このとき重要なのは、後に格納したデータから先に取り出されるという、**後入れ先出し** (LIFO : Last In First Out) の原則です。

たとえば、データを1, 2, 3の順にスタックにプッシュします。次にスタックからデータをポップすると、取り出されるデータは3, 2, 1の順になります。格納するときはデータを一番上に積み上げ、取り出すときは一番上から取り出すイメージです。



後置記法(逆ポーランド記法)

日常的に使っている「 $3 + 2$ 」のような数式は、演算子を2つの数値の間に置くことから、中置記法と呼ばれることがあります。これに対し、「 $32 +$ 」のように、演算子を2つの数値の後に置く表記法を後置記法(逆ポーランド記法)といいます。

中置記法には「 $+ - \times \div$ を先に計算する」「カッコで囲んだ部分を先に計算する」といったルールがありますが、後置記法にはこのようなルールはありません。単に、演算子を左から順に処理していくだけです。

中置記法

例： $(1 + 2) \times (5 - 3)$

後置記法

 $12 + 53 - \times$

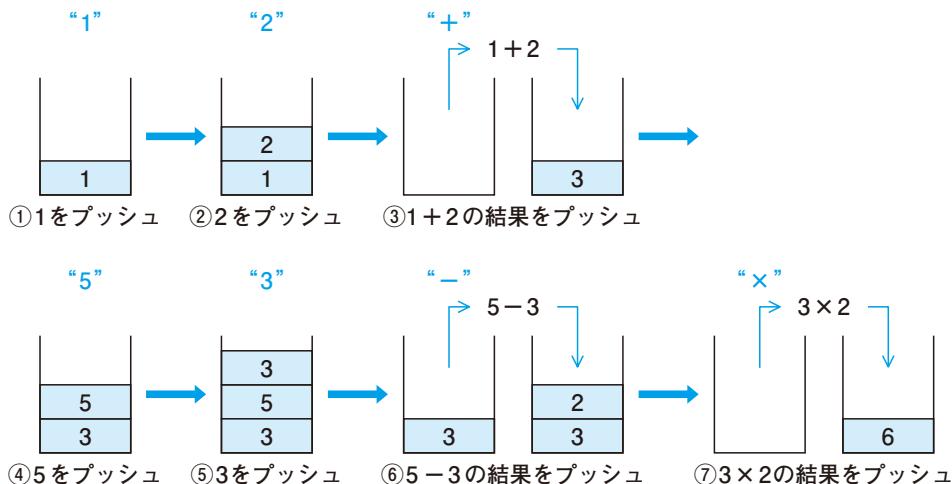
後置記法とスタック

後置記法による数式は、次のようにスタックを利用するとうまく計算できるという特徴があります。数式を左から順に調べて、

- ①数値の場合はスタックにpushする
- ②演算子 (+ - × ÷) の場合はスタックから数値を2個popして式を計算し、結果をスタックにpushする

この操作を数式の最後まで繰り返すと、最後にスタックに残った数値が式全体の計算結果になります。

先程の例「 $12 + 53 - \times$ 」をスタックを使って計算すると、次のようにになります。



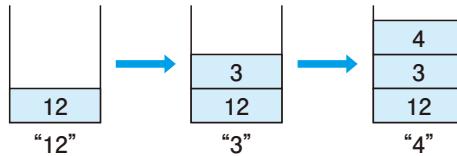
問題解説

行番号 21 の α の行は、変数 `ans` に格納された数値をスタックにpushします。変数 `ans` には、演算子の種類によって、 $+ - \times \div$ のいずれかの計算結果が格納されます。したがって、この行が1回目に実行されるのは、数式の1回目の演算子が処理されたときです。

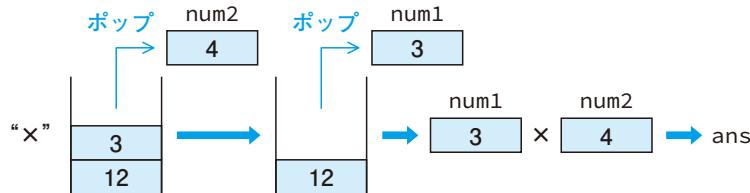
そこで、関数 `calc` に指定された引数の数式をみると、

```
calc(["12", "3", "4", "×", "+", "8", "÷"])
```

となっており、最初の3つは数値が続きます。これらは順番にスタックにpushされます。



次に、演算子“ \times ”がくるので、上の2つの数値がポップされ、 3×4 の計算結果が変数 ans に格納されます。



1回目の α の行では、この値をスタックにプッシュします。したがって、スタックの内容は次のようになります。



以上から、正解は **エ** です。

問07 **エ**



合格のカギ

スタック

スタックは基本的なデータ構造なので、プログラム問題ではよく出題されると予想されます。また、後置記法で記述された式をスタックを使って計算するプログラムは、アルゴリズムの解説書でもよくとりあげられる題材です。



08

次のプログラム中の と に入る正しい答えの組合せを、解答群の中から選べ。

手続 `append` は、引数で与えられた文字を单方向リストに追加する手続である。单方向リストの各要素は、クラス `ListElement` を用いて表現する。クラス `ListElement` の説明を図に示す。`ListElement` 型の変数はクラス `ListElement` のインスタンスの参照を格納するものとする。大域変数 `listHead` は、单方向リストの先頭の要素の参照を格納する。リストが空のときは、`listHead` は未定義である。

メンバ変数	型	説明
<code>val</code>	文字型	リストに格納する文字。
<code>next</code>	<code>ListElement</code>	リストの次の文字を保持するインスタンスの参照。 初期状態は未定義である。
コンストラクタ		説明
<code>ListElement(文字型 : qVal)</code>		引数 <code>qVal</code> でメンバ変数 <code>val</code> を初期化する。

図 クラス `ListElement` の説明

〔プログラム〕

```

01 大域 : ListElement: listHead ← 未定義の値
02 Oappend( 文字型 : qVal )
03   ListElement: prev, curr
04   curr ← ListElement(qVal)
05   if (listHead が  )
06     listHead ← curr
07   else
08     prev ← listHead
09     while (prev.next が 未定義でない )
10       prev ← prev.next
11     endwhile
12     prev.next ← 
13   endif

```

解答群

	a	b
ア	未定義	curr
イ	未定義	curr.next
ウ	未定義	listHead
エ	未定義でない	curr
オ	未定義でない	curr.next
カ	未定義でない	listHead

※サンプル問題より

```

01 大域 : ListElement: listHead ← 未定義の値

02 Oappend( 文字型 : qVal)
03   ListElement: prev, curr
04   curr ← ListElement(qVal)
05   if (listHead が [ a ])
06     listHead ← curr
07   else
08     prev ← listHead
09     while (prev.next が 未定義でない)
10       prev ← prev.next
11   endwhile
12   prev.next ← [ b ]
13 endif

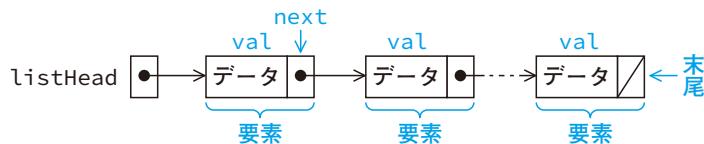
```

単方向リスト

単方向リスト（線形リスト）は、リストを構成する個々の要素（データの入れ物）を数珠つなぎにして、複数のデータをひと連なりにまとめたデータ構造です。個々の要素は、

- ①データを格納する部分（val）
- ②次の要素の参照を格納する部分（next）

で構成されています。前の要素の next に、次の要素の参照を格納し、次の要素の next にまた次の要素の参照を格納し——という操作を繰り返して、複数の要素をつないでいきます。



単方向リストに含まれる要素を参照するには、先頭要素から次の要素へと、目的の要素に到達するまでリストを順にたどっていきます。

クラスの定義

単方向リストの要素のような構造をもったデータは、整数型、文字型といった通常のデータ型では表現できません。そのため本問では、ListElement というクラスを用いてリストの要素を表現しています。

クラスとは、本来はオブジェクト指向プログラミングの用語ですが、ここでは複数のメンバ変数とメソッドを備えたデータ型の一種と考えてかまいません。クラスを定義するための構文は擬似言語には用意されていないので、使用するクラスについては問題文に説明があります。

クラスを使用するには、通常のデータ型と同様に、

クラス名 : **変数名**

のように変数を宣言します。本問のプログラムでは、行番号 03 で `prev`, `curr` という 2 つの `ListElement` 型の変数を宣言しています。

クラスの変数は参照を格納するだけなので、クラスの実体（オブジェクト）を作成する場合は **コンストラクタ** を使い、

変数名 ← コンストラクタ()

のように変数を初期化する必要があります。本問のプログラムでは、行番号 04 で `ListElement` クラスのコンストラクタを実行し、参照を変数 `curr` に格納しています。コンストラクタによって作成されるクラスの実体（オブジェクト）を **インスタンス** といいます。

問題解説

単方向リストに新しい要素を追加するには、リストの末尾の要素を取得し、その要素のメンバ変数 `next` に、追加する要素への参照を格納します。

行番号 04：リストの新しい要素を作成し、引数に指定された文字列 `qVal` を格納します。作成した要素への参照は、変数 `curr` に設定しておきます。

04 curr ← ListElement(qVal)

行番号 05, 06：現在のリストが空の場合は作成した要素がリストの先頭要素になります。6 行目で、先頭要素への参照を格納する大域変数 `listHead` に、作成した要素への参照 `curr` を格納しているので、5 行目の `if` 文の条件式には「リストが空の場合」に真となる条件式が入ります。リストが空のとき、`listHead` は未定義なので、次のようにになります。

05 if (listHead が 未定義) ← リストが空のとき
06 listHead ← curr ← 追加要素を先頭要素に設定

行番号 08～11：現在のリストが空でない場合は、`else` 以下の処理を実行します。まず、変数 `prev` をリストの先頭要素に設定してから、`next` が未定義になるまで、リストを順番にたどります。この処理が終わると、変数 `prev` にリストの末尾の要素が設定されます。

08 prev ← listHead ← prev に先頭要素を設定
09 while (prev.next が 未定義でない) ← 次の要素が存在するあいだ、処理を繰り返す
10 prev ← prev.next ← prev に次の要素を設定
11 endwhile

行番号 12：変数 `prev` にリストの末尾の要素が設定されたら、その `next` に、追加する要素への参照を格納します。

12 prev.next ← curr ← 末尾要素のnextに、追加要素への参照を格納する

以上から、空欄 **a** には「未定義」、空欄 **b** には「`curr`」が入ります。
正解は **ア** です。

解答

問08 ア



合格の力ギ

単方向リスト

複数のデータをまとめて扱う方法として、もっとも単純なデータ構造は配列ですが、リストも非常によく利用されます。要素の追加のほか、要素の挿入や削除についても調べておきましょう。



問 09

次のプログラム中の [] に入る正しい答えの組合せを、解答群の中から選べ。

手続 `traverse` は、二分木の各節を走査し、節の値を出力する。二分木の各節はクラス `Node` によって表現する。クラス `Node` の説明を図 1 に示す。`Node` 型の変数は、クラス `Node` のインスタンスへの参照を格納するものとする。

メンバ変数	説明
文字型 : <code>val</code>	節の値を格納する。
<code>Node: left</code>	節の左の子の参照を格納する。左の子がない場合は未定義となる。
<code>Node: right</code>	節の右の子の参照を格納する。右の子がない場合は未定義となる。

図 1 クラス `Node` の説明

図 2 に示す二分木の根への参照を引数に指定し、手続 `traverse` を実行したところ、出力結果は “CBDAFEG” となった。

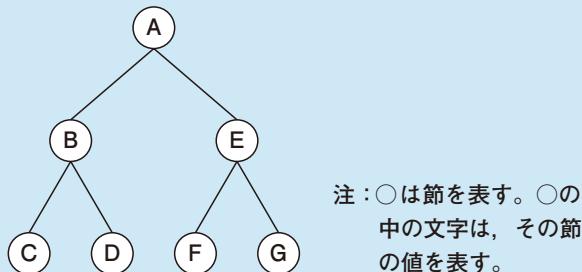


図 2 二分木の例

[プログラム]

```
○ traverse (Node: node)
  if (node が 未定義 )
    return
  else
    [ ]
  endif
```

解答群

ア node.val を 1 文字出力する

```
traverse(node.left)  
traverse(node.right)
```

イ node.val を 1 文字出力する

```
traverse(node.right)  
traverse(node.left)
```

ウ traverse(node.left)

node.val を 1 文字出力する

```
traverse(node.right)
```

エ traverse(node.right)

node.val を 1 文字出力する

```
traverse(node.left)
```

オ traverse(node.left)

```
traverse(node.right)  
node.val を 1 文字出力する
```

カ traverse(node.right)

```
traverse(node.left)  
node.val を 1 文字出力する
```

※オリジナル問題

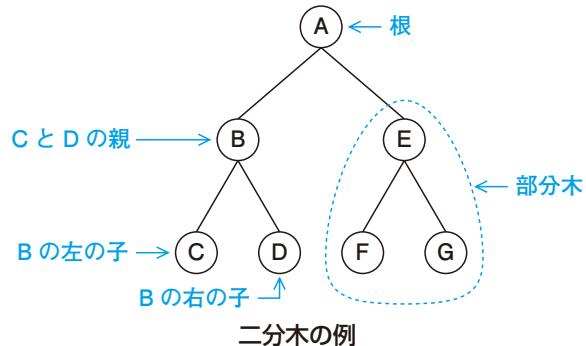
プログラムの解説

```
01 ○traverse (Node: node)  
02     if (node が 未定義 )  
03         return ← node が空のときは何もしない  
04     else  
05         [ ]  
06     endif
```

二分木

複数の節（ノード）をツリー状に接続したデータ構造を**木構造**といいます。下位に接続された節を**子**、上位に接続された節を**親**といい、木構造の頂点にあるもっとも上位の節を**根**といいます。

木構造のうち、2つ以上の子をもたないものをとくに**二分木**といいます。



二分木の走査

二分木を構成する各節を漏れなく訪ねることを**走査**といいます。二分木の走査には、①**間順走査**、②**前順走査**、③**後順走査**の3種類があります。ここでは間順走査について説明しましょう。

間順走査は、次のような順序で行います。

- ①現在の節の左側の子を根とする部分木を走査する（左側の子がない場合は何もしない）
- ②現在の節の値を出力する
- ③現在の節の右側の子を根とする部分木を走査する（右側の子がない場合は何もしない）

手順①と③は再帰的な手続きになっていることに注意しましょう。図2の二分木を上の手順に沿って走査すると、次のようにになります。

- ① “A” の節の左側の子を根とする部分木を走査する
 - ① “B” の節の左側の子を根とする部分木を走査する
 - ① “C” の節の左側の子はないので何もしない
 - ② “C” を出力する
 - ③ “C” の節の右側の子はないので何もしない
 - ② “B” を出力する
 - ③ “B” の節の右側の子を根とする部分木を走査する
 - ① “D” の節の左側の子はないので何もしない
 - ② “D” を出力する
 - ③ “D” の節の右側の子はないので何もしない
- ② “A” を出力する。
- ③ “A” の節の右側の子を根とする部分木を走査する
 - ① “E” の節の左側の子を根とする部分木を走査する
 - ① “F” の節の左側の子はないので何もしない
 - ② “F” を出力する
 - ③ “F” の節の右側の子はないので何もしない
 - ② “E” を出力する
 - ③ “E” の節の右側の子を根とする部分木を走査する
 - ① “G” の節の左側の子はないので何もしない
 - ② “G” を出力する
 - ③ “G” の節の右側の子はないので何もしない

以上から、値が出力される順序は C → B → D → A → F → E → G となります。

問題解説

出力結果 “CBDAFEG” を、次のように分けて考えます。

((CBD) A (FEG))

(CBD) と (FEG) は、図 2 の二分木の部分をそれぞれ左の子→親→右の子の順に出力したものです。さらに、(CBD) → A → (FEG) という順序も、“A” の節を親として、左の子→親→右の子の順になっています。

したがって空欄の処理は、

traverse (node.left)	◀ 左の子を走査
node.val を 1 文字出力する	◀ この節の値を出力
traverse (node.right)	◀ 右の子を走査

のような間順走査となります。正解は **ウ** です。

○ 解答 ○

問09 **ウ**



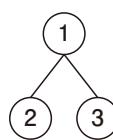
合格のカギ

二分木の走査

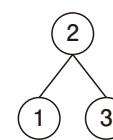
前順走査と後順走査は、次のようにになります。

前順走査

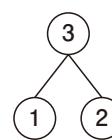
- ①現在の節の値を出力する
- ②現在の節の左側の子を根とする部分木を走査する
- ③現在の節の右側の子を根とする部分木を走査する



前順走査
(行きかけ順)



間順走査
(通りかけ順)



後順走査
(帰りかけ順)

後順走査

- ①現在の節の左側の子を根とする部分木を走査する
- ②現在の節の右側の子を根とする部分木を走査する
- ③現在の節の値を出力する

根を出力するタイミングによって、「行きかけ順」「通りかけ順」「帰りかけ順」ともいうよ。



図 2 の二分木を前順走査すると、出力結果は “ABCDEFG” となります。また、後順走査した場合は “CDBFGEA” となります。

なお、前順走査はあまり使われません。後順走査と間順走査を覚えておきましょう。



問 10

次の記述中の [] に入る正しい答えを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まる。

次のプログラムは、整数型の配列 array を、挿入ソートで昇順に整列する。このプログラムを実行したとき、 α の行を 2 回目に実行した直後の配列 array の内容は、[] である。

[プログラム]

整数型の配列： array = {3, 2, 1, 5, 4}

整数型： i, j, tmp

```
for (i を 2 から array の要素数 まで 1 ずつ増やす)
    tmp ← array[i]
    j ← i
    while ((j > 1) and (array[j - 1] > tmp))
        array[j] ← array[j - 1]
        j ← j - 1
    endwhile
    array[j] ← tmp 
endfor
```

解答群

- ア {1, 2, 3, 5, 4}
- イ {1, 2, 3, 4, 5}
- ウ {2, 1, 3, 5, 4}
- エ {2, 3, 1, 5, 4}
- オ {3, 1, 2, 5, 4}
- カ {3, 2, 1, 5, 4}

※オリジナル問題

プログラムの解説

```
01 整数型の配列： array = {3, 2, 1, 5, 4}
02 整数型： i, j, tmp
03 for (i を 2 から array の要素数 まで 1 ずつ増やす)
04     tmp ← array[i] ← 未整列の要素を変数 tmp に格納
05     j ← i ← 変数 j は tmp の挿入位置
06     while ((j > 1) and (array[j - 1] > tmp))
07         array[j] ← array[j - 1] ← tmp より大きい要素を右側にずらす
08         j ← j - 1
09     endwhile
10     array[j] ← tmp ← α
11 endfor
```

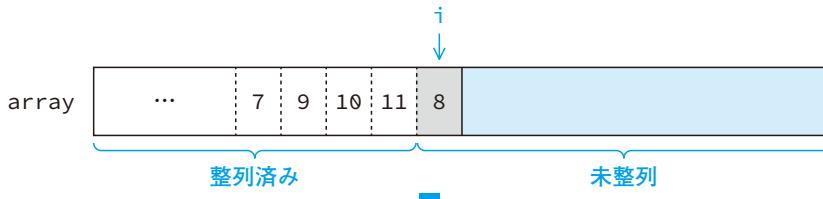
挿入ソート

データを昇順や降順に整列するためのアルゴリズムを、**ソートアルゴリズム**といいます。本問では、代表的なソートアルゴリズムのひとつである**挿入ソート**を取り上げます。

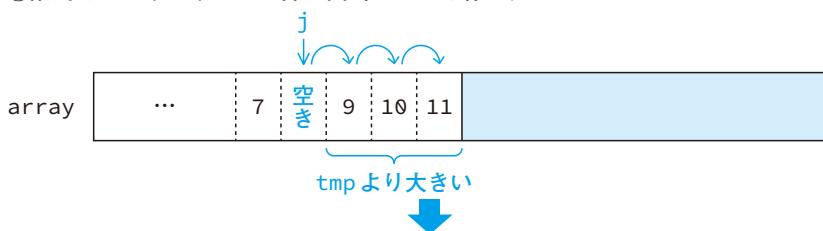
挿入ソートでは、配列を整列済みの部分と未整列の部分とに分け、未整列の部分から要素を1つ選んで、整列済みの部分の適切な位置に挿入します。この作業を繰り返して整列済みの部分を増やしていく、未整列の部分がなくなると整列が完了します。

本問のプログラムでは、配列 `array` の先頭から要素番号 `i` の手前までを整列済みの部分とします。`array[i]` の値（変数 `tmp` に格納）が適切な挿入位置になるまで整列済みの要素をひとつずつ右にずらしていく、見つかった場所 (`array[j]`) にその値を格納します。

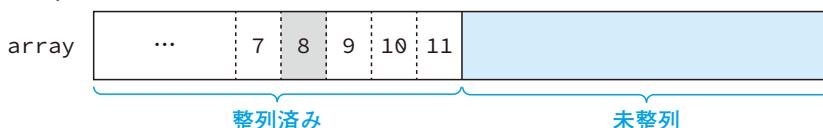
- ①array[i] の値を tmp に格納する。



- ②配列を1つずつずらして挿入箇所に空きを作る。



- ③tmpの値を整列済みの部分に挿入する。



問題解説

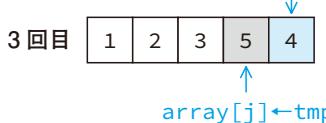
α の行は、変数 tmp に格納された値を、整列済みの部分に挿入する処理です。この行は、行番号 03 の for ~ endfor 構文によって、(array の要素数 - 1) 回繰り返されます。

配列の変化をトレースすると、次のようにになります。

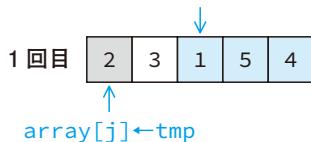
次はこの値を整列済みにする



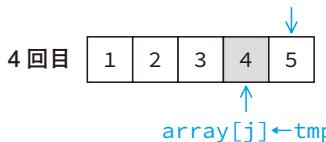
次はこの値を整列済みにする



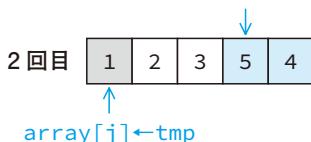
次はこの値を整列済みにする



次はこの値を整列済みにする



次はこの値を整列済みにする



以上から、2回目の α の行を実行後の配列の内容は、
 $\{1, 2, 3, 5, 4\}$ になります。正解は ア です。

解答

問10 ア



合格のカギ

ソートアルゴリズム

ソートアルゴリズムには挿入ソート以外にもいくつか種類があります。代表的なものを挙げておこうので確認しておきましょう。

- ・バブルソート
- ・選択ソート
- ・挿入ソート
- ・シェルソート
- ・クイックソート
- ・マージソート

科目 B はあまり長いプログラムは出題されないけど、アルゴリズムの一部が
出題される可能性はあるよ。





11

次のプログラム中の [] に入る正しい答えを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まる。

関数 `checkDigit` は、10 進 9 桁の整数の各桁の数字が上位の桁から順に格納された整数型の配列 `originalDigit` を引数として、次の手順で計算したチェックディジットを戻り値とする。

〔手順〕

- (1) 配列 `originalDigit` の要素番号 1 ~ 9 の要素の値を合計する。
- (2) 合計した値が 9 より大きい場合は、合計した値を 10 進の整数で表現したときの各桁の数字を合計する。この操作を、合計した値が 9 以下になるまで繰り返す。
- (3) (2) で得られた値をチェックディジットとする。

〔プログラム〕

```
○ 整数型 : checkDigit( 整数型の配列 : originalDigit )
    整数型 : i, j, k
    j ← 0
    for (i を 1 から originalDigit の要素数 まで 1 ずつ増やす )
        j ← j + originalDigit[i]
    endfor
    while (j が 9 より大きい )
        k ← j ÷ 10 の商 /* 10 進 9 桁の数の場合 , j が 2 桁を超えることはない */
        [ ]
    endwhile
    return j
```

解答群

- ア** $j \leftarrow j - 10 \times k$
- イ** $j \leftarrow k + (j - 10 \times k)$
- ウ** $j \leftarrow k + (j - 10) \times k$
- エ** $j \leftarrow k + j$

※出典：令和 4 年度 IT パスポート試験公表問題を改変

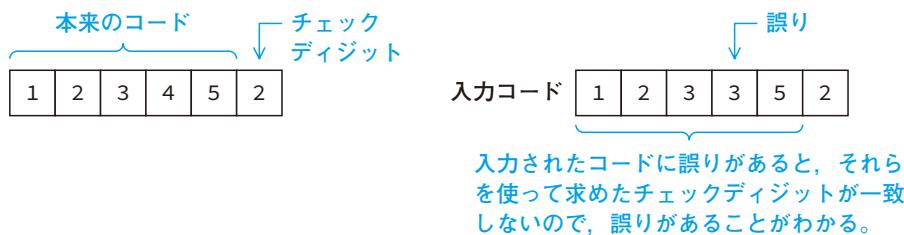
プログラムの解説

```
01 ○ 整数型 : checkDigit( 整数型の配列 : originalDigit )
02 整数型 : i, j, k
03 j ← 0
04 for (i を 1 から originalDigit の要素数 まで 1 ずつ増やす )
05   j ← j + originalDigit[i]
06 endfor
07 while (j が 9 より大きい)
08   k ← j ÷ 10 の商 /* 10 進9桁の数の場合 , j が2桁を超えることはない */
09   [ ]
10 endwhile
11 return j
```

チェックディジット

チェックディジットとは、コード番号などの数字の一連の並びをもとに、ある一定の手順で計算した数字で、そのコードが正しく入力されているかどうかを確認するために使います。

入力したコードに誤りがあると、そのコードから計算したチェックディジットが一致しないので、誤っていることがわかる仕組みです。



問題解説

プログラムは、問題文の計算手順 (1) ~ (3) にしたがっています。手順とプログラムとの対応を確認しましょう。

(1) 配列 originalDigit の要素番号 1 ~ 9 の要素の値を合計する。

配列 originalDigit の各要素の合計を求めます。プログラムでは、この処理を次のような繰返し処理によって行います。

```
03   j ← 0
04   for (iを1からoriginalDigitの要素数まで1ずつ増やす)
05     j ← j + originalDigit[i]
06   endfor
```

以上で、変数 j に配列 originalDigit の各要素の合計が格納されます。

- (2) 合計した値が 9 より大きい場合は、合計した値を 10 進の整数で表現したときの各桁の数字を合計する。この操作を、合計した値が 9 以下になるまで繰り返す。

「この操作を、合計した値が 9 以下になるまで繰り返す」とあるので、この処理がプログラムの次の部分に対応していることがわかります。

07	while (j が 9 より大きい)
	:
10	endwhile

while ~ endwhile の間で実行する処理の内容は、「合計した値を 10 進の整数で表現したときの各桁の数字を合計する」というものです。合計した値は変数 j に格納されています。この値から、10 進整数で表したときの 10 の位の数と 1 の位の数を取り出して足し合わせます。たとえば変数 j の値が「78」であれば、7 と 8 を取り出し、 $7 + 8 = 15$ を求めればよいわけです。

行番号 08 は、10 の位の数を求めて変数 k に格納する処理です。

08	k ← j ÷ 10 の商
----	---------------

1 の位の数を求めるにはどうすればよいでしょうか。たとえば 78 の 1 の位の数「8」を取り出すには、78 から 70 を引けばよいでしょう。プログラムでは、j から $k \times 10$ を引けばよいことがわかります。

j ← k × 10

ここで必要なのは、10 の位の数と 1 の位の数の合計ですから、空欄の処理は次のようにになります。

j ← k + (j - k × 10)

以上から、正解は **イ** です。

解答

問11 **イ**



合格のカギ

プログラムの仕様とプログラムの対応

本問では、チェックディジットを計算する問題文の手順（1）～（3）とプログラムの各行に、

手順（1） ⇔ 行番号 03～06

手順（2） ⇔ 行番号 07～10

手順（3） ⇔ 行番号 11

のような明確な対応があります。したがって、問題文の記述とプログラムを照合すれば、空欄に必要な処理がわかるようになっています。



12

次のプログラム中の a と b に入る正しい答えの組合せを、

解答群の中から選べ。ここで、配列の要素は 1 から始まる。

ある学年の生徒全員のテストの点数が、整数型の配列 `points` に格納されている。テストは 100 点満点である。点数の分布を見るために、ヒストグラムを作成するプログラムである。ヒストグラムの階級は、“0 点以上 10 点未満”, “10 点以上 20 点未満”, …, “90 点以上” の 10 個に区分し、階級ごとに人数分の個数の “*” を出力する。生徒数が 150 人の場合の出力結果の例を図に示す。

```
0 ~ 9 : *
10 ~ 19 : **
20 ~ 29 : ***
30 ~ 39 : ****
40 ~ 49 : *****
50 ~ 59 : *****
60 ~ 69 : *****
70 ~ 79 : *****
80 ~ 89 : *****
90 ~    : *****
```

図 出力結果の例

注：階級はプログラムでは出力しない。

[プログラム]

```
整数型の配列 : points /* 生徒全員の点数が格納されている */
整数型の配列 : count ← {0,0,0,0,0,0,0,0,0,0}

for (i を 1 から points の要素数 まで 1 ずつ増やす)
    if (points[i] が  a)
        num ← 10
    else
        num ←  b
    endif
    count[num] ← count[num] + 1
endfor

for (j を 1 から count の要素数 まで 1 ずつ増やす)
    count[j] 個の "*" を出力する
endfor
```

解答群

	a	b
ア	100 以下	(points[i] ÷ 10 の商) + 1
イ	100 以下	(points[i] ÷ 10 の商) - 1
ウ	100 に等しい	(points[i] ÷ 10 の商)
エ	100 に等しい	(points[i] ÷ 10 の商) + 1
オ	100 以上	(points[i] ÷ 10 の商) - 1
カ	100 以上	(points[i] ÷ 10 の商)

※オリジナル問題

プログラムの解説

```

01 整数型の配列： points /* 生徒全員の点数が格納されている */
02 整数型の配列： count ← {0,0,0,0,0,0,0,0,0,0}
03 for (i を 1 から points の要素数 まで 1 ずつ増やす)
04   if (points[i] が  )
05     num ← 10
06   else
07     num ← 
08   endif
09   count[num] ← count[num] + 1
10 endfor
11 for (j を 1 から count の要素数 まで 1 ずつ増やす)
12   count[j] 個の "*" を出力する
13 endfor

```

プログラムは、0 以上 100 以下の範囲のデータを，“0 点以上 10 点未満”，“10 点以上 20 点未満”，…，“90 点以上” の 10 個に区分して集計します。集計は、

データが 0 以上 10 未満なら count[1] を 1 増やす
 データが 10 以上 20 未満なら count[2] を 1 増やす
 :
 のように、配列 count の該当する要素ごとに行います。たとえばデータが 75 点であれば、集計先の要素番号は 8 になります。これは、

データ ÷ 10 の商 + 1

で計算できます。

ただしこの計算では、データが 100 点満点の場合の要素番号が 11 になってしまいます。そのため、データが 100 の場合は例外として、要素番号 10 に集計するように調整する必要があります。

問題解説

行番号 04 : if 文の条件式が真の場合、集計先の配列 count の要素番号を 10 にします。これは、データが 100 点満点だった場合の例外措置です。したがって、空欄 a には「100 に等しい」が入ります。

```
04   if (points[i] が 100 に等しい)
05       num ← 10 ← 100 点満点の場合は要素番号10に集計
```

空欄 a

行番号 07 : データが 100 点満点以外の場合は、「データの 10 の位の数 + 1」が集計先の要素番号になります。この値は次のように計算できます。

```
07   num ← (point[i] ÷ 10 の商) + 1
```

空欄 b

以上から、空欄 a が「100 に等しい」、空欄 b が「(points[i] ÷ 10 の商) + 1」の組合せの **エ** が正解です。

解答

問12 エ



合格のカギ

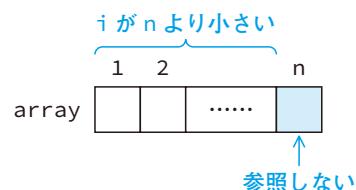
端の数

プログラミングで、端の数をどう処理するかはデリケートな問題です。条件式では「以上」とするか「より大きい」とするのか、配列の要素番号に +1 や -1 が必要かどうかなどに注意しましょう。

とくに擬似言語では、配列の要素番号が 1 から始まる場合が多いため注意が必要です。

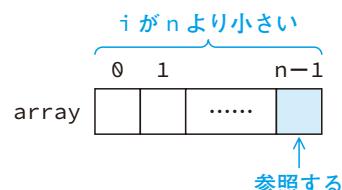
●配列の要素番号が 1 から始まる場合

```
n ← array の要素数
i ← 1
X while (i が n より小さい)
    array[i] を参照
    i ← i+1
endwhile
```



●配列の要素番号が 0 から始まる場合

```
n ← array の要素数
i ← 0
O while (i が n より小さい)
    array[i] を参照
    i ← i+1
endwhile
```





問 13

次の記述中の [] に入る正しい答えを、解答群の中から選べ。

関数 `nuron` は、2つの引数 x と y の値に応じて、0 または 1 を返す。引数と戻り値の組合せは、図のとおりである。

x	y	戻り値
0	0	
0	1	
1	0	
1	1	

図 引数と戻り値の組合せ

〔プログラム〕

```

○ 整数型 : nuron( 整数型 :x, 整数型 : y)
    実数型 : tmp, bias
    実数型の配列 : w ← {0.6, 0.6}
        bias ← -0.7
        tmp ← x × w[1] + y × w[2] + bias
        if (tmp > 0.0)
            return 0
        else
            return 1
        endif
    
```

解答群

ア

0
0
0
1

イ

0
1
1
1

ウ

1
1
1
0

エ

1
0
0
0

オ

0
1
1
0

※オリジナル問題

プログラムの解説

```

01 ○ 整数型 : nuron( 整数型 :x, 整数型 : y)
02     実数型 : tmp, bias
03     実数型の配列 : w ← {0.6, 0.6}
04     bias ← -0.7
05     tmp ← x × w[1] + y × w[2] + bias
06     if (tmp > 0.0)
07         return 0
08     else
09         return 1
10    endif
    
```

パーセプトロン

関数 `nuron` は、**パーセプトロン**（単純パーセプトロン）を用いて論理回路を実装したものです。

パーセプトロンは、機械学習で用いられる脳の神経回路をモデルとした回路で、複数の入力信号を受け取り、1つの信号を出力します。

パーセプトロンで出力信号を得るには、まず入力信号に重みを掛けて足し合わせ、**バイアス**を加えた値を計算します。本問では、2つの入力信号に対応する重みは実数型の配列 `w` に、バイアスは変数 `bias` に格納されているので、次のようにになります。

```
tmp ← x × w[1] + y × w[2] + bias
```

関数 `nuron` では、変数 `tmp` に格納された値が 0.0 より大きい場合は 0、0.0 以下の場合は 1 を出力信号とします。この 0.0 を**閾値**といいます。

問題解説

引数に指定した入力信号ごとに、変数 `tmp` の値を計算してみましょう。`w = {0.6, 0.6}`, `bias = -0.7` より

x	y	tmp
0	0	$0 \times 0.6 + 0 \times 0.6 - 0.7 \rightarrow -0.7$
0	1	$0 \times 0.6 + 1 \times 0.6 - 0.7 \rightarrow -0.1$
1	0	$1 \times 0.6 + 0 \times 0.6 - 0.7 \rightarrow -0.1$
1	1	$1 \times 0.6 + 1 \times 0.6 - 0.7 \rightarrow 0.5$

以上から、変数 `tmp` の値は $x = 1, y = 1$ のときのみ 0 より大きく、それ以外の場合は 0 以下となります。したがって出力信号（戻り値）は、次のようにになります。

x	y	戻り値
0	0	1
0	1	1
1	0	1
1	1	0

以上から、正解は **ウ** です。

解答

問13 ウ



合格のカギ

重みの調整

単純パーセプトロンでは、重み `w` の値やバイアス `bias` の値を調整することによって、入力信号に対応する出力のパターンを変化させることができます。関数 `nuron` の配列 `w` と `bias` の値をそれぞれ次のようにすると、AND 回路や OR 回路がつくれます（ただし、XOR 回路はつくれません）。

AND 回路：

```
w ← {-0.5, -0.5}  
bias ← 0.7
```

OR 回路：

```
w ← {-0.5, -0.5}  
bias ← 0.3
```

 **問 14**

次のプログラム中の [] に入る正しい答えを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まる。

複数の項目をカンマ（“,”）で区切って並べたレコードがある。手続 `comma2space` は、引数に指定されたレコードを読み込み、カンマ（“,”）を 1 文字以上のスペースに変換して出力する。スペースの文字数は、レコードの先頭を 1 桁目として、次の項目の 1 文字目が (10 の倍数 + 1) 桁目になるよう調整するものとする。引数に指定したレコードと出力結果の例を図に示す。各項目の文字数は 9 文字以下とする。

入力レコード:	<code>[F E - 1 0 0 1 , W H I T E , 1 2 3 , Y]</code>								
出力結果 :	<table border="0"> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">11</td> <td style="text-align: center;">21</td> <td style="text-align: right; vertical-align: bottom;">31(桁目)</td> </tr> <tr> <td><code>[F E - 1 0 0 1 - - - W H I T E - - - - 1 2 3 - - - - - - - - - Y </code></td> <td></td> <td></td> <td></td> </tr> </table>	1	11	21	31(桁目)	<code>[F E - 1 0 0 1 - - - W H I T E - - - - 1 2 3 - - - - - - - - - Y </code>			
1	11	21	31(桁目)						
<code>[F E - 1 0 0 1 - - - W H I T E - - - - 1 2 3 - - - - - - - - - Y </code>									

図 入力レコードと出力結果の例

〔プログラム〕

```

○ comma2space( 文字列型 : rec )
文字列型 : str ← ""
整数型 : count ← 1, i
for (i を 1 から rec の要素数まで 1 ずつ増やす )
    if (rec の i 番目の文字が "," に等しい )
        do
            str の末尾に " " を追加する
            count ← count + 1
        while ([ ])
    else
        str の末尾に rec の i 番目の文字を追加する
        count ← count + 1
    endif
endfor
str を出力する

```

解答群

- ア (count mod 10) が 0 に等しい
- イ (count mod 10) が 0 に等しくない
- ウ (count mod 10) が 0 以上
- エ (count mod 10) が 1 に等しい
- オ (count mod 10) が 1 に等しくない
- カ (count mod 10) が 1 以上

※オリジナル問題

プログラムの解説

```
01 ○ comma2space( 文字列型 : rec )
02 文字列型 : str ← ""
03 整数型 : count ← 1, i
04 for (i を 1 から rec の要素数まで 1 ずつ増やす )
05   if (rec の i 番目の文字が "," に等しい )
06     do
07       str の末尾に " " を追加する
08       count ← count + 1
09     while ( [ ] )
10   else
11     str の末尾に rec の i 番目の文字を追加する
12     count ← count + 1
13   endif
14 endfor
15 str を出力する
```

カンマ区切りのレコードを、固定長（10 行）のスペース区切りに変換して出力するプログラムです。カンマ（“,”）を読み込んだら、そのときの行位置に応じて 1 文字以上のスペースを出力し、次の項目の先頭が 10 の倍数 + 1 行目になるように調整します。

問題解説

プログラムは、現在の行数を変数 count で管理しており、スペース文字を出力（行番号 07）したときや、項目の文字を出力（行番号 12）したときに 1 加算します。

1 つの項目を出力した後、次の項目を出力するには、変数 count の値がちょうど 10 の倍数になるまでスペースを出力します。この処理を行っているのが次の部分です。

```
06 do
07   str の末尾に " " を追加する
08   count ← count + 1
09   while ( [ ] )
```

たとえば、10 行目（count = 10）にスペースを出力すると、行番号 08 で count に 1 が加算され、11 行目になります。11 行目には次の項目の先頭文字を出力するので、繰返し処理を抜けます。この判定は、変数 count を 10 で割った余りが 1 かどうかで行うことができます。

変数 count を 10 で割った余りが 1 でなければ、繰返しを継続します。do ~ while の条件式には、繰返しの継続条件を記述するので、次のようにになります。

```
09 while (count mod 10 が 1 に等しくない)
```

問 14 オ

以上から、正解は オ です。



合格のカギ

文字列データ

文字列データは、文字列型のインスタンスとして扱う場合と、文字型の配列として扱う場合があります。文字列を 1 文字ずつ処理する場合は、文字型の配列として考えます。

 **問 15**

次の記述中の **a** と **b** に入れる正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まるものとする。

関数 `median` は、引数として与えられた配列に格納された整列済みデータの中央値（メディアン）を返す。関数 `median` を `median({5, 5, 7, 7, 7, 9, 11, 13, 13})` として呼び出したときの戻り値は 7 となる。また、関数 `median` を `median({4, 5, 5, 6, 7, 9, 9, 10, 11, 12})` として呼び出したときの戻り値は 8 となる。

〔プログラム〕

```

○実数型 : median( 実数型の配列 : data)
    整数型 : k
    実数型 : ret
    k ← a
    if ((data の要素数 mod 2) が 0 に等しい)
        ret ← b /* 実数値 */
    else
        ret ← data[k]
    endif
    return ret

```

解答群

	a	b
ア	(data の要素数 ÷ 2 の商)	(data[k] + data[k + 1]) ÷ 2
イ	(data の要素数 ÷ 2 の商)	(data[k - 1] + data[k]) ÷ 2
ウ	(data の要素数 ÷ 2 の商) + 1	(data[k] + data[k + 1]) ÷ 2
エ	(data の要素数 ÷ 2 の商) + 1	(data[k - 1] + data[k]) ÷ 2
オ	(data の要素数 ÷ 2 の商) - 1	(data[k] + data[k + 1]) ÷ 2
カ	(data の要素数 ÷ 2 の商) - 1	(data[k - 1] + data[k]) ÷ 2

※オリジナル問題


合格の力ギ

実際の値をあてはめて考えよう

配列の真ん中の要素番号は、おおまかに「要素数 ÷ 2」で求められることはすぐわかりますが、+ 1 が必要かどうかは場合によります（要素番号が 1 から始まる場合は必要）。要素が 5 個のとき、9 個のとき…のように、具体的な値を当てはめて確認しましょう。

```

01 ○ 実数型 : median( 実数型の配列 : data)
02 整数型 : k
03 実数型 : ret
04   k ← [a]
05   if ((data の要素数 mod 2) が 0 に等しい )
06     ret ← [b] /* 実数値 */
07   else
08     ret ← data[k]
09   endif
10   return ret

```

[中央値とは]

複数のデータの代表となる値として、平均値、中央値、最頻値などがあります。このうち**中央値**（メディアン）とは、データを昇順または降順に整列したときに中央に位置する値のことです。ただし、データの個数が偶数の場合は、中央に位置する値が2個あるため、それらの平均値を中央値とします。

中央値の計算は、データがあらかじめ整列済みであれば比較的簡単です。まず、データの個数が偶数かどうかを調べ、偶数であれば真ん中の2個の値の平均値を求めます（行番号06）。データの個数が奇数でなければ、真ん中の値をそのまま返すだけです。

問題解説

行番号04：行番号04では、配列の真ん中の要素の要素番号を計算します。たとえば、配列の要素数が5個の場合、真ん中の要素の要素番号は3です。これは、 $5 \div 2$ の商2に1を足して求めます。したがって、空欄aは次のようにになります。

04 k ← (data の要素数 ÷ 2) の商+1

要素番号が0から始まる場合は、「+1」は不要だよ。



行番号06：行番号06では、データの個数が偶数だった場合の中央値を求めます。

たとえば、配列の要素数が10個の場合、真ん中の要素の要素番号は5と6です。行番号04で求めた変数kには、 $10 \div 2 + 1 = 6$ が格納されているので、真ん中の要素はdata[k-1]とdata[k]となります。この2つの要素の値の平均値を中央値とします。

06 ret ← (data[k-1] + data[k]) ÷ 2

以上から、空欄aが「(data の要素数 ÷ 2) の商+1」、空欄bが「(data[k-1] + data[k]) ÷ 2」の組合せの**エ**が正解です。

○ 解答 ○

問15 エ



16

次の記述中の ~ に入れる正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まる。

要素の多くが 0 の行列を疎行列という。次のプログラムは、二次元配列に格納された行列のデータ量を削減するために、疎行列の格納に適したデータ構造に変換する。

関数 `transformSparseMatrix` は、引数 `matrix` で二次元配列として与えられた行列を、整数型配列の配列に変換して返す。関数 `transformSparseMatrix` を `transformSparseMatrix({{3,0,0,0,0}, {0,2,2,0,0}, {0,0,0,1,3}, {0,0,0,2,0}, {0,0,0,0,1}})` として呼び出したときの戻り値は、`{ { }, { }, { } }` である。

〔プログラム〕

○ 整数型配列の配列 : `transformSparseMatrix(整数型の二次元配列 : matrix)`

整数型 : `i, j`

整数型配列の配列 : `sparseMatrix`

```
sparseMatrix ← {{}, {}, {}} /* 要素数 0 の配列を三つ要素にもつ配列 */
for (i を 1 から matrix の行数 まで 1 ずつ増やす)
    for (j を 1 から matrix の列数 まで 1 ずつ増やす)
        if (matrix[i, j] が 0 でない)
            sparseMatrix[1] の末尾 に i の値 を追加する
            sparseMatrix[2] の末尾 に j の値 を追加する
            sparseMatrix[3] の末尾 に matrix[i, j] の値を追加する
    endif
endfor
endfor
return sparseMatrix
```

解答群

	a	b	c
ア	1,2,2,3,3,4,5	1,2,3,4,5,4,5	3,2,2,1,2,3,1
イ	1,2,2,3,3,4,5	1,2,3,4,5,4,5	3,2,2,1,3,2,1
ウ	1,2,3,4,5,4,5	1,2,2,3,3,4,5	3,2,2,1,2,3,1
エ	1,2,3,4,5,4,5	1,2,2,3,3,4,5	3,2,2,1,3,2,1

※サンプル問題より

```

01 ○ 整数型配列の配列 : transformSparseMatrix( 整数型の二次元配列 : matrix )
02 整数型 : i, j
03 整数型配列の配列 : sparseMatrix
04 sparseMatrix ← {{}, {}, {}}
05 for (i を 1 から matrix の行数 まで 1 ずつ増やす)
06   for (j を 1 から matrix の列数 まで 1 ずつ増やす)
07     if (matrix[i, j] が 0 でない)
08       sparseMatrix[1] の末尾に i の値 を追加する
09       sparseMatrix[2] の末尾に j の値 を追加する
10       sparseMatrix[3] の末尾に matrix[i, j] の値 を追加する
11     endif
12   endfor
13 endfor
14 return sparseMatrix

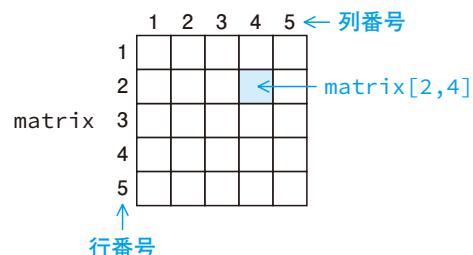
```

二次元配列

右図のように、データを入れる区画を行方向と縦方向に格子状に設けた配列を、**二次元配列**といいます。擬似言語では、二次元配列の行列を

配列名 [行番号 , 列番号]

のように指定します。

**疎行列の圧縮表現**

二次元配列を扱うには、データが格納されているかどうかに関わりなく、行数×列数の記憶領域が必要になります。そのため、要素の多くが0の疎行列の場合には、無駄な領域が多くなってしまいます。そこで本問では、データ構造を次のように変換して、データ量を圧縮しています。

```

07   if (matrix[i, j] が 0 でない)
08     sparseMatrix[1] の末尾に i の値 を追加する } 行番号と列番号を登録
09     sparseMatrix[2] の末尾に j の値 を追加する
10     sparseMatrix[3] の末尾に matrix[i, j] の値 を追加する ← 値を登録
11   endif
12 endfor

```

二次元配列の0でない1個の要素は、sparseMatrixの3個の要素に変換されます。そのため、0でない要素が多くなると、かえってデータ量が増えてしまいます。たとえば、5行5列の二次元配列の場合は、0でない個数が9個以上になると、sparseMatrixのほうがデータ量が多くなります。

問題解説

本問の二次元配列 `matrix` の内容は、次のとおりです。

	1	2	3	4	5
1	3	0	0	0	0
2	0	2	2	0	0
3	0	0	0	1	3
4	0	0	0	2	0
5	0	0	0	0	1

この配列を `sparseMatrix` に変換します。なお、プログラムが `matrix` の要素を調べる順番に注意しましょう。0でない要素は次の7個です。

データの格納場所	値
<code>matrix[1, 1]</code>	3
<code>matrix[2, 2]</code>	2
<code>matrix[2, 3]</code>	2
<code>matrix[3, 4]</code>	1
<code>matrix[3, 5]</code>	3
<code>matrix[4, 4]</code>	2
<code>matrix[5, 5]</code>	1

`sparseMatrix[1]` と `sparseMatrix[2]` には、上記のうち、`matrix` の行番号（*i*の値）と列番号（*j*の値）が順に格納されます。

```
sparseMatrix[1] : {1, 2, 2, 3, 3, 4, 5}
sparseMatrix[2] : {1, 2, 3, 4, 5, 4, 5}
```

また、`sparseMatrix[3]` には、対応する `matrix` の要素の値が順に格納されます。

```
sparseMatrix[3] : {3, 2, 2, 1, 3, 2, 1}
```

以上から、正解は **イ** です。

○ 解答 ○

問16 イ



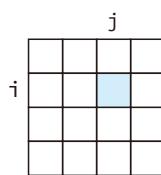
合格のカギ

二次元配列と配列の配列

二次元配列は、一般的なプログラム言語では「配列の配列」として表しますが、情報処理試験の擬似言語では二次元配列と配列の配列を区別しています。二次元配列の要素は `matrix[i, j]` のように表しますが、配列の配列は `matrix[i][j]` となることに注意してください。また、擬似言語の配列の配列では、要素数が異なる配列の配列もつくれます。

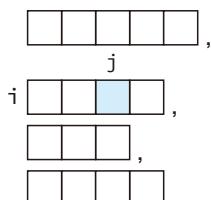
● 二次元配列

`matrix[i, j]`



● 配列の配列

`matrix[i][j]`



要素数が
異なって
いてもよい



17

次の記述を読んで、図1中の項番(一)～(三)それぞれに対処する組織の適切な組合せを、解答群の中から選べ。

製造業のA社では、ECサイト（以下、A社のECサイトをAサイトという）を使用し、個人向けの製品販売を行っている。Aサイトは、A社の製品やサービスが検索可能で、ログイン機能を有しており、あらかじめAサイトに利用登録した個人（以下、会員という）の氏名やメールアドレスといった情報（以下、会員情報という）を管理している。Aサイトは、B社のPaaSで稼働しており、PaaS上のDBMSとアプリケーションサーバを利用している。

A社は、Aサイトの開発、運用をC社に委託している。A社とC社との間の委託契約では、Webアプリケーションプログラムの脆弱性対策は、C社が実施するとしている。

最近、A社の同業他社が運営しているWebサイトで脆弱性が悪用され、個人情報が漏えいするという事件が発生した。そこでA社は、セキュリティ診断サービスを行っているD社に、Aサイトの脆弱性診断を依頼した。脆弱性診断の結果、対策が必要なセキュリティ上の脆弱性が複数指摘された。図1にD社からの指摘事項を示す。

- (一) Aサイトで利用しているDBMSに既知の脆弱性があり、脆弱性を悪用した攻撃を受けるおそれがある。
- (二) Aサイトで利用しているアプリケーションサーバのOSに既知の脆弱性があり、脆弱性を悪用した攻撃を受けるおそれがある。
- (三) ログイン機能に脆弱性があり、Aサイトのデータベースに蓄積された情報のうち、会員には非公開の情報を閲覧されるおそれがある。

図1 D社からの指摘事項

解答群

	(一)	(二)	(三)
ア	A社	A社	A社
イ	A社	A社	C社
ウ	A社	B社	B社
エ	B社	B社	B社
オ	B社	B社	C社
カ	B社	C社	B社
キ	B社	C社	C社
ク	C社	B社	B社
ケ	C社	B社	C社
コ	C社	C社	B社

※サンプル問題より

問題解説

脆弱性診断の結果、A サイトには 3 つの脆弱性（セキュリティ上の欠点）が見つかりました。これらの脆弱性に対処すべき組織を、A サイトに関係のある A 社、B 社、C 社からそれぞれ選択する問題です。3 つの脆弱性をひとつずつ検討していきましょう。

- (一) A サイトで利用している DBMS に既知の脆弱性があり、脆弱性を悪用した攻撃を受けるおそれがある。

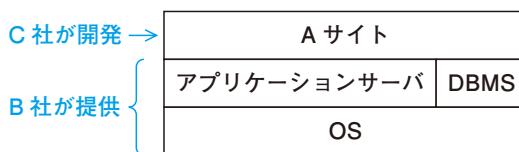
A サイトは B 社の **PaaS** で稼働しており、PaaS 上の DBMS とアプリケーションサーバを使用しています。

PaaS とは、Platform as a Service の略で、アプリケーションを実行するために必要な環境（プラットフォーム）を、インターネットを介して利用者に提供するサービスです。

具体的には、OS や DBMS（データベース管理システム）、アプリケーションサーバなど、自社サイトを構築するために必要なシステムを提供します。指摘事項（一）は DBMS の脆弱性であり、DBMS は B 社が提供するプラットフォームの一部ですから、B 社が対処すべきです。

- (二) A サイトで利用しているアプリケーションサーバの OS に既知の脆弱性があり、脆弱性を悪用した攻撃を受けるおそれがある。

アプリケーションサーバは B 社が提供するプラットフォームの一部であり、OS はそのアプリケーションサーバが稼働する基盤ですから、OS の脆弱性については B 社が対処すべきです。



- (三) ログイン機能に脆弱性があり、A サイトのデータベースに蓄積された情報のうち、会員には非公開の情報を閲覧されるおそれがある。

ログイン機能は、A サイトの機能の一部です。A 社は C 社と委託契約を結んでおり、A サイトの Web アプリケーションプログラムの脆弱性対策は C 社が実施することになっています。したがって、ログイン機能の脆弱性は C 社が対処しなければなりません。

以上から、(一) と (二) が B 社、(三) が C 社の **オ** が正解です。

解答

問 17 オ



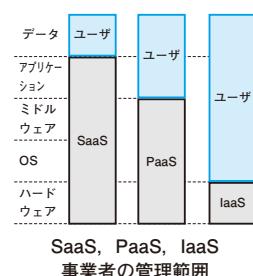
SaaS, PaaS, IaaS

SaaS、PaaS、IaaS の違いを理解しておきましょう。

SaaS (Software as a Service) : アプリケーションの機能を利用者に提供

PaaS (Platform as a Service) : アプリケーションの稼働環境を利用者に提供

IaaS (Infrastructure as a Service) : システムを稼働する回線やハードウェア環境のみを提供



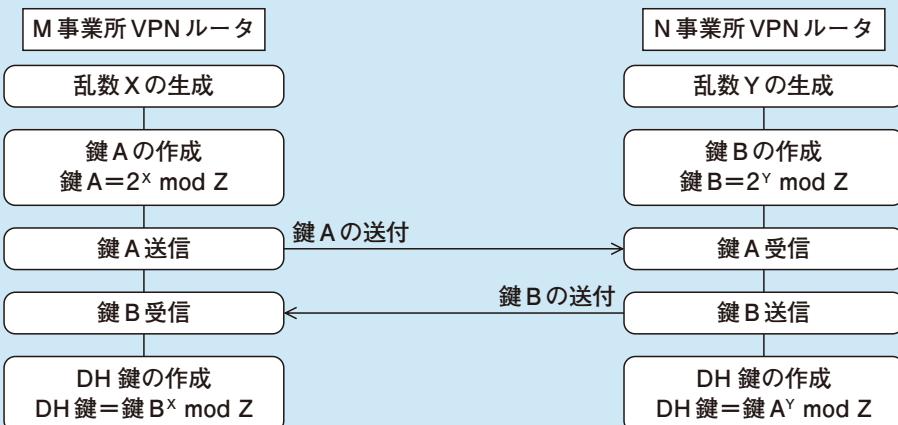


18

次の記述中の [] に入る正しい答えを、解答群の中から選べ。

A 社は、関東の N 事業所で利用している営業支援システムを、関西の M 事業所でも利用するために、IPsec を利用した VPN の導入を検討している。

VPN の実現には、VPN ルータを利用する。IPsec では、VPN ルータ間で暗号化に利用する鍵を端末間で安全に交換する仕組みの一つとして、Diffie-Hellman 鍵交換法（以下、DH 法という）を利用している。DH 法の例を図に示す。DH 法で作成された鍵（以下、DH 鍵という）を暗号化に利用する。



注記 1 X, Y は正の整数とする。

注記 2 2^X は、2 の X 乗を示す。

注記 3 $P \bmod Q$ は、P の Q による剰余を示す。

注記 4 Z は、M 事業所 VPN ルータ、N 事業所 VPN ルータに事前に設定された素数である。

図 DH 法の例

Z = 11, X = 7, Y = 5 の場合、DH 鍵は [] である。

解答群

- ア 2
- イ 5
- ウ 7
- エ 10
- オ 13

※出典：基本情報技術者試験平成 25 年度秋期午後問題問 4 を改変

問題解説

IPsec は、インターネットでやり取りするデータを暗号技術を用いて安全に送受信するためのプロトコルで、IP パケットの暗号化や認証、改ざん防止などの機能を備えています。**VPN** (Virtual Private Network : 仮想専用通信網) は、遠隔地にあるコンピュータ間の専用ネットワークを公衆回線を利用して実現するもので、インターネットによる VPN では IPsec がよく利用されています。

データの暗号化には**公開鍵暗号方式**と**共通鍵暗号方式**がありますが、VPN では一般に暗号化と復号の処理が高速な**共通鍵暗号方式**が用いられます。共通鍵暗号方式では、データの送信側と受信側とで、事前に共通鍵の受け渡しが必要になるため、共通鍵の送受信をどのように安全に行うかが問題になります。そのための方法のひとつが **Diffie-Hellman 鍵交換法** (DH 法) です。

DH 法を使用すると、共通鍵そのものをいちども送受信することなく、送信側と受信側とで同一の共通鍵を作成することができます。

問題文の図にしたがって、 $Z = 11$, $X = 7$, $Y = 5$ のときの DH 鍵を作成してみましょう。なお、DH 鍵は、鍵 A からつくる方法と鍵 B からつくる方法の 2 通りがあります。どちらを使っても同じ値になりますが、ここでは鍵 B からつくる方法で説明します。

- ① 「鍵 $B = 2^Y \bmod Z$ 」に、 $Y = 5$, $Z = 11$ を代入します。

$$\text{鍵 } B = 2^5 \bmod 11 = 32 \bmod 11 \quad \leftarrow 32 \div 11 \text{ の余り}$$

以上から、鍵 $B = 10$ となります。

- ② 「DH 鍵 = 鍵 $B^X \bmod Z$ 」に、鍵 $B = 10$, $X = 7$, $Z = 11$ を代入します。

$$\text{DH 鍵} = 10^7 \bmod 11 = 10000000 \bmod 11$$

$10000000 \div 11 = 909090$ 余り 10 より、DH 鍵 = 10 となります。以上から、正解は **工** です。

解答

問18 工



合格の力ギ

共通鍵暗号方式と公開鍵暗号方式

共通鍵暗号方式と公開鍵暗号方式については、科目 A でもよく出題される基本事項です。科目 B ではどのように出題されるかわかりませんが、しくみを理解しておきましょう。

● 共通鍵暗号方式

暗号化と復号に同一の鍵（共通鍵）を用いる方式。公開鍵暗号方式に比べると処理は高速ですが、暗号の送信元と受信先とで、あらかじめ鍵の受け渡しを行う必要があります。

鍵を安全にやり取りするため、鍵のやり取りだけを公開鍵暗号方式で行ったり、DH 法を使って鍵を生成する方法が考案されています。

● 公開鍵暗号方式

暗号化用の鍵と復号用が異なる方式。暗号の受信者は、あらかじめ暗号化用の鍵（公開鍵）と復号用の鍵（秘密鍵）のペアを用意しておき、公開鍵だけを送信元に渡します。送信元は、受信者の公開鍵を使ってメッセージを暗号化し、受信者は手元にある秘密鍵を使って暗号を復号します。

暗号化用の鍵をやり取りしないので共通鍵暗号方式より安全ですが、一般に暗号化・復号の処理に時間がかかります。



19

次の記述中の [a] と [b] に入る適切な答えの組合せを、解答群の中から選べ。

A 社の Web サーバにログインするには、ID とパスワードが必要である。このたび A 社では、パスワードの強度（パスワードの候補数）の強化を検討することになった。これまでパスワードは英小文字 26 文字だけを受け付け、長さは 6 文字だった。これに対し、他の 3 通りのパスワードの強度を比較した。その比較結果を表に示す。

表 パスワードの強度比較

パスワードとして受け付ける文字種と長さ	強度の比較
(a) 英小文字、6 文字（従来の設定）	—
(b) 英小文字、8 文字	(a) と比較して [a] 倍
(c) 英大文字・英小文字、8 文字	(b) と比較して [b] 倍
(d) 英大文字・英小文字・数字・記号、8 文字	(c) と比較してさらに多い

解答群

	a	b
ア	26	2×8
イ	26	26
ウ	26	2^8
エ	2×26	2×8
オ	2×26	26
カ	2×26	2^8
キ	2×26	26^8
ク	26^2	2×8
ケ	26^2	2^8
コ	26^2	26^8

※出典：基本情報技術者試験平成 28 年度春期午後問題問 1 を改変

問題解説

パスワードの強度（パスワードの候補数）は、一般に

文字の種類の数^{文字数}

で求めることができます。

空欄 a：英小文字（a～z）は26種類あるので、長さ6文字では $26 \times 26 \times 26 \times 26 \times 26 \times 26 = 26^6$ 通りのパスワードをつくれます。英小文字（a～z）8文字では、 $26 \times 26 = 26^8$ 通りになります。長さ8文字のパスワードの種類は、長さ6文字のパスワードの種類に 26^2 を掛けたものですから、 26^2 倍となります。

空欄 b：英小文字（a～z）と英大文字（A～Z）を合わせると、文字の種類は全部で 26×2 種類になります。長さ8文字では、

$$(26 \times 2)^8 = 26^8 \times 2^8 \text{ 通り}$$

のパスワードがつくれます。（b）の 26^8 通りと比較すると、 2^8 倍となります。

以上から、**空欄 a**が「 26^2 」、**空欄 b**が「 2^8 」の **ケ** が正解です。

なお、パスワードの候補数が多くても、推測しやすい文字列をパスワードに設定していれば強度は高くなりません。大文字と小文字、数字、記号を混ぜる、意味のある単語にしないといった規則を必要に応じて決める必要があります。

○ 解答

問19 ケ



合格のカギ

パスワード

パスワード関連では、ワンタイムパスワードやハッシュ値についても理解しておきましょう。

●ワンタイムパスワード

1回限りの使い捨てのパスワード。タイムスタンプ認証は、利用者のパスワードと時刻をもとに、その時点のパスワードを生成する方式。チャレンジレスポンス認証は、サーバが利用者にランダムなデータ（チャレンジ）を送信し、利用者はチャレンジとパスワードからハッシュ値（レスポンス）を生成してサーバに送信する方式です。

●ハッシュ値

元のデータから決まった手順で生成される固定長のデータのこと。一般に、ハッシュ値から元のデータを推測することは非常に困難です。

サーバ側が利用者のパスワードを管理する場合、パスワードそのものを保存しておくと盗難による漏えいのおそれがあります。そこで、パスワードからハッシュ値を生成して記録するようになると、パスワードの漏えいを防ぐことができます。利用者がログインしたときは、入力されたパスワードからハッシュ値を生成し、それを記録されているハッシュ値と比較して認証します。



20

販売支援システムの情報セキュリティに関する次の記述を読んで、設問に答えよ。

中堅の商社である A 社では、営業員が顧客先で営業活動を行い、自社に戻ってから見積書を作成している。この度、営業員がタブレット端末（以下、タブレットという）を携帯し、顧客先で要求を聞きながら、タブレットを使って見積書を作成し、その場で顧客に提示できる販売支援システムを構築することにした。

営業員は、タブレットの Web ブラウザからインターネット経由で HTTP over TLS（以下、HTTPS という）によって販売支援システムにアクセスする。このとき、営業員は、社員 ID とパスワードを入力してログインする。

〔販売支援システムの構成〕

- (1) 販売支援システムは次のサーバで構成され、A 社のネットワークに設置される。
 - ① リバースプロキシサーバ（以下、RP サーバという）1 台
 - ② アプリケーションソフトウェアが稼働する Web サーバ 2 台
 - ③ 見積書作成に必要なデータを格納するデータベースサーバ（以下、DB サーバという）1 台
- (2) Web サーバ 2 台はクラスタリング構成にして、1 台が故障してもサービスが継続できるようになる。
- (3) DB サーバへのアクセスの監視は、PC と同じ LAN にある監視サーバで行う。
- (4) インターネットから販売支援システムへの通信は、RP サーバを経由して行う。RP サーバは、HTTPS を HTTP に変換し、販売支援システムの他のサーバと、HTTP で通信する。

A 社のネットワーク構成を図 1 に示す。

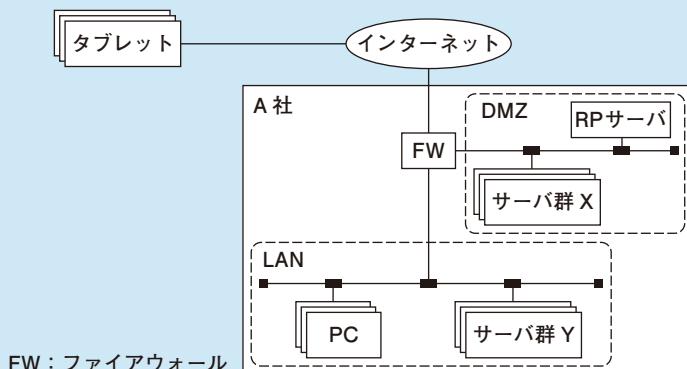


図 1 A 社のネットワーク構成

販売支援システムに関わる通信経路について、通信経路で利用するプロトコル及び宛先ポート

番号を表1に示す。また、FWにおけるフィルタリングの設定を表2に示す。

表1 通信経路で利用するプロトコル及び宛先ポート番号

通信経路	プロトコル	宛先ポート番号
監視サーバからDBサーバ	監視専用	1600
タブレットからRPサーバ	HTTPS	443
RPサーバからWebサーバ	HTTP	80
WebサーバからDBサーバ	DB専用	1552

表2 FWにおけるフィルタリングの設定

項目番	条件			動作
	送信元	宛先	宛先ポート番号	
1	任意	RPサーバ	443	許可
2	任意	DBサーバ	1552	許可
3	任意	任意	任意	拒否

注記 項番が小さいものから順に突き合わせ、最初に一致したものが適用される。

設問 WebサーバとDBサーバについて、図1中の配置場所の組合せとして適切な答えを、解答群の中から選べ。

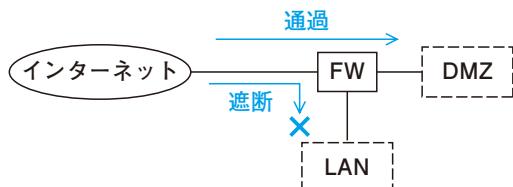
解答群

	Webサーバ	DBサーバ
ア	サーバ群X	サーバ群X
イ	サーバ群X	サーバ群Y
ウ	サーバ群Y	サーバ群X
エ	サーバ群Y	サーバ群Y

※出典：基本情報技術者試験平成28年度秋期午後問題問1を改変

問題解説

サーバ群XはDMZ、サーバ群YはLANに配置されています。DMZ (DeMilitarized Zone) とは「非武装地帯」のことです。外部とインターネットを介してやり取りするサーバ群を、内部ネットワーク(LAN)から切り離したもので、DMZを設けることで、外部から内部ネットワークへ直接アクセスする必要がなくなるので、外部から内部ネットワークへのアクセスはファイアウォール(FW)ですべて遮断できます。



Web サーバは DMZ、DB サーバは LAN 上に配置するのが常識的な構成ですが、この問題でもそのかを確認しておきましょう。

販売支援システムは、RP サーバ、Web サーバ、DB サーバで構成され「インターネットから販売支援システムへの通信は、RP サーバを経由して」行います。リバースプロキシサーバ（RP サーバ）とはインターネットとサーバとの間に設置して、外部からのアクセスを代理で受け付け、サーバに中継するサーバです。DB サーバへは Web サーバ上で稼働するアプリケーションからアクセスします。また、DB サーバへのアクセスは「PC と同じ LAN にある監視サーバ」によって監視されます。各通信経路で利用するプロトコルは、表 1 の「通信経路で利用するプロトコル及び宛先ポート番号」より、次の図のようになります。

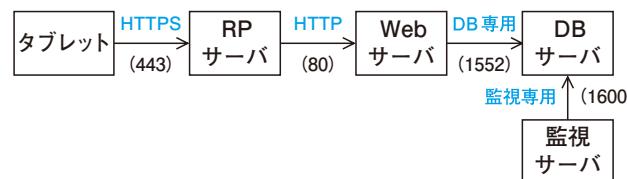


表 2 「FW におけるフィルタリングの設定」をみると、上の図の 4 つの通信経路のうち、ファイアウォールの通過が許可されているのは、宛先ポート番号 443、1552 の 2 つだけです。この 2 つはそれぞれ

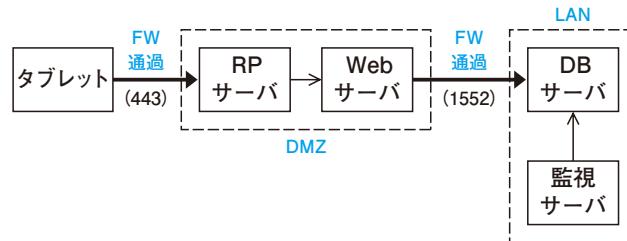
- ・タブレット→ RP サーバ (443)
- ・Web サーバ→ DB サーバ (1552)

です（表 1 参照）。他の通信はファイアウォールを通過できないので、

- ・RP サーバ→ Web サーバ (80)
- ・監視サーバ→ DB サーバ (1600)

は、同じゾーン内に配置しなければなりません。

RP サーバは DMZ にあるので、Web サーバも DMZ に配置します。また、監視サーバは LAN にあるので、DB サーバも LAN に配置します。



以上から、Web サーバの配置場所はサーバ群 X、DB サーバの配置場所はサーバ群 Y となります。正解は 1 です。

解答
問20 1



合格のカギ

セキュリティ対策の定石

情報システムのセキュリティ対策には、「このようなリスクにはこのような対策を講じる」といった定石がいくつかあります。基本情報技術者の試験では、定石からはずれた問題はまず出題されません。本問の DMZ やファイアウォールの設定も、多くの情報システムで採用されているものです。こうしたセキュリティ対策の定石をしっかりと押さえておきましょう。

基本情報技術者

予想問題②

• 科目A 110

(全 60 問 試験時間 : 90 分)

• 科目B 146

(全 20 問 試験時間 : 100 分)



基本情報技術者試験 予想問題②

科目A



問 01

1バイトのデータで0のビット数と1のビット数が等しいもののうち、符号なしの2進整数としてみたときに最大となるものを、10進整数として表したもののはどれか。

ア 120

イ 127

ウ 170

エ 240



問 02

16ビットの2進数 n を16進数の各桁に分けて、下位の桁から順にスタックに格納するために、次の手順を4回繰り返す。a, bに入る適切な語句の組合せはどれか。ここで、 $XXXX_{16}$ は16進数XXXXを表す。

[手順]

- (1) a を x に代入する。
- (2) x をスタックにプッシュする。
- (3) n を b 論理シフトする。

	a	b
ア	$n \text{ AND } 000F_{16}$	左に4ビット
イ	$n \text{ AND } 000F_{16}$	右に4ビット
ウ	$n \text{ AND } FFF0_{16}$	左に4ビット
エ	$n \text{ AND } FFF0_{16}$	右に4ビット



問 03

全体集合 S 内に異なる部分集合 A と B があるとき、 $\overline{A} \cap \overline{B}$ に等しいものはどれか。ここで、 $A \cup B$ は A と B の和集合、 $A \cap B$ は A と B の積集合、 \overline{A} は S における A の補集合、 $A - B$ は A から B を除いた差集合を表す。

ア $\overline{A} - B$

イ $(\overline{A} \cup \overline{B}) - (A \cap B)$

ウ $(S - A) \cup (S - B)$

エ $S - (A \cap B)$

解説

問01 2進数

1バイト=8ビットなので、0と1のビットがそれぞれ4つずつあれば、0と1のビット数が等しくなります。このような2進数で最大のものは11110000です。これを10進数に変換します。2進数を10進数に変換するには、2進数の各桁に重みをかけ、それらを足し合わせます。正解はエです。

合格のカギ

■ 40%


合格のカギ

桁(n)	7	6	5	4	3	2	1	0
2進数	1	1	1	1	0	0	0	0
重み(2^n)	128	64	32	16	8	4	2	1
	↓	↓	↓	↓	↓	↓	↓	↓
	128 + 64 + 32 + 16 + 0 + 0 + 0 + 0 = 240							

問01

注意 もし符号付き2進数だったら、左端のビットは符号ビットになるため、01111000(10進整数で120)が最大になるよ。

**問02 AND演算とシフト演算 キホン!**

80%

16進数の1桁は2進数の4桁分に相当するので、2進数nの下位から4桁ずつ順番に取り出し、スタックに格納します。

手順(1)は、2進数nの下位4桁をxに代入する処理です。空欄aにはAND演算が入ります。AND演算は、対応するビットが両方とも1のときだけ1になるので、次のようにすれば下位4桁だけを取り出せます。

このビットを取り出す

XXXX XXXX XXXX	XXXX	← 2進数n (Xは0または1)
AND 0000 0000 0000	1111	← 下位4桁が1、他は0
0000 0000 0000	XXXX	← nの下位4桁以外はすべて0になる

2進数00000000000001111を16進数で表すと000Fになります。以上から、空欄aには「n AND 000F₁₆」が入ります。

手順(3)は、nの桁をずらして、次に取り出す4桁を下位に移動する処理です。このような処理をシフト演算といいます。

次に取り出すビット

XXXX XXXX XXXX XXXX	XXXX	← 2進数n
0000 XXXX XXXX XXXX	XXXX	← 次の4桁が下位にくるように各桁をずらす

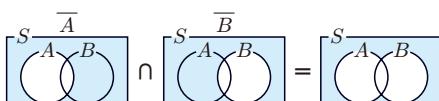
上図のように、各桁を「右に4ビット」シフトすれば、次に取り出す4桁が下位に移動します。

以上から、正解はイです。

問03 集合演算

20%

$A \cap B$ は、ベン図で表すと次のようになります。



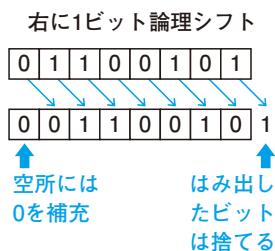
この結果と同様になるものを解答群から選びます。

○ ア - =

× イ - =

× ウ ∪ =

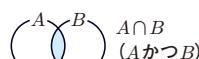
シフト演算 問02
ビット列を右または左にずらす演算。ずらした結果、はみ出たビットは捨てられる。論理シフトは、符号を考慮しない単純なシフト演算。符号を考慮するシフト演算を算術シフトという。


覚えよう!
問03

和集合といえば



積集合といえば



○ 解答

問01 イ 問02 イ
問03 ア



問 04

相関係数に関する記述のうち、適切なものはどれか。

- ア 全ての標本点が正の傾きをもつ直線上にあるときは、相関係数が +1 になる。
- イ 変量間の関係が線形のときは、相関係数が 0 になる。
- ウ 変量間の関係が非線形のときは、相関係数が負になる。
- エ 無相関のときは、相関係数が -1 になる。



問 05

3 台の機械 A, B, C が良品を製造する確率は、それぞれ 60%, 70%, 80% である。機械 A, B, C が製品を一つずつ製造したとき、いずれか二つの製品が良品で残り一つが不良品になる確率は何%か。

ア 22.4

イ 36.8

ウ 45.2

エ 78.8



問 06

標本化、符号化、量子化の三つの工程で、アナログをデジタルに変換する場合の順番として、適切なものはどれか。

- ア 標本化、量子化、符号化
- ウ 量子化、標本化、符号化

- イ 符号化、量子化、標本化
- エ 量子化、符号化、標本化



問 07

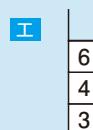
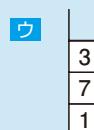
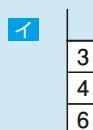
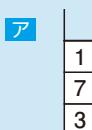
次の二つのスタック操作を定義する。

PUSH n : スタックにデータ（整数値 n ）をプッシュする。

POP : スタックからデータをポップする。

空のスタックに対して、次の順序でスタック操作を行った結果はどれか。

PUSH 1 → PUSH 5 → POP → PUSH 7 → PUSH 6 → PUSH 4 → POP
→ POP → PUSH 3



解説

問 04 相関係数

相関係数は、2 変量間の相関の度合いを表した数値です。相関係数の値は -1 から 1 の範囲で、**正の相関**のときはプラス、**負の相関**のときはマイナスになります。また、相関の度合いが強いほど ±1 に近くなり、**無相関**のときは 0 になります。

合格のカギ

20%

- ア 正解です。相関の度合いは、標本点がすべて直線上にあるとき最大となります。また、直線が正の傾きの場合は正の相関となるので、相関係数は +1 になります。
- × イ 変量間の関係が線形（一次式で表せる）ときは、相関があると考えられるので、相関係数は 0 にはなりません。
- × ウ 相関係数が負になるときは、負の相関（直線の傾きが負）がある場合です。変量間の関係は、相関があるとき線形になります。
- × エ 無相関のときは、相関係数は 0 になります。

合格のカギ

覚えよう！ 問04

相関係数といえば

- 正の相関のとき : $0 < r \leq +1$
- 無相関のとき : 0
- 負の相関のとき : $0 > r \geq -1$

問05 1つだけ不良品になる確率

■ 20%

2つが良品で、残り1つが不良品になるケースには、① A が不良品で、B, C が良品、② B が不良品で、A, C が良品、③ C が不良品で、A, B が良品の3通りあります。それぞれの確率は次のようにになります。

①機械 A が不良品で、B, C が良品となる確率

$$(1 - 0.6) \times 0.7 \times 0.8 = 0.224$$

②機械 B が不良品で、A, C が良品となる確率

$$0.6 \times (1 - 0.7) \times 0.8 = 0.144$$

③機械 C が不良品で、A, B が良品となる確率

$$0.6 \times 0.7 \times (1 - 0.8) = 0.084$$

2つが良品で、残り1つが不良品になる確率は、①, ②, ③のいずれかの確率なので、 $①+②+③=0.452=45.2\%$ となります。正解は ウ です。

問06 PCM方式 キホン！

■ 40%

音声などのアナログ信号をデジタルデータに変換する方式として、PCM 方式があります。PCM の手順は次のとおりです。

①連続するアナログ信号を短い間隔で細かく測定する（標本化）

②個々の測定値を数値に変換する（量子化）

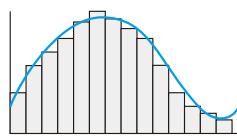
③数値を決められたビット数の符号に変換する（符号化）

以上から、正しい順番は ア 標本化→量子化→符号化となります。

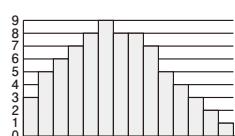
覚えよう！ 問06

PCM 方式といえば
といえ

①標本化（サンプリング）



②量子化



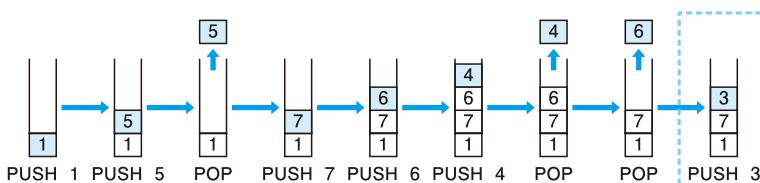
③符号化

011011010110111…

問07 スタック操作 キホン！

■ 60%

スタックは、格納するときは下から順に積み上げ、取り出すときは上から順に取り出すデータ構造です。スタックにデータを格納することを プッシュ、データを取り出すことを ポップといいます。スタック操作の結果は次のとおりです。正解は ウ です。



覚えよう！ 問07

スタックといえば

● 最後に格納したデータを最初に取り出す後入れ先出し型のデータ構造

プッシュ：データをスタッ

クに格納する
ポップ：スタッ

クからデータを取り出す

○ 解答 ○

問04 ア 問05 ウ
問06 ア 問07 ウ

問 08

要素番号が 0 から始まる配列 TANGO がある。 n 個の単語が TANGO [1] から TANGO [n] に入っている。図は、 n 番目の単語を TANGO [1] に移動するために、TANGO [1] から TANGO [$n - 1$] の単語を順に一つずつ後ろにずらして単語表を再構成する流れ図である。a に入れる処理として、適切なものはどれか。



- ア TANGO [i] → TANGO [$i + 1$]
- イ TANGO [i] → TANGO [$n - i$]
- ウ TANGO [$i + 1$] → TANGO [$n - i$]
- エ TANGO [$n - i$] → TANGO [i]

問 09

関数 $f(x, y)$ が次のとおり定義されているとき、 $f(775, 527)$ の値は幾らか。ここで、 $x \bmod y$ は x を y で割った余りを返す。

$f(x, y)$: if $y = 0$ then return x else return $f(y, x \bmod y)$

ア 0

イ 31

ウ 248

エ 527

問 10

外部割込みに分類されるものはどれか。

- ア インターバルタイマによって、指定時間が経過したときに生じる割込み
- イ 演算結果のオーバフローやゼロによる除算で生じる割込み
- ウ 仮想記憶管理において、存在しないページへのアクセスによって生じる割込み
- エ ソフトウェア割込み命令の実行によって生じる割込み

解説

問08 流れ図

流れ図の最初の処理で、TANGO [n] の内容を TANGO [0] に移動しています。この結果、配列 TANGO の内容は次のようにになります。



40%

合格のカギ

問08

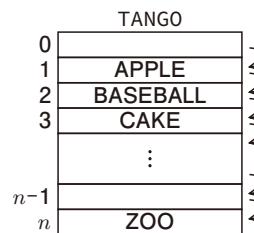
注意 ずらしていく順序を、
 TANGO[0] → TANGO[1],
 TANGO[1] → TANGO[2]…,
 TANGO[n-1] → TANGO[n]
 の順にすると、結局全部の
 要素が TANGO[0] の内容に
 なってしまってうまくいか
 ないよ。



上の状態から、配列 TANGO を目的の状態にするには、TANGO[n-1] の単語を TANGO[n] へ、TANGO[n-2] の単語を TANGO[n-1] へ、… TANGO[0] の単語を TANGO[1] へのように、配列の内容を末尾から 1 つずつ順番にずらしていきます。繰り返しごとの変数 i の値と、単語をずらしていく処理とは、次のように対応します。

変数 i	処理
$n-1$	TANGO[n-1] → TANGO[n]
$n-2$	TANGO[n-2] → TANGO[n-1]
…	…
0	TANGO[0] → TANGO[1]

\uparrow i \uparrow $i+1$



以上から、空欄 a に入る処理は「TANGO[i] → TANGO[i+1]」となります。正解は **ア** です。

問09 再帰的関数

60%

関数 $f(x, y)$ は、 $y = 0$ のとき x 、 $y \neq 0$ のとき $f(y, x \bmod y)$ を返す再帰的関数です。返却値は x と y の最大公約数になります。

- ① $f(775, 527)$: $y \neq 0$ なので、 $f(527, 775 \bmod 527)$ を返す
 $775 \div 527 = 1$ 余り 248
- ② $f(527, 248)$: $y \neq 0$ なので、 $f(248, 527 \bmod 248)$ を返す
 $527 \div 248 = 2$ 余り 31
- ③ $f(248, 31)$: $y \neq 0$ なので、 $f(31, 248 \bmod 31)$ を返す
 $248 \div 31 = 8$ 余り 0
- ④ $f(31, 0)$: $y = 0$ なので、31 を返す

以上から、 $f(775, 527)$ は 31 を返します。正解は **イ** です。

再帰的関数

問09

処理の中で自分自身を呼び出す関数。定義の中に、かならず底入れの規定が含まれる。この規定がないと、無限に自分自身を呼び出してしまう。

覚えよう!

問10

内部割込みといえば

- プログラムによって発生

外部割込みといえば

- ハードウェアによって発生

入出力割込み	入出力処理の完了やエラーで発生
タイマ割込み	タイマによって発生
機械チェック割込み	ハードウェアの障害などによって発生
コンソール割込み	割込みスイッチの操作により発生

問10 外部割込み キホン!

40%

実行中のプログラムを中断し、CPUに強制的に別の処理を実行させることを割込みといいます。割込みには、プログラムによって発生する内部割込みと、ハードウェアによって発生する外部割込みの2種類があります。

- **ア** タイマによる割込みは外部割込み（タイマ割込み）です。
- × **イ** オーバフローやゼロによる除算による割込みは、プログラムのエラーによって生じるので内部割込みです。
- × **ウ** 存在しないページへのアクセス（ページフォールト）はOSによって発生するので内部割込みです。
- × **エ** ソフトウェアによって発生する割込みは内部割込みです。

○ 解答 ○

問08 **ア** 問09 **イ**
 問10 **ア**



問 11

キャッシュメモリに関する記述のうち、適切なものはどれか。

- ア キャッシュメモリにヒットしない場合に割込みが生じ、プログラムによって主記憶からキャッシュメモリにデータが転送される。
- イ キャッシュメモリは、実記憶と仮想記憶とのメモリ容量の差を埋めるために採用される。
- ウ データ書き込み命令を実行したときに、キャッシュメモリと主記憶の両方を書き換える方式と、キャッシュメモリだけを書き換えておき、主記憶の書き換えはキャッシュメモリから当該データが追い出されるときに行う方式がある。
- エ 半導体メモリのアクセス速度の向上が著しいので、キャッシュメモリの必要性は減っている。



問 12

2台のコンピュータを並列に接続して使うシステムがある。それぞれの MTBF と MTTR を次の表に示す。どちらか1台が稼働していればよい場合、システム全体の稼働率は何%か。

	MTBF	MTTR
コンピュータ 1	480 時間	20 時間
コンピュータ 2	950 時間	50 時間

ア 91.2

イ 95.5

ウ 96.5

エ 99.8



問 13

コンパイラにおける最適化の説明として、適切なものはどれか。

- ア オブジェクトコードを生成する代わりに、インタプリタ用の中間コードを生成する。
- イ コンパイルを実施するコンピュータとは異なるアーキテクチャをもったコンピュータで動作するオブジェクトコードを生成する。
- ウ ソースコードを解析して、実行時の処理効率を高めたオブジェクトコードを生成する。
- エ プログラムの実行時に、呼び出されたサブプログラム名やある時点での変数の内容を表示するようなオブジェクトコードを生成する。



問11 キャッシュメモリ

■■■ 50%

キャッシュメモリは、CPUが頻繁に読み込むデータを記録しておく高速なメモリです。主記憶から読み出したデータは、キャッシュメモリに蓄えられ、その後CPUが同じデータを読み出すときはキャッシュメモリから読み出すことで、データ転送を高速化します。

- × **A** キャッシュメモリにデータがない場合でも割込みは発生せず、CPUが主記憶からデータを読み込みます。
- × **イ** キャッシュメモリは、CPUと主記憶との処理速度の差を埋めるための装置です。
- **ウ** 正解です。キャッシュメモリに蓄えられたデータを書き換える場合、キャッシュメモリと主記憶の両方を同時に書き換える方式を**ライトスルー方式**、主記憶の書換えを後回しにする方式を**ライトバック方式**といいます。
- × **エ** CPUも高速化しており、キャッシュメモリの必要性は減っていません。

問12 稼働率の計算 キポン!

■■■ 90%

コンピュータ1、コンピュータ2の稼働率は、MTBF（平均故障間隔）とMTTR（平均修理時間）からそれぞれ次のように求めることができます。

$$\text{コンピュータ1: 稼働率} = \frac{480}{480 + 20} = \frac{480}{500} = 0.96$$

$$\text{コンピュータ2: 稼働率} = \frac{950}{950 + 50} = \frac{950}{1000} = 0.95$$

2台のコンピュータはどちらか1台が稼働していればよいので、**並列システム**と考えます。システム全体の稼働率は次のようにになります。

$$\begin{aligned}\text{稼働率} &= 1 - (1 - 0.96) \times (1 - 0.95) = 1 - 0.04 \times 0.05 \\ &= 1 - 0.002 = 0.998 \rightarrow 99.8\%\end{aligned}$$

以上から、正解は**エ**です。

問13 コンパイラの最適化 キポン!

■■■ 30%

コンパイラは、プログラム言語で記述されたソースコードを、CPUが解釈できる機械語のオブジェクトコードに変換するツールです。コンパイラにおける**最適化**とは、ソースコードを解析して冗長な部分を取り除き、実行時の処理効率を高めたオブジェクトコードを生成する処理をいいます。

- × **ア** バイトコンパイルの説明です。
- × **イ** クロスコンパイルの説明です。
- **ウ** 正解です。
- × **エ** オブジェクトコードにデバッグ情報を付加するコンパイラの機能の説明です。



問11

キャッシュメモリといえば

- 主記憶から読み出したデータを蓄え、CPUが後で同じデータを読み出すときのデータ転送を高速化する
- 主記憶へのアクセス速度とCPUの処理速度の差を埋める
- ヒット率が高いほど速度向上効果が大きい
- ライトスルー方式とライトバック方式がある

④ MTBF

問12

システムが1度故障してから次に故障するまでの稼働時間の平均。

④ MTTR

問12

システムの修理にかかる時間の平均。

④ 稼働率

問12

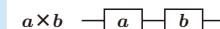
システムの運用時間全体のうち、実際にシステムが稼働している時間の割合のこと。MTBFとMTTRがわかれば、稼働率は次のように求められる。

$$\text{稼働率} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

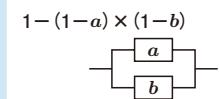


問12

直列システムの稼働率



並列システムの稼働率



④ デバッグ

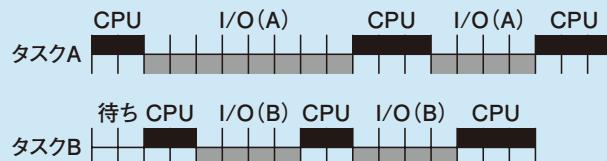
問13

プログラムの誤り（バグ）を発見し、修正すること。

○ 解答 ○

問11 ウ 問12 エ
問13 ウ

問 14 CPU が 1 台で、入出力装置（I/O）が同時動作可能な場合の二つのタスク A, B のスケジューリングは図のとおりであった。この二つのタスクにおいて、入出力装置が CPU と同様に、一つの要求だけを発生順に処理するように変更した場合、両方のタスクが終了するまでの CPU 使用率はおよそ何%か。



- ア 43
- イ 50
- ウ 60
- エ 75

問 15 ページング方式の仮想記憶において、ページ置換アルゴリズムに LRU 方式を採用する。主記憶に割り当てられるページ枠が 4 のとき、ページ 1, 2, 3, 4, 5, 2, 1, 3, 2, 6 の順にアクセスすると、ページ 6 をアクセスする時点で置き換えられるページはどれか。ここで、初期状態では主記憶にどのページも存在しないものとする。

ア 1

イ 2

ウ 4

エ 5

問 16 GPL の下で公開された OSS を使い、ソースコードを公開しなかった場合にライセンス違反となるものはどれか。

- ア OSS とアプリケーションソフトウェアとのインターフェースを開発し、販売している。
- イ OSS の改変を他社に委託し、自社内で使用している。
- ウ OSS の入手、改変、販売を全て自社で行っている。
- エ OSS を利用して性能テストを行った自社開発ソフトウェアを販売している。

問 17 SRAM と比較した場合の DRAM の特徴はどれか。

- ア 主にキャッシュメモリとして使用される。
- イ データを保持するためのリフレッシュ又はアクセス動作が不要である。
- ウ メモリセル構成が単純なので、ビット当たりの単価が安くなる。
- エ メモリセルにフリップフロップを用いてデータを保存する。

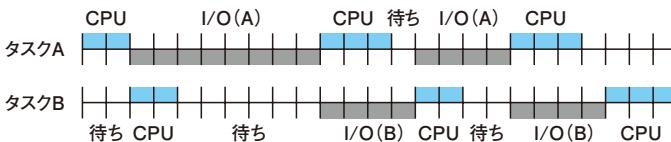
解説

問14 タスクスケジューリング

入出力装置の同時動作ができない場合、タスク A, B のスケジューリングは、次のように変わります。



70%

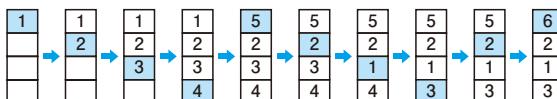


タスク開始から、両方のタスクが終了するまで、全部で 25 目盛かかります。そのうち、CPU を使用している時間は 15 目盛分なので、CPU の使用率は $15 \div 25 = 0.6 \rightarrow 60\%$ になります。正解は **ウ** です。

問15 ページ置換えアルゴリズム

■ 40%

LRU (Least Recently Used) 方式は、ページ枠が足りなくなったとき、最後に参照されてから最も長時間経過したものを新しい内容に置き換える方式です。 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 6$ の順にアクセスしたときのページ枠の推移は、次のようにになります。



図のように、最後のページ 6 にアクセスすると、ページ 5 が置き換えられます。正解は **エ** です。

問16 OSS

■ 20%

OSS (オープンソースソフトウェア) は、ソースコードの公開を条件に、再配布や改変の自由を認めたソフトウェアです。**GPL** (GNU Public License) は、OSS のライセンスの一種です。

- × **ア** OSS 自体を改変・販売しているわけではないので、ライセンス違反ではありません。
- × **イ** 再配布ではないので、ソースコードを公開する義務はありません。
- **ウ** 改変した OSS を再配布（販売を含む）する場合には、ソースコードの公開が義務付けられています。公開しないとライセンス違反になります。
- × **エ** OSS は開発に利用しているだけなので、ライセンス違反ではありません。



問15

ページ置換えアルゴリズム
といえば

- **LRU**：最後に参照されてから最も長く時間が経過しているページを置き換える
- **FIFO**：いちばん古くからあるページを置き換える



問16

OSS といえば

- ソースコード公開を条件に、改変・再配布の自由を認める
- 再配布しない場合はソースコードを公開する必要はない



問17

DRAM が内容を保持するためには、メモリセルの内容を定期的に再書き込みすること。



問17

異なる信号を入力するまで 0 または 1 の状態（2 つの安定状態）を保持する回路。

○ 解答 ○

- | | | | |
|-----|----------|-----|----------|
| 問14 | ウ | 問15 | エ |
| 問16 | ウ | 問17 | ウ |

問17 DRAM と SRAM

■ 80%

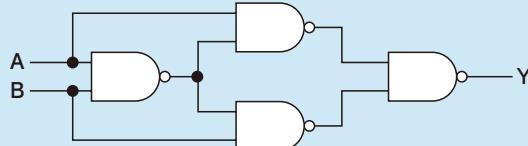
DRAM (Dynamic RAM) と **SRAM** (Static RAM) は、どちらも電源を切ると記録されている内容が消えてしまう揮発性メモリの一種です。両者の特徴はそれぞれ次のとおりです。

DRAM	<ul style="list-style-type: none"> ・内容を保持するために定期的なリフレッシュ動作が必要。 ・構造が単純なのでビット当たりの単価が安い。 ・主に主記憶装置に用いられる。
SRAM	<ul style="list-style-type: none"> ・リフレッシュ動作が不要で高速。 ・フリップフロップ回路で構成される。 ・主にキャッシュメモリとして用いられる。

以上から、DRAM の特徴として正しい記述は **ウ** です。**ア**、**イ**、**エ** はいずれも SRAM の特徴です。

問 18

図の論理回路と等価な回路はどれか。



ア



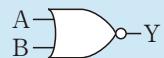
イ



ウ



エ



問 19

顧客に、A～Zの英大文字 26 種類を用いた顧客コードを割り当てる。現在の顧客総数は 8,000 人であって、毎年 2 割ずつ顧客が増えていくものとする。3 年後まで全顧客にコードを割り当てるようにするために、顧客コードは少なくとも何桁必要か。

ア 3

イ 4

ウ 5

エ 6

問 20

コンピュータグラフィックスにおける、レンダリングに関する記述として、適切なものはどれか。

- ア 異なる色のピクセルを混ぜて配置することによって、中間色を表現すること
- イ 複数の静止画を 1 枚ずつ連続表示することによって、動画を作ること
- ウ 物体の表面に陰影を付けたり、光を反射させたりして、画像を作ること
- エ 物体をワイヤフレーム、ポリゴンなどを用いて、モデル化すること

問 21

関係データベースの操作のうち、射影 (projection) の説明として、適切なものはどれか。

- ア ある表の照会結果と、別の表の照会結果を合わせて一つの表にする。
- イ 表の中から特定の条件に合致した行を取り出す。
- ウ 表の中から特定の列だけを取り出す。
- エ 二つ以上の表の組から条件に合致した組同士を合わせて新しい表を作り出す。

解説

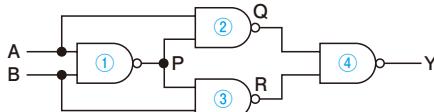
問18 論理回路

図は、**否定論理積素子 (NAND)** を 4 つ組み合わせた論理回路です。各素子の出力を真理値表にまとめると、次のようにになります。



90%

A	B	① A NAND B → P	② A NAND P → Q	③ P NAND B → R	④ Q NAND R → Y
0	0	0 NAND 0 → 1	0 NAND 1 → 1	1 NAND 0 → 1	1 NAND 1 → 0
0	1	0 NAND 1 → 1	0 NAND 1 → 1	1 NAND 1 → 0	1 NAND 0 → 1
1	0	1 NAND 0 → 1	1 NAND 1 → 0	1 NAND 0 → 1	0 NAND 1 → 1
1	1	1 NAND 1 → 0	1 NAND 0 → 1	0 NAND 1 → 1	1 NAND 1 → 0



出力Yは、AとBが両方とも0か両方とも1のとき0、それ以外は1になります。この出力は、**排他的論理和(XOR)**の出力と等価です。

- × ア 論理和素子(OR)の図記号です。
- × イ 論理積素子(AND)の図記号です。
- ウ 排他的論理和素子(XOR)の図記号です。
- × エ 否定論理和素子(NOR)の図記号です。

問19 顧客コードの割り当て キホン!

■ 20%

顧客は現在8,000人で、毎年2割ずつ増えているので、1年後： $8,000 \times 1.2 = 9,600$ 人。2年後： $9,600 \times 1.2 = 11,520$ 人。3年後： $11,520 \times 1.2 = 13,824$ 人。

3年後には13,824人になります。一方、アルファベット大文字は26種類あるので、顧客コードは1桁で26個、2桁で $26^2 = 676$ 個、3桁で $26^3 = 17,576$ 個になります。したがって、3桁あれば3年後の顧客数を上回ります。正解はアです。

問20 コンピュータグラフィックス

■ 30%

レンダリングとは、物体の形や光源などの情報を基に計算を行い、物体の表面の陰影や光の反射などのある画像を作成することです。

- × ア アンチエイリアシングの説明です。
- × イ アニメーションの説明です。
- ウ 正解です。
- × エ モデリングの説明です。

問21 関係演算 キホン!

■ 50%

関係データベースに対する操作には、選択、射影、結合などの操作があります。このうち射影(projection)は、表から特定の列を得る操作です。

- × ア 併合の説明です。
- × イ 選択の説明です。
- ウ 正解です。
- × エ 結合の説明です。

合格のカギ

否定論理積(NAND) 問18
2つの入力が1のとき0、その他の場合は1を出力する回路。出力はANDの反対になる。

0 NAND 0 → 1
0 NAND 1 → 1
1 NAND 0 → 1
1 NAND 1 → 0

排他的論理和(XOR) 問18
入力が両方とも同じ値のとき0、異なる値のとき1になる論理演算。

0 XOR 0 → 0
0 XOR 1 → 1
1 XOR 0 → 1
1 XOR 1 → 0

論理和(OR) 問18
入力が両方とも0のとき0、それ以外は1になる論理演算。

0 OR 0 → 0
0 OR 1 → 1
1 OR 0 → 1
1 OR 1 → 1

論理積(AND) 問18
入力が両方とも1のとき1、それ以外は0になる論理演算。

0 AND 0 → 0
0 AND 1 → 0
1 AND 0 → 0
1 AND 1 → 1

否定論理和(NOR) 問18
論理和(OR)の否定。

0 NOR 0 → 1
0 NOR 1 → 0
1 NOR 0 → 0
1 NOR 1 → 0

覚えよう!

問21

関係演算といえば

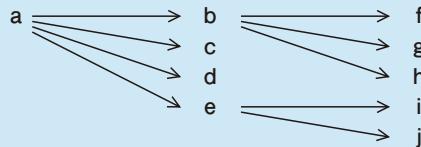
- 射影：特定の列を取り出す
- 選択：特定の行を取り出す
- 結合：2つの表を共通の項目をキーにして結合する

○ 解答 ○

問18 ウ 問19 ア
問20 ウ 問21 ウ

問 22

属性 a の値が決まれば属性 b の値が一意に定まるなどを、 $a \rightarrow b$ で表す。例えば、社員番号が決まれば社員名が一意に定まるなどの表現は、社員番号 \rightarrow 社員名である。この表記法に基づいて、図の関係が成立している属性 a～j を、関係データベース上の三つのテーブルで定義する組合せとして、適切なものはどれか。



- ア テーブル 1 (a)
テーブル 2 (b, c, d, e)
テーブル 3 (f, g, h, i, j)
- ウ テーブル 1 (a, b, f, g, h)
テーブル 2 (c, d)
テーブル 3 (e, i, j)

- イ テーブル 1 (a, b, c, d, e)
テーブル 2 (b, f, g, h)
テーブル 3 (e, i, j)
- エ テーブル 1 (a, c, d)
テーブル 2 (b, f, g, h)
テーブル 3 (e, i, j)

問 23

トランザクション T はチェックポイント取得後に完了したが、その後にシステム障害が発生した。トランザクション T の更新内容をその終了直後の状態にするために用いられる復旧技法はどれか。ここで、チェックポイントの他に、トランザクションログを利用する。

- ア 2相ロック イ シャドウページ ウ ロールバック エ ロールフォワード

問 24

符号化速度が 192k ビット／秒の音声データ 2.4M バイトを、通信速度が 128k ビット／秒のネットワークを用いてダウンロードしながら途切れることなく再生するためには、再生開始前のデータのバッファリング時間として最低何秒間が必要か。

- ア 50

- イ 100

- ウ 150

- エ 250

解説

問22 関数従属 キポン!

ある属性の値によって、他の属性の値が一意に定まる関係を、**関数従属**といいます。図の関係からは、以下の 3 つの関数従属が読み取れます。

- ①属性 a の値が決まると、属性 b, c, d, e の値が一意に定まる
- ②属性 b の値が決まると、属性 f, g, h の値が一意に定まる
- ③属性 e の値が決まると、属性 i, j の値が一意に定まる



合格のカギ

これらを3つのテーブルとして定義します。各テーブルは、属性a, b, eをそれぞれ主キーとし、主キーに関数従属する他の属性をそれぞれの項目とします。すると、各テーブルは次のようになります。

テーブル1 (a, b, c, d, e)

テーブル2 (b, f, g, h)

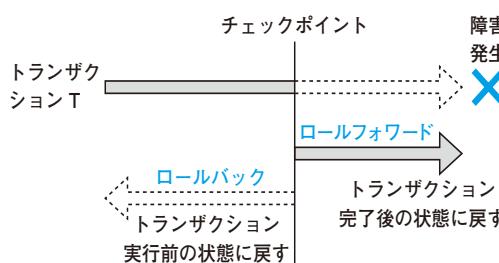
テーブル3 (e, i, j)

以上から、正解はイです。

問23 データベースの復旧技法 キホン!

20%

データベースの更新は、ある程度まとまってからディスク上に書き込まれます。この書き込み処理をチェックポイントといいます。トランザクションTはチェックポイント取得後に終了したので、最後のチェックポイントからトランザクションT終了までの内容を、トランザクションログ(ジャーナル)を利用して復旧しなければなりません。このような復旧技法をロールフォワードといいます。



- × ア 2相ロックとは、トランザクション開始後の第1相で、必要な資源のロックを行い、すべての処理が終わってから（第2相）まとめてロックを解除する方式です。
- × イ シャドウページとは、ディスク上に変更を加えるために用意される作業用の領域です。
- × ウ ロールバックは、チェックポイントの取得時点から、トランザクションログを利用してトランザクションの開始前の状態にデータベースを復旧する復旧方法です。
- エ 正解です。

問24 バッファリング時間の計算

20%

音声データの符号化にかかる時間とダウンロードにかかる時間は、それ次のようになります。

1バイト = 8ビット

$$\text{符号化 : } 2.4\text{M バイト} \div 192\text{k ビット/秒} = \frac{2.4 \times 8 \times 1000}{192} = 100 \text{ 秒}$$

k ビットに換算

$$\text{ダウンロード : } 2.4\text{M バイト} \div 128\text{k ビット/秒} = 2.4 \times 8 \times 1000 \div 128 = 150 \text{ 秒}$$

ダウンロード時間のほうが $150 - 100 = 50$ 秒長くかかります。したがって、符号化をはじめる前に、最低 50 秒間はダウンロードを先行して行ないと、符号化の途中でダウンロード待ちが発生してしまいます。以上から、正解はアです。

④ 主キー

問22

関係データベースの各行を一意に識別するための項目のこと。

覚えよう！

問23

ロールバックといえば

- トランザクション実行前の状態に復元

ロールフォワードといえば

- トランザクション完了後の状態に復元

○ 解答

問22

イ 問23

エ

問24

ア



問 25

OSI 基本参照モデルの各層で中継する装置を、物理層で中継する装置、データリンク層で中継する装置、ネットワーク層で中継する装置の順に並べたものはどれか。

- ア ブリッジ、リピータ、ルータ
- イ ブリッジ、ルータ、リピータ
- ウ リピータ、ブリッジ、ルータ
- エ リピータ、ルータ、ブリッジ



問 26

2台のPCにIPv4アドレスを割り振りたい。サブネットマスクが255.255.255.240のとき、両PCのIPv4アドレスが同一サブネットに所属する組合せはどれか。

- ア 192.168.1.14 と 192.168.1.17
- イ 192.168.1.17 と 192.168.1.29
- ウ 192.168.1.29 と 192.168.1.33
- エ 192.168.1.33 と 192.168.1.49



問 27

TCP/IPにおけるARPの説明として、適切なものはどれか。

- ア IP アドレスから MAC アドレスを得るプロトコルである。
- イ IP ネットワークにおける誤り制御のためのプロトコルである。
- ウ ゲートウェイ間のホップ数によって経路を制御するプロトコルである。
- エ 端末に対して動的に IP アドレスを割り当てるためのプロトコルである。





問25 LAN間接続装置

■ 80%

OSI基本参照モデルは、ネットワークプロトコルを7つの階層に分類し、それぞれの役割を規定したものです。ネットワークを中継する装置は、OSI基本参照モデルのどの層のデータを中継するかによって分類できます。

- ①リピータはケーブルの伝送距離を延長するために、信号を物理層で中継します。
 - ②ブリッジはMACアドレスを基に、データリンク層でフレームを中継します。
 - ③ルータはIPアドレスを基に、ネットワーク層でパケットを中継します。
- 以上から、ウの「リピータ、ブリッジ、ルータ」の順が正解です。

◎ 物理層

問25

OSI基本参照モデルの第1層。信号を物理的に伝送する。

◎ データリンク層

問25

OSI基本参照モデルの第2層。隣り合うノード間でデータ(フレーム)を伝送する。

◎ ネットワーク層

問25

OSI基本参照モデルの第3層。ネットワークを相互に中継し、端末間でデータ(パケット)を伝送する。

◎ 覚えよう!

問25

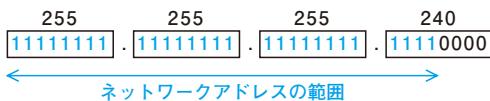
OSI基本参照モデルとLAN間接続装置

OSI基本参照モデル	LAN間接続装置
第4～7層 その他	ゲートウェイ
第3層 ネットワーク層	ルータ
第2層 データリンク層	ブリッジ、レイヤ2スイッチ
第1層 物理層	リピータハブ、リピータ

問26 IPv4アドレス キホン!

■ 50%

サブネットマスクは、IPv4アドレスから、ネットワークアドレスとして使う範囲を定義します。サブネットマスク 255.255.255.240を2進数で表すと



です。1のビットが続く上位28ビットの範囲がネットワークアドレスとなります。

同一サブネットに所属するPCは、ネットワークアドレスが同一です。そこでア～エのうち、上位28ビット(色文字の部分)が同じになるIPv4アドレスの組合せを調べます。

- × ア 192.168.1.00001110 と 192.168.1.00010001
- イ 192.168.1.00010001 と 192.168.1.00011101
- × ウ 192.168.1.00011101 と 192.168.1.00100001
- × エ 192.168.1.00100001 と 192.168.1.00110001

対策 上位24ビットはすべて「192.168.1」なので、下位8ビットを2進数に変換すれば解けるよ。



◎ IPv4

問26

Internet Protocol version4の略。ネットワークでは、ネットワークに接続する端末ごとに固有のIPアドレスを割り当てる。IPアドレスの長さは32ビットなので、割当て可能なアドレスは 2^{32} 個になる。

IPv6ではIPアドレスの長さを128ビットに拡張し、割当て可能なアドレスは一気に 2^{128} 個に増えた。

問27 ARP

■ 20%

イーサネットでは、ネットワーク上の各コンピュータをMACアドレスといふ一意の番号で識別しています。一方、TCP/IPネットワークは、各コンピュータをIPアドレスで識別します。そのため、IPアドレスをMACアドレスに変換する仕組みが必要になります。これを行っているのがARP(Address Resolution Protocol)というプロトコルです。正解はアです。

○ 解答 ○

- 問25 ウ 問26 イ
問27 ア

問 28

ディレクトリトラバーサル攻撃に該当するものはどれか。

- ア Web アプリケーションの入力データとしてデータベースへの命令文を構成するデータを入力し、想定外の SQL 文を実行させる。
- イ Web サイトに利用者を誘導した上で、Web アプリケーションによる HTML 出力のエスケープ処理の欠陥を悪用し、利用者のブラウザで悪意のあるスクリプトを実行させる。
- ウ セッション ID によってセッションが管理されるとき、ログイン中の利用者のセッション ID を不正に取得し、その利用者になりますとしてサーバにアクセスする。
- エ パス名を含めてファイルを指定することによって、管理者が意図していないファイルを不正に閲覧する。

問 29

リバースブルートフォース攻撃に該当するものはどれか。

- ア 攻撃者が何らかの方法で事前に入手した利用者 ID とパスワードの組みのリストを使用して、ログインを試行する。
- イ パスワードを一つ選び、利用者 ID として次々に文字列を用意して総当たりにログインを試行する。
- ウ 利用者 ID、及びその利用者 ID と同一の文字列であるパスワードの組みを次々に生成してログインを試行する。
- エ 利用者 ID を一つ選び、パスワードとして次々に文字列を用意して総当たりにログインを試行する。

問 30

公開鍵暗号を利用した電子商取引において、認証局（CA）の役割はどれか。

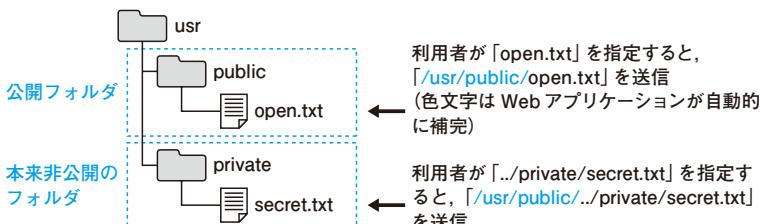
- ア 取引当事者間で共有する秘密鍵を管理する。
- イ 取引当事者の公開鍵に対するデジタル証明書を発行する。
- ウ 取引当事者のデジタル署名を管理する。
- エ 取引当事者のパスワードを管理する。

解説

問28 ディレクトリトラバーサル攻撃

ディレクトリトラバーサル攻撃は、管理者が意図していないパス名を指定して、サーバ内の本来は許されていないファイルにアクセスする攻撃手法です。





- ア SQL インジェクション攻撃の説明です。
- イ クロスサイトスクリプティング攻撃の説明です。
- ウ セッションハイジャック攻撃の説明です。
- エ 正解です。

合格のカギ

SQL インジェクション攻撃

問28

Web アプリケーションの入力データに、データベースへの悪意ある命令文を埋め込んで実行させ、データベースを改ざんしたり、情報を不正に入手したりする攻撃手法。

問29 リバースブルートフォース攻撃 ■■■■■ 30%

リバースブルートフォース攻撃は、よく用いられるパスワードを1つ定めて、それを異なる利用者IDに次々に試して不正にログインしようとする攻撃です。1つの利用者IDに対して複数のパスワードを試すブルートフォース攻撃と逆のやり方なので、リバースブルートフォース攻撃といいます。

- ア パスワードリスト攻撃の説明です。
- イ 正解です。
- ウ ジョーアカウント攻撃の説明です。
- エ ブルートフォース攻撃（総当たり攻撃）の説明です。

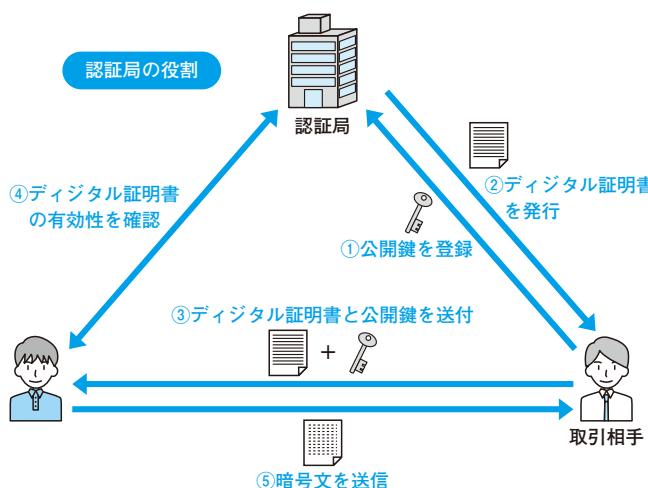
ブルートフォース攻撃

問29

1つの利用者IDに対して、考えられるパスワードを総当たりで試してみる攻撃手法。

問30 認証局(CA)の役割 キホン! ■■■■■ 20%

公開鍵暗号では、送信者が通信を暗号化するときに、受信者の公開鍵を使っています。しかし、そもそも受信者が身元を偽って公開鍵を作成していたら、通信を暗号化しても意味がありません。そこで、公開鍵が受信者本人のものであることを第三者機関が保証する仕組みになっています。この第三者機関が認証局(CA)です。認証局は、受信者の公開鍵に対するデジタル証明書を発行し、その公開鍵の正当性を保証しています。以上から、正解はイです。



覚えよう！

問30

公開鍵暗号方式といえば

- 受信者の公開鍵で暗号化
- 受信者の秘密鍵で復号

公開鍵暗号

問30

暗号化用の鍵（公開鍵）と復号用の鍵（秘密鍵）のペアを利用した暗号方式。送信者は、受信者の公開鍵で通信文を暗号化する。暗号文を受け取った受信者は、秘密に管理している秘密鍵で暗号文を復号する。

デジタル証明書

問30

公開鍵が本人のものであることを証明する証明書。認証局(CA)が発行し、公開鍵に添付される。

○ 解答 ○

- | | |
|-------|-------|
| 問28 エ | 問29 イ |
| 問30 イ | |



問 31

Xさんは、Yさんにインターネットを使って電子メールを送ろうとしている。電子メールの内容を秘密にする必要があるので、公開鍵暗号方式を使って暗号化して送信したい。そのときに使用する鍵はどれか。

- ア Xさんの公開鍵
ウ Yさんの公開鍵

- イ Xさんの秘密鍵
エ Yさんの秘密鍵



問 32

PCへの侵入に成功したマルウェアがインターネット上の指令サーバと通信を行う場合に、宛先ポートとしてTCPポート番号80が多く使用される理由はどれか。

- ア DNSのゾーン転送に使用されることから、通信がファイアウォールで許可されている可能性が高い。
イ WebサイトのHTTPS通信での閲覧に使用されることから、侵入検知システムで検知される可能性が低い。
ウ Webサイトの閲覧に使用されることから、通信がファイアウォールで許可されている可能性が高い。
エ ドメイン名の名前解決に使用されることから、侵入検知システムで検知される可能性が低い。



問 33

生体認証システムを導入するときに考慮すべき点として、最も適切なものはどれか。

- ア 本人のデジタル証明書を、信頼できる第三者機関に発行してもらう。
イ 本人を誤って拒否する確率と他人を誤って許可する確率の双方を勘案して装置を調整する。
ウ マルウェア定義ファイルの更新が頻繁な製品を利用することによって、本人を誤って拒否する確率の低下を防ぐ。
エ 容易に推測できないような知識量と本人が覚えられる知識量とのバランスが、認証に必要な知識量の設定として重要となる。



問 34

WAF (Web Application Firewall) を利用する目的はどれか。

- ア Webサーバ及びWebアプリケーションに起因する脆弱性への攻撃を遮断する。
イ Webサーバ内でワームの侵入を検知し、ワームの自動駆除を行う。
ウ Webサーバのコンテンツ開発の結合テスト時にWebアプリケーションの脆弱性や不整合を検知する。
エ Webサーバのセキュリティホールを発見し、OSのセキュリティパッチを適用する。



問31 公開鍵暗号方式 キホン!

■■■ 50%

公開鍵暗号方式では、暗号化用の鍵と復号用の鍵をペアで使います。このうち復号用の鍵は、第三者の手に渡ると解読に使われてしまうので、受信者の手元で厳重に保管します（**秘密鍵**）。

一方、それと対になる受信者の暗号化用の鍵は、自分宛のメッセージを暗号化してもらうために、送信者に公開します（**公開鍵**）。暗号化用の鍵は暗号化専用なので、公開しても解読に使われる心配はありません。以上から、

暗号化用の鍵=受信者の公開鍵　**復号用の鍵=受信者の秘密鍵**

となります。問題は「Yさん宛のメールを暗号化するのに使用する鍵」なので、

Yさんの公開鍵が正解です。

問32 TCP ポート番号 80

■■■ 20%

内部に侵入したマルウェアが外部と密かに通信するためには、ファイアウォールなどで通信が遮断されない経路を利用する必要があります。TCP ポートの 80 番は、**ウェルノウンポート番号**と呼ばれる既定のポート番号の 1 つで、Web サーバとの通信に普段から利用されています。したがって、Web サイトの閲覧に使用している PC なら、通信が許可されている可能性が高いと考えられます。

- A** DNS のゾーン転送には TCP ポートの 53 番が使われます。
- I** HTTPS 通信には TCP ポートの 443 番が使われます。
- ウ** 正解です。
- E** DNS の名前解決には、UDP ポートの 53 番が使われます。

問33 生体認証システムの導入

■■■ 20%

生体認証システム（バイオメトリクス認証）は、指紋や静脈といった人間の身体的特徴を検出して本人を識別する認証技術です。

生体認証の認証エラーには、①間違って本人を拒否してしまう場合（**FRR**）と、②間違って他人を許可してしまう場合（**FAR**）があります。本人判定の基準を厳しくすると、②のエラーが少なくなる代わりに①のエラーが増え、本人判定の基準をゆるくすると、①のエラーが少くなる代わりに②のエラーが増えます。そのため、双方を勘案して検出装置を調整しなければなりません。正解は **I** です。

問34 WAF (Web Application Firewall)

■■■ 20%

WAF（Web アプリケーションファイアウォール）は、クライアントと Web サーバとの間に設置して、クライアントから Web サーバに対する攻撃を遮断するファイアウォールです。Web サーバや Web アプリケーションにセキュリティ上の欠点があると、そこを突いて不正侵入やサービス妨害などの攻撃を受けるおそれがあります。WAF は、Web サーバに送信されたデータを検査して不正なデータを遮断し、Web サーバを防御します。正解は **A** です。

マルウェア

問32

コンピュータウイルスやスパイウェア、ワームなどの悪質なプログラムの総称。

ポート番号

問32

インターネットの TCP や UDP プロトコルで、宛先のアプリケーションの識別に使われる番号。代表的なアプリケーションについては、標準のポート番号があらかじめ決められており、ウェルノウンポート番号と呼ばれる。ファイアウォールのパケットフィルタリングでは、パケットの宛先ポート番号や送信元ポート番号を調べて、通信を許可／遮断するかどうかを決定する。

FRR

問33

False Rejection Rate：本人拒否率。生体認証で本人を誤って拒否する確率。

FAR

問33

False Accept Rate：他人受入率。他人が誤って本人として認証される確率。

ワーム

問34

ネットワークを介して他のシステムに侵入し、自己増殖していく不正なプログラム。

覚えよう！

問34

WAFといえ

- Web アプリケーションへの通信内容を監視
- SQL インジェクションなどの攻撃を遮断

○ 解答 ○

- | | |
|-----------------------------------------------|-----------------------------------------------|
| 問31 <input checked="" type="radio"/> ウ | 問32 <input checked="" type="radio"/> ウ |
| 問33 <input type="radio"/> イ | 問34 <input checked="" type="radio"/> ア |

問 35 社内ネットワークとインターネットの接続点にパケットフィルタリング型ファイアウォールを設置して、社内ネットワーク上のPCからインターネット上のWebサーバの80番ポートにアクセスできるようにするとき、フィルタリングで許可するルールの適切な組合せはどれか。

ア	送信元	宛先	送信元ポート番号	宛先ポート番号
	PC	Web サーバ	80	1024 以上
	Web サーバ	PC	80	1024 以上

イ	送信元	宛先	送信元ポート番号	宛先ポート番号
	PC	Web サーバ	80	1024 以上
	Web サーバ	PC	1024 以上	80

ウ	送信元	宛先	送信元ポート番号	宛先ポート番号
	PC	Web サーバ	1024 以上	80
	Web サーバ	PC	80	1024 以上

エ	送信元	宛先	送信元ポート番号	宛先ポート番号
	PC	Web サーバ	1024 以上	80
	Web サーバ	PC	1024 以上	80

問 36 JIS Q 27001:2014（情報セキュリティマネジメントシステム－要求事項）において、リスクを受容するプロセスに求められるものはどれか。

- ア 受容するリスクについては、リスク所有者が承認すること
- イ 受容するリスクを監視やレビューの対象外とすること
- ウ リスクの受容は、リスク分析前に行うこと
- エ リスクを受容するかどうかは、リスク対応後に決定すること

問 37 ウイルス検出におけるビヘイビア法に分類されるものはどれか。

- ア あらかじめ検査対象に付加された、ウイルスに感染していないことを保証する情報と、検査対象から算出した情報を比較する。
- イ 検査対象と安全な場所に保管してあるその原本とを比較する。
- ウ 検査対象のハッシュ値と既知のウイルスファイルのハッシュ値とを比較する。
- エ 検査対象をメモリ上の仮想環境下で実行して、その挙動を監視する。



問35 パケットフィルタリング

■■■30%

パケットフィルタリングは、通過するパケットのIPアドレスやポート番号を検査して、通信の許可／不許可を判断するファイアウォールです。許可／不許可の判断は、あらかじめ設定しておいたフィルタリングルールに従います。

社内PCからWebサーバにアクセスできるようにフィルタリングルールを設定するには、PCからWebサーバへの通信と、WebサーバからPCへの通信の両方を許可する必要があります。

Webサーバはサービス要求窓口として、ポート番号80番を使用します。したがって、PCからWebサーバへの送信は、宛先ポート番号に**80**を使用します。なお、PC側の送信元ポート番号は、Webブラウザが**1024**以上の番号を適宜使用します。

送信元	宛先	送信元ポート番号	宛先ポート番号
PC	Webサーバ	1024以上	80

一方、WebサーバからPCへの送信では、Webサーバが送信元なので、送信元ポート番号に**80**を指定します。

送信元	宛先	送信元ポート番号	宛先ポート番号
Webサーバ	PC	80	1024以上

以上の組合せを許可するファイアウォールの設定は、**ウ**です。

問36 リスク受容

■■■30%

リスクを完全にゼロにすることはできないので、許容範囲内のリスクは受け入れます。これを**リスク受容**といいます。JIS Q 27001によれば、受容するリスクについてはリスク所有者（リスクをとる責任者）の承認が必要です。

- **ア** 正解です。
- × **イ** リスクの大きさが変化する場合もあるので、いったん受容したリスクでも、監視やレビューは引き続き行います。
- × **ウ** リスクが受容可能な大きさかどうかは**リスク分析**によって明らかになるので、リスク分析前に受容することはできません。
- × **エ** 受容するリスクと受容できないリスクを定め、受容できないリスクについては**リスク対応**を実施します。したがって、リスクを受容するかどうかはリスク対応前に決定する必要があります。

JIS Q 27001 問36
情報セキュリティマネジメントシステム（ISMS）の基準を定めた国際規格ISO/IEC27001を日本語化した規格。ISMSの認証基準として使われている。

問37 ビヘイビア法

■■■20%

ビヘイビア法は、ウイルスの感染や発症時の挙動を監視して、その動作や異常な現象からウイルスを検知する手法です。使用中のシステムでウイルスを実行することはできないので、検査対象は隔離された仮想環境下で実行します。

- × **ア** チェックサム法の説明です。
- × **イ** コンペア法の説明です。
- × **ウ** ハッシュ値による検出方法の説明です。
- **エ** 正解です。

ハッシュ値 問37
任意の長さのデータをもとに生成された固定長のデータ。ハッシュ値を生成するプログラムをハッシュ関数という。同一データから生成したハッシュ値は等しくなる。一方、ハッシュ値から元のデータを復元することはできない。

○ 解答 ○

問35 **ウ** 問36 **ア**
問37 **エ**

問 38 UML2.0において、オブジェクト間の相互作用を時系列に表す図はどれか。

- ア アクティビティ図
ウ シーケンス図

- イ コンポーネント図
エ 状態遷移図

問 39 オブジェクト指向の基本概念の組合せとして、適切なものはどれか。

- ア 仮想化、構造化、投影、クラス
イ 具体化、構造化、連続、クラス
ウ 正規化、カプセル化、分割、クラス
エ 抽象化、カプセル化、継承、クラス

問 40 ブラックボックステストにおけるテストケースの設計方法として、適切なものはどれか。

- ア プログラム仕様書の作成又はコーディングが終了した段階で、仕様書やソースリストを参照して、テストケースを設計する。
イ プログラムの機能仕様やインターフェースの仕様に基づいて、テストケースを設計する。
ウ プログラムの処理手順や内部構造に基づいて、テストケースを設計する。
エ プログラムの全ての条件判定で、真と偽をそれぞれ1回以上実行させることを基準に、テストケースを設計する。

問 41 ソフトウェアのリバースエンジニアリングの説明はどれか。

- ア 開発支援ツールなどを用いて、設計情報からソースコードを自動生成する。
イ 外部から見たときの振る舞いを変えずに、ソフトウェアの内部構造を変える。
ウ 既存のソフトウェアを解析し、その仕様や構造を明らかにする。
エ 既存のソフトウェアを分析し理解した上で、ソフトウェア全体を新しく構築し直す。

解説

問38 UML2.0の図

UMLで、オブジェクト間の相互作用を時系列に表すには、シーケンス図を使います。

- × **ア アクティビティ図**は、プログラムの制御の流れを表す流れ図のような図です。
× **イ コンポーネント図**は、システムを構成する物理的な要素（コンポーネント）と、コンポーネント間の関係を表す図です。

 **合格のカギ**

 **UML**

Unified Modeling Language: オブジェクト指向開発で利用する様々な種類の図式を規格化したもの。

○ ウ 正解です。

× エ 状態遷移図（ステートマシン図）は、発生するイベントに応じてプログラムの状態がどう変化するかを表す図です。

合格のカギ

問39 オブジェクト指向の基本概念 キホン!

40%

オブジェクト指向の主な基本概念として、以下の4つがあります。

抽象化	複数のクラスから共通する属性やメソッドを抜き出した上位クラスを定義し、異なるクラスのオブジェクトを統一的に扱えるようにすること。
カプセル化	オブジェクト内部の情報を隠蔽し、用意したインターフェースを介してのみアクセスできるようにすること。
継承	上位クラスの属性やメソッドを、下位クラスに引き継ぐこと。
クラス	オブジェクトが備える属性やメソッドの定義。

以上から、正解はエです。

問40 ブラックボックステスト キホン!

30%

ブラックボックステストは、プログラムが仕様どおりに動作するかどうかを、内部のロジックは考慮せず、入力に対する出力に着目してテストします。したがって、テストケースはプログラムの機能仕様やインターフェース仕様に基づいて設計します。

ブラックボックステストに対して、プログラム内部のロジックが正しいかどうかを検証するテストをホワイトボックステストといいます。

× ア プログラム仕様書やソースリストは、ホワイトボックステストで参照します。

○ イ 正解です。

× ウ プログラムの内部構造に基づいてテストケースを設計するのは、ホワイトボックステストです。

× エ すべての条件判定が真になる場合と偽になる場合をテストすることを条件網羅といいます。条件網羅はホワイトボックステストのテストケースを設計する手法です。

問41 リバースエンジニアリング

30%

ソフトウェア開発は、おおまかに設計→プログラミング→運用の順に進めるのが一般的です。リバースエンジニアリングはこの順序を逆転（リバース）し、できあがったソフトウェアからソースコードを解析したり、ソースコードから仕様や構造を復元する手法です。

× ア 設計情報からソースコードを生成するのはリバースではなく、通常のエンジニアリングの工程です。

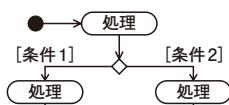
× イ リファクタリングの説明です。

○ ウ 正解です。

× エ リエンジニアリングの説明です。

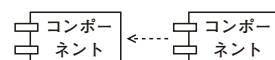
アクティビティ図の例

問38



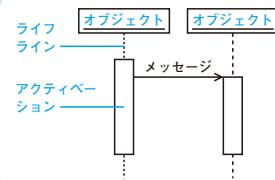
コンポーネント図の例

問38



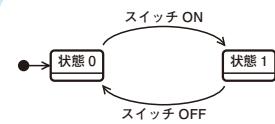
シーケンス図の例

問38



状態遷移図の例

問38



覚えよう！

問40

ブラックボックステスト といえば

- ・プログラムの外部仕様に沿ってテストケースを設定
- ・同値分割法、限界値分析などがある

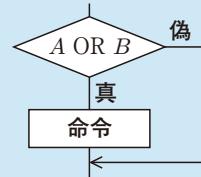
ホワイトボックステスト といえば

- ・プログラムの内部構造に沿ってテストケースを設定
- ・条件網羅、判定条件網羅、分歧網羅などがある

解答

問38 ウ 問39 エ
問40 イ 問41 ウ

- 問 42** プログラム中の図の部分を判定条件網羅（分岐網羅）でテストするときのテストケースとして、適切なものはどれか。



ア

A	B
偽	真

イ

A	B
偽	真
真	偽

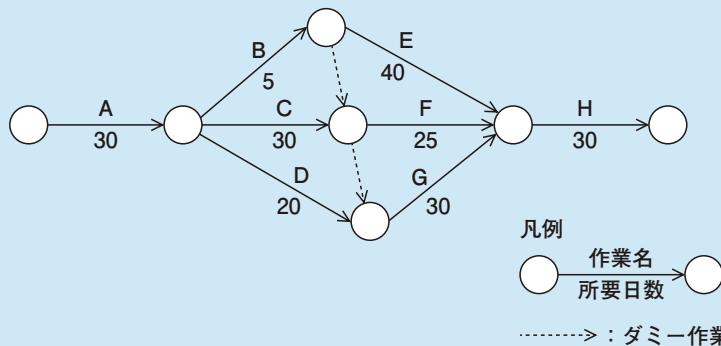
ウ

A	B
偽	偽
真	真

エ

A	B
偽	真
真	偽
真	真

- 問 43** 図のアローダイアグラムで表されるプロジェクトは、完了までに最短で何日を要するか。



ア 105

イ 115

ウ 120

エ 125

- 問 44** ファンクションポイント法の説明はどれか。

- ア 開発するプログラムごとのステップ数を積算し、開発規模を見積もる。
- イ 開発プロジェクトで必要な作業のWBSを作成し、各作業の工数を見積もる。
- ウ 外部入出力や内部論理ファイル、外部照会、外部インターフェースファイルの個数と特性などから開発規模を見積もる。
- エ 過去の類似例を探し、その実績や開発するシステムとの差異などを分析・評価して開発規模を見積もる。

解説



問42 判定条件網羅（分岐網羅） キホン!

60%

判定条件網羅（分岐網羅）では、プログラム中のすべての経路が実行されるようにテストケースを設定します。問題の流れ図には、条件「 $A \text{ OR } B$ 」が真


合格のカギ

になる場合と、偽になる場合の2つの経路があるので、

- ① 「A OR B」が偽になる場合→A, Bが両方とも偽
- ② 「A OR B」が真になる場合→A, Bのいずれかまたは両方が真

の2種類の組合せが必要になります。この条件を満たすのはウのテストケースです。

- × ア 命令網羅（プログラム中の命令をすべて実行する）のテストケースです。
- × イ 条件網羅（単独条件がそれぞれ真の場合と偽の場合）のテストケースです。
- ウ 正解です。
- × エ AとBが両方とも偽になるケースがないので、条件網羅と変わりありません。


覚えよう！
問42
判定条件網羅といえば

- 判定条件が「真」になる場合と「偽」になる場合をテスト

命令網羅といえば

- すべての命令を少なくとも1回は実行

条件網羅といえば

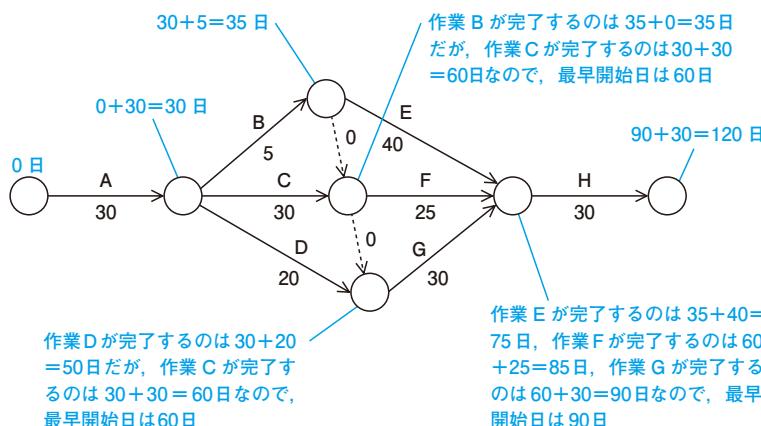
- 各判定条件がそれぞれ「真」になる場合と「偽」になる場合をテスト

複数条件網羅といえば

- 各判定条件のすべての「真」「偽」の組合せをテスト

問43 アローダイアグラム キホン！
 80%

図の○で表される各結合点ごとに、すべての先行作業が完了する日=次の作業を開始できる日（最早開始日）を求めます。なお、ダミー作業の所要日数は0とみなします。



以上から、最後の作業 H が完了するのは、開始から 120 日後になります。
正解はウです。


最早開始日
問43

すべての先行作業が完了し、次の作業が開始できる最も早い日。最初の結合点から経路に沿って所要時間を足していく。複数の経路がある場合は、所要時間の大きいほうが最早開始日になる。

問44 ファンクションポイント法 キホン！
 60%

ファンクションポイント法は、開発するソフトウェアの機能を、入出力データやファイル、画面などの個数と、それぞれの複雑度によって点数化し、開発規模を見積もる手法です。

- × ア プログラムステップ法の説明です。
- × イ WBS法の説明です。
- ウ 正解です。
- × エ 類推法の説明です。


覚えよう！
問44
ファンクションポイント法といえば

- 入出力、画面、ファイルなどの機能の個数によって開発規模を見積もる
- 個々の機能を複雑さに応じて点数化する

○ 解答 ○

問42 ウ 問43 ウ
問44 ウ



問 **45**

サービスデスク組織の構造とその特徴のうち、ローカルサービスデスクのものはどれか。

- ア** サービスデスクを1拠点又は少数の場所に集中することによって、サービス要員を効率的に配置したり、大量のコールに対応したりすることができる。
- イ** サービスデスクを利用者の近くに配置することによって、言語や文化が異なる利用者への対応、専門要員によるVIP対応などができる。
- ウ** サービス要員が複数の地域や部門に分散していても、通信技術の利用によって単一のサービスデスクであるかのようにサービスが提供できる。
- エ** 分散拠点のサービス要員を含めた全員を中央で統括して管理することによって、統制のとれたサービスが提供できる。



問 **46**

経営者が社内のシステム監査人の外観上の独立性を担保するために講じる措置として、適切なものはどれか。

- ア** システム監査人にITに関する継続的学習を義務付ける。
- イ** システム監査人に必要な知識や経験を定め、公表する。
- ウ** システム監査人の監査技法修得制度を設ける。
- エ** システム監査人の所属部署を経営者の直轄とする。



問 **47**

システム監査報告書に記載する指摘事項に関する説明のうち、適切なものはどれか。

- ア** 監査証拠による裏付けの有無にかかわらず、監査人が指摘事項とする必要があると判断した事項を記載する。
- イ** 監査人が指摘事項とする必要があると判断した事項のうち、監査対象部門の責任者が承認した事項を記載する。
- ウ** 調査結果に事実誤認がないことを監査対象部門に確認した上で、監査人が指摘事項とする必要があると判断した事項を記載する。
- エ** 不備の内容や重要性は考慮せず、全てを漏れなく指摘事項として記載する。

解説

問45 ローカルサービスデスク

■■30%

サービスデスクは、ITサービスを利用する顧客に対する窓口として、問合せ受付などのサポート業務を行います。サービスデスクの種類として、①中央サービスデスク、②ローカルサービスデスク、③バーチャルサービスデスクがあります。このうちローカルサービスデスクは、サービス要員を利用者の側に配置することで、現地でのサポートをしやすくするものです。

- ア 中央サービスデスクの説明です。
- イ 正解です。
- ウ バーチャルサービスデスクの説明です。
- エ 分散拠点を中央で統括管理するのは、ローカルサービスデスクではありません。



覚えよう！ 問45

サービスデスクの種類

- 中央サービスデスク：拠点を1か所に集中
- ローカルサービスデスク：利用者側に配置
- バーチャルサービスデスク：仮想的に単一のサービスデスクを提供

問46 システム監査人の独立性 キホン！

■■50%

システム監査人の外観上の独立性とは、監査人と監査対象との間に、第三者から見て公正な監査を疑わせるような利害関係がないことをいいます。

一般に、監査対象に監査人の上司が含まれていたりすると、公平な監査をするのが難しくなります。こうした問題は監査部門を経営者の直属にすれば解消されます。正解はエです。

覚えよう！ 問46

監査人の独立性といえば

- 監査人が、監査される側と利害関係をもたないこと

問47 監査報告書の指摘事項

■■30%

監査報告書に記載する指摘事項は、監査の結果、改善が必要と考えられる事項です。指摘事項は客観的な事実にもとづくものでなければなりません。そのため指摘事項の記載にあたっては、根拠となるシステム監査人の所見や監査証拠などについて、監査対象部門に事実確認を行います。

- ア 監査証拠による裏付けが必要です。
- イ 監査対象部門の責任者の承認は必ずしも必要ではありません。
- ウ 正解です。
- エ 不備のすべてを漏れなく指摘事項とする必要はありません。

覚えよう！ 問46

監査人の独立性の観点から避けるべきことといえば

- ①自分が所属している部署の業務を監査する。
 - ②監査人の上司を監査する。
 - ③納入したシステムを納入業者自身が監査する。
- といったケースは、システム監査人の独立性の観点から避けるべきである。

○ 解答 ○

- | | |
|-------|-------|
| 問45 イ | 問46 エ |
| 問47 ウ | |

問 48 エンタープライズアーキテクチャ (EA) を説明したものはどれか。

- ア** オブジェクト指向設計を支援する様々な手法を統一して標準化したものであり、クラス図などのモデル図によってシステムの分析や設計を行うための技法である。
- イ** 概念データモデルを、エンティティとリレーションシップで表現することによって、データ構造やデータ項目間の関係を明らかにするための技法である。
- ウ** 各業務と情報システムを、ビジネス、データ、アプリケーション、テクノロジの四つの体系で分析し、全体最適化の観点から見直すための技法である。
- エ** 企業のビジネスプロセスを、データフロー、プロセス、ファイル、データ源泉／データ吸収の四つの基本要素で抽象化して表現するための技法である。

問 49 SOA の説明はどれか。

- ア** 売上・利益の増加や、顧客満足度の向上のために、営業活動にITを活用して営業の効率と品質を高める概念のこと
- イ** 経営資源をコアビジネスに集中させるために、社内業務のうちコアビジネス以外の業務を外部に委託すること
- ウ** コスト、品質、サービス、スピードを革新的に改善させるために、ビジネスプロセスをデザインし直す概念のこと
- エ** ソフトウェアの機能をサービスという部品とみなし、そのサービスを組み合わせることによってシステムを構築する概念のこと

問 50 企業が保有する顧客や市場などの膨大なデータから、有用な情報や関係を見つけ出す手法はどれか。

- | | |
|-----------------------|---------------------|
| ア データウェアハウス | イ データディクショナリ |
| ウ データフローダイアグラム | エ データマイニング |

問 51 グリーン調達の説明はどれか。

- ア** 環境保全活動を実施している企業がその活動内容を広くアピールし、投資家から環境保全のための資金を募ることである。
- イ** 第三者が一定の基準に基づいて環境保全に資する製品を認定する、エコマークなどの環境表示に関する国際規格のことである。
- ウ** 太陽光、バイオマス、風力、地熱などの自然エネルギーによって発電されたグリーン電力を、市場で取引可能にする証書のことである。
- エ** 品質や価格の要件を満たすだけでなく、環境負荷が小さい製品やサービスを、環境負荷の低減に努める事業者から優先して購入することである。



問48 エンタープライズアーキテクチャ (EA)

■■■■■ 40%

エンタープライズアーキテクチャ (EA) とは、組織全体の業務と情報システムを4つの体系（ビジネス、データ、アプリケーション、テクノロジ）にしたがって分析し、全体最適化を図る手法です。

- ア UML（統一モデリング言語）の説明です。
- イ E-R図の説明です。
- ウ 正解です。
- エ データフローダイアグラム (DFD) の説明です。

問49 SOA

■■■■■ 40%

SOAは、ソフトウェアの機能をサービスととらえ、複数のサービスを部品のように組み合わせてシステムを構築していく設計手法です。

- ア SFA (Sales Force Automation) の説明です。
- イ BPO (Business Process Outsourcing) の説明です。
- ウ BPR (Business Process Reengineering) の説明です。
- エ 正解です。

問50 データマイニング

■■■■■ 20%

大量に蓄積されたデータから、統計的・数学的手法を用いて何らかの有用な情報や関係を見つける手法をデータマイニングといいます。

- ア データウェアハウスは、蓄積された大量の情報を時系列的にデータベース化したものです。
- イ データディクショナリは、DBMS（データベース管理システム）が管理するデータや利用者、プログラムなどの情報を一か所に集めて管理するためのデータの集合体です。
- ウ データフローダイアグラム (DFD) は、業務過程をデータの流れに着目して図式化したものです。
- エ 正解です。

問51 グリーン調達

■■■■■ 30%

グリーン調達（グリーン購入）とは、環境への負荷が少ない製品やサービスを優先的に購入することです。

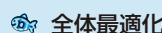
- ア 環境保全活動に取り組む企業に積極的に投資することをグリーン投資といいます。
- イ 環境ラベリング制度の国際規格（ISO 14020シリーズ）の説明です。
- ウ グリーン電力証書の説明です。
- エ 正解です。



問48

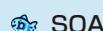
エンタープライズアーキテクチャといえば

- ビジネスアーキテクチャ：業務機能の構成
- データアーキテクチャ：業務機能に使われる情報の構成
- アプリケーションアーキテクチャ：業務機能と情報の流れをまとめたサービス群の構成
- テクノロジアーキテクチャ：各サービスを実現するための技術の構成



問48

企業の情報システムと業務の全体を、経営戦略に基づいて見直すこと。



問49

Service-Oriented Architecture: サービス指向アーキテクチャ。業務過程の個々のプロセスをサービスとみなし、複数のサービスを組み合わせてシステムを構築する設計手法。



問49

SOA といえば

- サービスを組み合わせてシステムを構成する設計手法

参考 データマイニングは「データ採掘」という意味だね。

問50



○ 解答 ○

問48 ウ 問49 エ

問50 エ 問51 エ



問 52

SWOT分析を用いて識別した、自社製品に関する外部要因はどれか。

- ア 営業力における強み
- イ 機能面における強み
- ウ 新規参入による脅威
- エ 品質における弱み



問 53

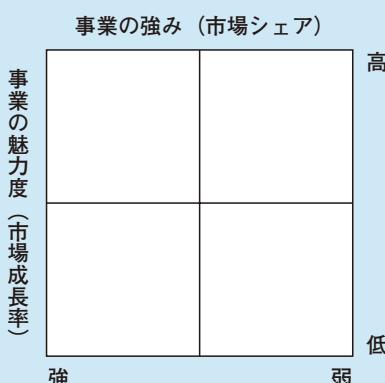
競争上のポジションで、ニッチャの基本戦略はどれか。

- ア シェア追撃などのリーダ攻撃に必要な差別化戦略
- イ 市場チャンスに素早く対応する模倣戦略
- ウ 製品、市場の専門特化を図る特定化戦略
- エ 全市場をカバーし、最大シェアを確保する全方位戦略



問 54

図に示すマトリックスを用いたポートフォリオマネジメントによって、事業計画や競争優位性の分析を行う目的はどれか。



- ア 目標として設定したプロモーション効果を測定するために、自社の事業のポジションを評価する。
- イ 目標を設定し、資源配分の優先順位を設定するための基礎として、自社の事業のポジションを評価する。
- ウ 目標を設定し、製品の品質を高めることによって、市場での優位性を維持する方策を評価する。
- エ 目標を設定するために、季節変動要因や地域的広がりを加味することによって、市場の変化を評価する。

問52 SWOT分析 キホン!

■■■■■ 40%

SWOT分析とは、ある企業や組織がもっている強み (Strengths) と弱み (Weaknesses)、機会 (Opportunities) と脅威 (Threats) を分析し、目標を達成するための戦略を立てていく手法です。

4つの要因のうち、強みと弱みは企業自体がもっている**内部要因**であり、機会と脅威は企業の外側に存在する**外部要因**です。

- × **A** 営業力における強みは内部要因です。
- × **イ** 機能面における強みは内部要因です。
- **ウ** 正解です。ライバル企業が市場に新規参入してくる脅威は、自社の外部に存在する外部要因です。
- × **エ** 品質における弱みは内部要因です。

問53 ニッチャの基本戦略

■■■■■ 50%

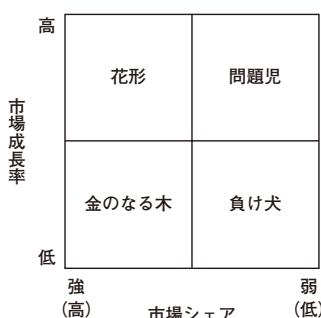
経営学者のコトラー (Kotler) は、企業の競争上のポジションを①リーダ、②チャレンジャ、③フォロワ、④ニッチャの4つに分類しています。このうち**ニッチャ**の基本戦略は、専門的な市場に特化する特定化戦略で、狭くても確実なニーズのある市場を狙います。

- × **ア** チャレンジャの戦略です。
- × **イ** フォロワの戦略です。
- **ウ** 正解です。
- × **エ** リーダの戦略です。

問54 プロダクトポートフォリオマネジメント キホン!

■■■■■ 30%

プロダクトポートフォリオマネジメント (PPM) では、マトリックス表の4つの象限を市場シェア、市場成長率の度合いに応じて「花形」「金のなる木」「問題児」「負け犬」に分類します。自社の事業や製品がこのうちのどこに位置するかによって、資源配分の優先順位を設定します。



花形：市場の成長にともない、優先的に資源配分が必要。

金のなる木：現在の主たる資金供給源であり、新たに資源を配分する必要はない。

問題児：資源配分によって、新たな資金供給源となる可能性がある。

負け犬：新たに資源を配分する必要性は低く、将来的な撤退も考慮する。

- × **ア** プロモーション効果は測定できません。
- **イ** 正解です。
- × **ウ** 製品の品質については分析できません。
- × **エ** 季節変動要因や地域的広がりなどは分析できません。

合格のカギ

覚えよう！

問52

SWOT分析といえば

内部要因（環境）

- 強み (Strengths)
- 弱み (Weaknesses)

外部要因（環境）

- 機会 (Opportunities)
- 脅威 (Threats)

覚えよう！

問53

企業の競争戦略といえば

- リーダ：すべての顧客をターゲットとする全方位戦略
- チャレンジャ：市場シェア獲得をねらう差別化戦略
- フォロワ：市場動向に素早く追随する模倣戦略
- ニッチャ：専門市場に特化する特定化戦略

覚えよう！

問54

プロダクトポートフォリオマネジメントといえば

- **花形**：市場成長率、市場シェアともに高い
- **金のなる木**：市場成長率は低いが市場シェアが高い
- **問題児**：市場成長率は高いが市場シェアは低い
- **負け犬**：市場成長率、市場シェアともに低い

○ 解答 ○

問52 **ウ** 問53 **ウ**
問54 **イ**



問 **55**

IoT (Internet of Things) の実用例として、適切でないものはどれか。

- ア インターネットにおけるセキュリティの問題を回避するために、サーバに接続せず、単独でファイルの管理、演算処理、印刷処理などの作業を行うコンピュータ
- イ 大型の機械などにセンサと通信機能を内蔵して、稼働状況、故障箇所、交換が必要な部品などを、製造元がインターネットを介してリアルタイムに把握できるシステム
- ウ 検針員に代わって、電力会社と通信して電力使用量を送信する電力メータ
- エ 自動車同士及び自動車と路側機が通信することによって、自動車の位置情報をリアルタイムに収集して、渋滞情報を配信するシステム



問 **56**

CGM (Consumer Generated Media) の説明はどれか。

- ア オークション形式による物品の売買機能を提供することによって、消費者同士の個人売買の仲介役を果たすもの
- イ 個人が制作したデジタルコンテンツの閲覧者・視聴者への配信や利用者同士の共有を可能とするもの
- ウ 個人商店主のオンラインショップを集め、共通ポイントの発行やクレジットカード決済を代行するもの
- エ 自社の顧客のうち、希望者をマーリングリストに登録し、電子メールを通じて定期的に情報を配信するもの



問 **57**

車載機器の性能の向上に関する記述のうち、ディープラーニングを用いているものはどれか。

- ア 車の壁への衝突を加速度センサが検知し、エアバッグを膨らませて搭乗者をけがから守った。
- イ システムが大量の画像を取得し処理することによって、歩行者と車をより確実に見分けることができるようになった。
- ウ 自動でアイドリングストップする装置を搭載することによって、運転経験が豊富な運転者が運転する場合よりも燃費を向上させた。
- エ ナビゲーションシステムが、携帯電話回線を通してソフトウェアのアップデートを行い、地図を更新した。



問 **58**

財務諸表のうち、一定時点における企業の資産、負債及び純資産を表示し、企業の財政状態を明らかにするものはどれか。

- ア 株主資本等変動計算書
- ウ 損益計算書

- イ キャッシュフロー計算書
- エ 貸借対照表

問55 IoT の実用例

■■■ 40%

IoT (Internet of Things) とは、「モノのインターネット」のことで、情報端末ばかりでなく、様々なモノに通信機能をもたせ、インターネットを介して情報を収集・解析して高度な判断やサービスを実現することです。

- ア インターネットに接続せず、オフラインでコンピュータを使用するのは、IoT の例ではありません。
- × イ 設備の稼働監視に IoT を活用した工場を、スマート工場といいます。
- × ウ IoT 化した電力メータをスマートメータといいます。
- × エ IoT を活用した交通情報システムの例です。

合格のカギ

スマートメータ

問55



問56 CGM (消費者生成メディア)

■■■ 40%

CGM (Consumer Generated Media) とは、消費者みずからが発信する情報やデジタルコンテンツで構成される、インターネット上のメディアの総称です。代表的なものに、ブログやSNS、掲示板、動画投稿サイトなどがあります。

- × ア インターネットオークションの説明です。
- イ 正解です。
- × ウ オンラインモールの説明です。
- × エ メールマガジンの説明です。

覚えよう!

問56

CGM といえば

- 消費者生成メディア
- 個人が自ら発信するブログや SNS などの Web サイト

問57 ディープラーニング

■■■ 20%

ディープラーニング（深層学習）とは、コンピュータによる機械学習の技術です。たとえば、コンピュータに「画像に何が写っているのか」を認識させる場合、従来は画像の特徴を人間が数値化して入力していく作業が必要でした。ディープラーニングは、大量の画像データをコンピュータが自ら解析することで、認識能力を向上させるのが特徴です。車載機器では、歩行者と車をより確実に見分けるために、ディープラーニングの技術が利用できます。

以上から、正解は イ です。

問58 財務諸表 キホン!

■■■ 30%

財務諸表のうち、ある時点における企業の財政状態を、資産と負債・純資産に分けて表示したものを、**貸借対照表**（バランスシート）といいます。

- × ア 株式資本等変動計算書は、貸借対照表の純資産の変動状況を表したもののです。
- × イ キャッシュフロー計算書は、ある会計期間における資金の増減を表したもののです。
- × ウ 損益計算書は、ある時点における収益と費用の状態を表したもののです。
- エ 正解です。

貸借対照表

問58

資産の部		負債及び純資産の部	
期初	期末	期初	期末
流動資産	3,210	主な支払手形・買掛金	2,120
現金及び預金	675	支払手形・買掛金	1,230
受取手形・売掛金	165	定期預金	722
有価証券	152	未払法人会費	535
備蓄	47	未払法人会費	535
そ の 他	16	未払法人会費	535
固定資産	6,865	社内報酬金引当金	1,745
有形固定資産	5,965	社内報酬金引当金	1,745
建物・構築物	1,755	社内報酬金引当金	1,745
機械及び装置	844	社内報酬金引当金	1,745
土 地	350	資本金	4,488
そ の 他	73	法定準備金	3,000
無形固定資産	1,250	法定準備金	3,000
商 戸	30	法定準備金	3,000
投 資 有 価 純資産	合計 9,906	純資産合計	5,418
資 産 合 計	9,906	負債及び純資産合計	9,906

解答

- | | |
|-------|-------|
| 問55 ア | 問56 イ |
| 問57 イ | 問58 エ |

問 59

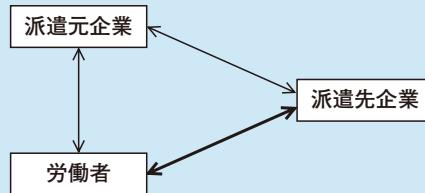
ある工場では表に示す3製品を製造している。実現可能な最大利益は何円か。ここで、各製品の月間需要量には上限があり、また、製造工程に使える工場の時間は月間200時間までで、複数種類の製品を同時に並行して製造することはできないものとする。

	製品 X	製品 Y	製品 Z
1個当たりの利益（円）	1,800	2,500	3,000
1個当たりの製造所要時間（分）	6	10	15
月間需要量上限（個）	1,000	900	500

- ア** 2,625,000 **イ** 3,000,000 **ウ** 3,150,000 **エ** 3,300,000

問 60

労働者派遣法に基づく、派遣先企業と労働者との関係（図の太線部分）はどれか。



- ア** 請負契約関係
イ 雇用契約関係
ウ 指揮命令関係
エ 労働者派遣契約関係



問59 最大利益の計算

■ 10%

製造時間が限られているので、製造所要時間当たりの利益が大きい製品を優先して作ったほうが、全体の利益も大きくなります。製造時間 1 分当たりに生じる利益は、

$$\text{製品 X} : 1,800 \text{ 円} \div 6 \text{ 分} = 300 \text{ 円/分}$$

$$\text{製品 Y} : 2,500 \text{ 円} \div 10 \text{ 分} = 250 \text{ 円/分}$$

$$\text{製品 Z} : 3,000 \text{ 円} \div 15 \text{ 分} = 200 \text{ 円/分}$$

で、製品 X が最も大きくなります。したがって、まず製品 X を優先して作ります。

製造工程に使える時間は、最大 200 時間 = $200 \times 60 = 12,000$ 分です。製品 X を需要上限である 1,000 個作ると、所要時間は $6 \text{ 分} \times 1,000 \text{ 個} = 6,000$ 分。製造時間の残りは $12,000 - 6,000 = 6,000$ 分になります。

残った時間で、製造時間当たりの利益が X の次に大きい製品 Y を作ります。製品 Y を需要上限の 900 個作ると、所要時間は $10 \text{ 分} \times 900 \text{ 個} = 9,000$ 分になります。残り時間をオーバーしてしまいます。そこで、残り時間 6,000 分をすべて製品 Y の製造にあてると、製品 Y は $6,000 \text{ 分} \div 10 \text{ 分} = 600$ 個作れます。以上で残り時間はゼロになるので、製品 Z を作る時間はありません。

以上から、利益が最大になるのは製品 X を 1,000 個、製品 Y を 600 個、製品 Z を 0 個製造したときで、その金額は

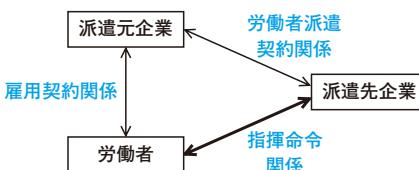
$$1,800 \text{ 円} \times 1,000 \text{ 個} + 2,500 \text{ 円} \times 600 \text{ 個} = 3,300,000 \text{ 円}$$

となります。正解は **工** です。

問60 労働者派遣契約 キホン!

■ 10%

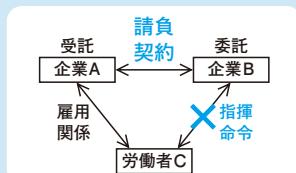
労働者派遣では、労働者は派遣元企業と雇用契約関係を結び、派遣先企業の指揮命令の下で働きます。派遣元企業、派遣先企業、労働者の関係は次のようにになります。



以上から、派遣先企業と労働者の関係は **ウ** の指揮命令関係です。

請負契約

問 60
企業が他の企業から業務を受託して遂行すること。



○ 解答 ○

問59 工 問60 ウ

基本情報技術者試験 予想問題②

科目B



問

01

次の記述中の [] に入る正しい答えを、解答群の中から選べ。

次のプログラムの出力結果は、“ [] ” となる。

[プログラム]

```
整数型 : x, y, z
x ← 5
y ← x + 3
if (y が 8 より大きい)
    y ← y - 1
else
    y ← y + 1
endif
z ← y mod 7
変数 y, z の値をカンマで区切って出力する
```

解答群

ア 7, 1 イ 7, 2 ウ 7, 6 エ 9, 1 オ 9, 2 カ 9, 6

※オリジナル問題

プログラムの解説

```
01 整数型 : x, y, z ← 变数の宣言
02 x ← 5 ← 变数への値の代入
03 y ← x + 3
04 if (y が 8 より大きい)
05     y ← y - 1
06 else
07     y ← y + 1
08 endif
09 z ← y mod 7 ← 剰余演算子
10 变数 y, z の値をカンマで区切って出力する ← このように、処理の内容が記述される場合もある。
```

01 | 変数の宣言

プログラム内で使用する変数は、

型名 : **変数名**

の型式で宣言します。同じ型名の変数は、カンマで区切って複数指定できます。本問では、**x**, **y**, **z** という整数型の変数を宣言しています。

なお、変数の宣言は、C 言語や Java など実際のプログラム言語でも必要な場合が多いですが、Python などのように変数の宣言が必要ない場合もあります。

02 | 変数への値の代入

情報処理試験の擬似言語では、変数への値の代入に代入演算子「←」を使います。変数への値の代入では、右辺の式の値を、左辺の変数に代入します。

例 : $x \leftarrow 5$ ← 変数 **x** に値 5 を代入する
 $y \leftarrow y - 1$ ← 変数 **y** の値から 1 を引いた値を、変数 **y** に代入する

なお、実際のプログラム言語では、多くの場合「=」が代入演算子として使われています。

09 | 剰余演算子

「mod」は剰余演算子で、整数 **a** を整数 **b** で割った余りを求めます。

例 : $z \leftarrow y \bmod 7$ ← 変数 **y** の値を 7 で割った余りを変数 **z** に代入する

問題解説

プログラムを上から順に読んで、変数の値の変化を追っていきましょう。

行番号 02 : 変数 **x** に値 5 が代入されます。

行番号 03 : 変数 **y** に値 $x + 3$ が代入されます。**x** の値は 5 なので、**y** の値は $5 + 3 = 8$ になります。

行番号 04 : 変数 **y** の値が 8 より大きいなら行番号 05、そうでないなら行番号 07 に分岐します。**y** の値は 8 なので、行番号 07 に分岐します。

行番号 07 : 変数 **y** の値に 1 を加えた値が、変数 **y** に代入されます。**y** の値は 8 なので、 $8 + 1 = 9$ が新しい **y** の値になります。

行番号 09 : **y** の値を 7 で割った余りが、変数 **z** に代入されます。**y** の値は 9 になっているので、 $9 \div 7$ の余り 2 が **z** に代入されます。

以上から、変数 **y** の値は 9、変数 **z** の値は 2 になっています。したがって、行番号 10 では「9, 2」が outputされます。正解は **オ** です。

文章でていねいに書くと面倒な感じがするけど、プログラムを行ごとに追っていくのは、慣れればそれほど大変な作業ではないよ。



○ 解答 ○

問01 **オ**

**合格のカギ** 代入演算子

代入演算子は、一般的なプログラム言語では「=」を用いますが、情報処理試験の擬似言語では「←」を用います。これは「等しい」を表す比較演算子「=」と区別するためです。一方、C 言語や Java, Python では、「等しい」を表す比較演算子に「==」を使います。



02

プログラム中の **a** と **b** に入れる正しい答えの組合せを、解答群の中から選べ。

手続 printStars は、“☆” と “★” を交互に、引数 num で指定された数だけ出力する。ここで、引数 num の値が 0 以下のときは、何も出力しない。

[プログラム]

```
○printStars( 整数型 : num )
    整数型 : cnt ← 0
    文字列型 : starColor ← "SC1"
    [a]
        if (starColor が "SC1" と等しい)
            "☆" を出力する
            starColor ← "SC2"
        else
            "★" を出力する
            starColor ← "SC1"
        endif
        cnt ← cnt + 1
    [b]
```

解答群

	a	b
ア	do	while (cnt が num 以下)
イ	do	while (cnt が num より小さい)
ウ	while (cnt が num 以下)	endwhile
エ	while (cnt が num より小さい)	endwhile

※出典：IT パスポート試験プログラミング的思考力を問う擬似言語のサンプル問題

プログラムの解説

```

01 OprintStars( 整数型 : num ) ← 手続の宣言
02 整数型 : cnt ← 0
03 文字列型 : starColor ← "SC1"
04   a
05   if (starColor が "SC1" と等しい)
06     "☆" を出力する
07     starColor ← "SC2"
08   else
09     "★" を出力する
10     starColor ← "SC1"
11   endif
12   cnt ← cnt + 1
13   b

```

01 | 手続の宣言

先頭に○記号の付いた行は、関数または手続の宣言です。情報処理試験の擬似言語では、手続を次のように定義します。

○ 手続名 (引数の型名) : 引数名, …)

キーワード	説明
手続名	手続の名前（本問では printStars）
引数の型名	手続に指定する引数の型名（本問では整数型）
引数名	手続に指定する引数（本問では num）。引数は複数指定できる。 また、引数がない場合は省略できる。

なお、関数には戻り値がありますが、手続には戻り値がありません。

問題解説

手続 printStars は、“★”または“☆”を引数に指定された回数交互に出力します。このような処理には繰返し処理を使います。

■ while ~ endwhile と do ~ while の違い

擬似言語の繰返し処理の構文には、

- while ~ endwhile
- do ~ while
- for ~ endfor

の3種類があります。どれを使っても同じ処理ができますが、解答群には for ~ endfor の選択肢はありません。

while ~ endwhile 構文と do ~ while 構文の違いは、繰返しを継続するかどうかの判定を、繰返しの前に行うか (while ~ endwhile)、繰返しの最後に行うか (do ~ while) の違いになります（合格

の力ギ参照)。そのため、do～while では最低 1 回はかならず繰返し処理が実行されます。

本問の場合「引数 num の値が 0 以下のときは、何も出力しない。」という指定があるので、繰返し処理が 1 回も実行されない場合があると判断できます。したがって、**ウ** または **エ** の while～endwhile 構文を使います。

■ 条件式の指定

繰返しを継続するかどうかは、変数 cnt の値を引数 num と比較して判断します。たとえば、引数 num の値が 0 のとき、繰返し処理は 1 回も実行されません。変数 cnt の初期値は 0 なので、

```
while (cnt が num 以下)
    繰返し処理
endwhile
```

とすると、繰返し処理が 1 回実行されてしまいます。そこで、

```
while (cnt が num より小さい)
    繰返し処理
endwhile
```

とすれば、cnt の初期値 0 は num より小さく、繰返し処理は実行されなくなります。

以上から、空欄 a は「while (cnt が num より小さい)」になります。正解は **エ** です。

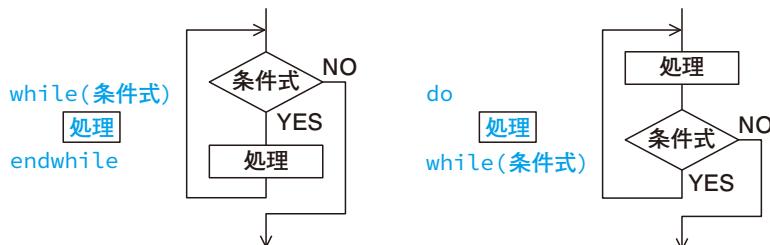
解答
問02 **エ**



合格の力ギ

while～endwhile と do～while

while～endwhile と do～while は、どちらも条件式が真の間、処理を繰返し実行します。



2つの構文の違いは、while～endwhile が繰返し処理の入口で条件式を評価するのに対し、do～while は、繰返し処理の出口で条件式を評価することです。そのため、while～endwhile では繰返し処理が 1 度も実行されないことがあるのに対し、do～while では最低 1 回はかならず繰返し処理が実行されます。

一般に、do～while は while～endwhile に書き替えることができます。そのためプログラム言語によっては、do～while に当たる構文がないものもあります (Python など)。



03

次のプログラム中の [] に入る正しい答えを、解答群の中から選べ。

ある学校の授業の成績は、期末テストの得点が 80 点以上なら “A”，60 点以上 80 点未満なら “B”，60 点未満なら “C” となる。関数 grade は、得点を表す 0 以上の整数を引数として受け取り、成績を “A”，“B”，“C” のいずれかで返す。

[プログラム]

```
○ 文字列型 : grade( 整数型 : score )
    文字列型 : ret
    if (score が 80 以上)
        ret ← "A"
    elseif ([ ])
        ret ← "B"
    else
        ret ← "C"
    endif
    return ret
```

解答群

- ア (score が 60 より大きい) and (score が 80 より小さい)
- イ (score が 60 以上) or (score が 80 より小さい)
- ウ (score が 60 と等しい) or (score が 80 より小さい)
- エ score が 60 以上
- オ score が 60 より大きい
- カ score が 60 以下
- キ score が 60 より小さい

※オリジナル問題

プログラムの解説

```
01 ○ 文字列型 : grade( 整数型 : score )
02 文字列型 : ret
03 if (score が 80 以上 )
04     ret ← "A"
05 elseif ( [ ] )
06     ret ← "B"
07 else
08     ret ← "C"
09 endif
10 return ret
```

} **if～elseif～else 構文**

| **if～elseif～else 構文**

この構文では、条件式を上から順番に評価し、最初に真（true）になった条件式に対応する処理を実行します。真になった条件式以降の条件式は評価せず、対応する処理も実行しません。条件式がどれも真にならなかったときは、else に対応する処理 n + 1 を実行します。

```
if [条件式 1]
    [処理 1] ← 条件式 1 が真のとき実行
elseif [条件式 2]
    [処理 2] ← 条件式 2 が真のとき実行
...
elseif [処理 n]
    [処理 n] ← 条件式 n が真のとき実行
else
    [処理 n + 1] ← 条件式 1 ~ n がいずれも真でないとき実行
endif
```

elseif と処理の組は、最後に 1 つだけ記述することができます。また必要なければ省略することもできます。

問題解説

行番号 03, 04：条件式「score が 80 以上」が真のとき、変数 ret には文字「A」が代入されます。この処理は、期末テストの得点が「80 点以上なら “A”」となることに対応します。

```
03 if (score が 80 以上 )
04     ret ← "A"
```

行番号 05, 06：空欄の条件式が真のとき、変数 ret には文字「B」が代入されます。この処理は、期末テストの得点が「60 点以上 80 点未満なら “B”」となることに対応します。したがって、空欄に入る条件式は、次のように書けます。

(score が 60 以上) and (score が 80 より小さい)

ただし、この条件式は、行番号 03 の条件式「score が 80 以上」が偽のときにしか評価されません。つまり「score が 80 より小さい」ときにしか評価されないので、わざわざ「score が 80 より小さい」という条件を付ける必要はないのです。したがって空欄に入る条件式としては、次のように指定すればじゅうぶんということになります。

```
05  elseif (score が 60 以上)
06      ret ← "B"
```

以上から、正解は **工** です。

行番号 07, 08：行番号 03, 行番号 05 の条件式がいずれも真でなかったとき、すなわち score の値が 60 未満のときは、変数 ret に文字「C」が代入されます。この処理は、期末テストの得点が「60 点未満なら “C”」となることに対応します。

○ 解答 ○

問03 工



合格の力ギ

elseifについて

C 言語や Java には、if～else 構文はありますですが、elseif はありません。if～else を入れ子にすることによって、elseif と同じ処理ができるからです。

if～else を入れ子にすれば elseif と同種になる。

```
if 条件 1
    処理 1
else
    if 条件 2
        処理 2
    else
        if 条件 3
            処理 3
        else
            :
:
```



```
if 条件 1
    処理 1
else if
    条件 2
        処理 2
    else if
        条件 3
            処理 3
    else
        :
:
```



04

次の記述中の [] に入る正しい答えの組合せを、解答群の中から選べ。

関数 `permutation` は、2個の正の整数 n, r を引数として受け取り、 n 個から r 個選んで並べる順列 ${}_nP_r$ を計算する。また、関数 `combination` は、2個の正の整数 n, r を引数として受け取り、 n 個から r 個選ぶ組合せ ${}_nC_r$ を計算する。ここで、順列 ${}_nP_r$ と組合せ ${}_nC_r$ は、それぞれ次の式で計算することができる。

$${}_nP_r = \frac{n!}{(n-r)!}, \quad {}_nC_r = \frac{n!}{r!(n-r)!}, \text{ ただし, } n! \text{ は } n \text{ の階乗を表す}$$

関数 `fact` は、0以上の整数 n を引数として受け取り、 n の階乗 $n!$ を返す。関数 `combination` を、`combination(5, 2)` のように呼び出したとき、プログラム1の α の行は、全部で [] 回実行される。

[プログラム1]

```
○整数型 : fact( 整数型 : n )
    整数型 : i, ret
    ret ← 1
    for (i を 1 から n まで 1ずつ増やす)
        ret ← ret × i ← α
    endfor
    return ret
```

[プログラム2]

```
○整数型 : permutation( 整数型 : n, 整数型 : r )
    整数型 : ret ← 0
    if (n ≥ r)
        ret ← fact(n) ÷ fact(n - r) の商
    endif
    return ret
```

[プログラム3]

```
○整数型 : combination( 整数型 : n, 整数型 : r )
    整数型 : ret ← 0
    if (n ≥ r)
        ret ← permutation(n, r) ÷ fact(r) の商
    endif
    return ret
```

解答群

ア 5	イ 6	ウ 7	エ 8	オ 9
力 10	キ 15	ク 20	ケ 25	コ 30

※オリジナル問題

プログラムの解説

[プログラム 1]

```

01 ○ 整数型 : fact( 整数型 : n )
02   整数型 : i, ret
03   ret ← 1
04   for (i を 1 から n まで 1ずつ増やす)
05     ret ← ret × i ← α } for構文
06   endfor
07   return ret

```

[プログラム 2]

```

01 ○ 整数型 : permutation( 整数型 : n, 整数型 : r )
02   整数型 : ret ← 0
03   if (n ≥ r)
04     ret ← fact(n) ÷ fact(n - r) の商
05   endif
06   return ret ← return文

```

[プログラム 3]

```

01 ○ 整数型 : combination( 整数型 : n, 整数型 : r )
02   整数型 : ret ← 0
03   if (n ≥ r)
04     ret ← permutation(n, r) ÷ fact(r) の商
05   endif
06   return ret ← return文

```

| for 構文

for 構文は、for～endfor の間の処理を、指定した条件にしたがって繰り返します。たとえば、プログラム 1 の行番号 04 では次のようにになります。

繰返しごとに値を変化させる変数

↓
 for (i を 1 から n まで 1ずつ増やす)
 ↑ ↑ ↑
 初期値 終値 増分値

変数は繰返し 1 回ごとに、指定した値だけ変化します。ここでは、変数 i の値は 1, 2, 3, … のように 1 ずつ増え、n になったときが最後の繰返しになります。すなわち、繰返し回数は全部で n 回になります。

return文

本問のプログラムでは、fact, permutation, combinationという3つの関数が定義されています。関数とは、引数に指定したデータにもとづいて何らかの処理をし、その結果を呼出し元に渡す手続きです。呼出し元に渡す値のことを戻り値といいます。

return文は戻り値を関数の呼び出し元のプログラムに渡し、関数の処理を終了します。たとえば、プログラム1の行番号07では、変数retの値を返します。

return 戻り値

関数の呼出し

関数は呼出し元のプログラムで、「[関数名] ([引数])」のように指定して呼び出します。たとえば、プログラム2の行番号04

04 ret ← fact(n) ÷ fact(n - r) の商

では、関数fact(n)とfact(n - r)を呼び出し、それぞれの戻り値を使って「fact(n)の戻り値 ÷ fact(n - r)の戻り値の商」を求め、その結果を変数retに代入します。

問題解説

関数combinationを、combination(5, 2)のように呼び出すと、プログラム3の行番号04

04 ret ← permutation(n, r) ÷ fact(r) の商

が実行され、関数permutationと関数factがそれぞれpermutation(5, 2), fact(2)のように呼び出されます。

関数permutationを、permutation(5, 2)のように呼び出すと、プログラム2の行番号04

04 ret ← fact(n) ÷ fact(n - r) の商

が実行され、関数factがfact(5), fact(3)のように呼び出されます。

関数factは、

04 for (i を 1 から n まで 1ずつ増やす)
05 ret ← ret × i ← α
06 endfor

のように、αの行を引数nに指定された回数だけ繰り返します。したがって、αの行は、fact(5)で5回、fact(3)で3回、fact(2)で2回実行されます。

以上からαの行の実行回数は合計で $5 + 3 + 2 = 10$ 回になります。正解は力です。

解答

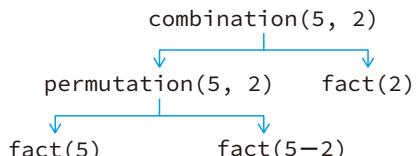
問04 力



合格の力

関数呼出しの構造

関数呼出しの構造を次のように理解すれば、解答しやすくなります。





問 05

次の記述中の [a], [b] に入れる正しい答えの組合せを、解答群の中から選べ。

1 から 100 までの 3 の倍数または 7 の倍数のうち、2 の倍数でも 5 の倍数でもない数はいくつあるかを求め、その個数を出力するプログラムである。

[プログラム]

```
整数 : i, count ← 0
for (i を 1 から 100 まで 1 ずつ増やす)
    if ([a])
        if ([b])
            count ← count + 1
        endif
    endif
endfor
count を出力する
```

解答群

	a	b
ア	$(i \bmod 3 = 0) \text{ and } (i \bmod 7 = 0)$	$(i \bmod 2 \neq 0) \text{ and } (i \bmod 5 \neq 0)$
イ	$(i \bmod 3 = 0) \text{ and } (i \bmod 7 = 0)$	$(i \bmod 2 \neq 0) \text{ or } (i \bmod 5 \neq 0)$
ウ	$(i \bmod 3 = 0) \text{ and } (i \bmod 7 = 0)$	$\text{not } ((i \bmod 2 = 0) \text{ and } (i \bmod 5 = 0))$
エ	$(i \bmod 3 = 0) \text{ or } (i \bmod 7 = 0)$	$(i \bmod 2 \neq 0) \text{ or } (i \bmod 5 \neq 0)$
オ	$(i \bmod 3 = 0) \text{ or } (i \bmod 7 = 0)$	$\text{not } ((i \bmod 2 = 0) \text{ or } (i \bmod 5 = 0))$
カ	$(i \bmod 3 = 0) \text{ or } (i \bmod 7 = 0)$	$\text{not } ((i \bmod 2 = 0) \text{ and } (i \bmod 5 = 0))$

※オリジナル問題

プログラムの解説

```
01 整数 : i, count ← 0
02 for (i を 1 から 100 まで 1 ずつ増やす)
03   if ([ a ])
04     if ([ b ])
05       count ← count + 1 ←----- 個数を教える
06   endif
07 endif
08 endfor
09 count を出力する
```

論理演算子

擬似言語の論理演算子には、次のような種類があります。

論理演算子	説明
A and B (論理積)	A と B が両方とも真 (true) のとき真、さもなければ偽 (false) を返す。
A or B (論理和)	A と B の少なくとも一方が真 (true) なら真、両方とも偽のときは偽 (false) を返す。
not A (否定)	A が真 (true) なら偽 (false), A が偽 (false) なら真 (true) を返す。

問題解説

プログラムは、1 から 100 までの整数について、条件に適合するかどうかを順に調べ、適合する場合は変数 count を 1 増やしてその個数を数えます。その条件は、

「3 の倍数または 7 の倍数のうち、2 の倍数でも 5 の倍数でもない」

です。ここでは、この条件を①3 の倍数または 7 の倍数と、②2 の倍数でも 5 の倍数でもないの 2 段階に分けて考えます。

①3 の倍数または 7 の倍数

変数 i の値がある数の倍数かどうかは、変数 i の値をその数で割った余りが 0 かどうかで判断できます。したがって、3 の倍数と 7 の倍数は、それぞれ次のような式で判断できます。

3 の倍数 : $i \bmod 3 = 0$

7 の倍数 : $i \bmod 7 = 0$

3の倍数か7の倍数かはどちらか一方でいいので、2つの式は論理演算子 or で結びます。

$(i \bmod 3 = 0) \text{ or } (i \bmod 7 = 0)$

← 空欄a

② 2の倍数でも5の倍数でもない

「2の倍数でない」と「5の倍数でない」は、それぞれ「2で割った余りが0でない」「5で割った余りが0でない」と同じなので、次のように書けます。

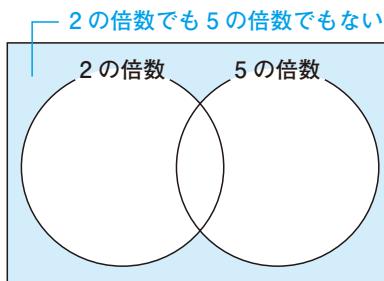
2の倍数でない： $i \bmod 2 \neq 0$ 5の倍数でない： $i \bmod 5 \neq 0$

「2の倍数でも5の倍数でもない」は、2つの式を論理式 and で結び、次のように書けばOKです。

$(i \bmod 2 \neq 0) \text{ and } (i \bmod 5 \neq 0)$

← 空欄b?

ところが、空欄aが「 $(i \bmod 3 = 0) \text{ or } (i \bmod 7 = 0)$ 」、空欄bが「 $(i \bmod 2 \neq 0) \text{ and } (i \bmod 5 \neq 0)$ 」の組合せは、解答群にありません。そこで、上の式をもう一度検討しましょう。「2の倍数でも5の倍数でもない」は、ベン図を使うと次の図の色網の部分になります。



上の図で、色網以外の部分は「2の倍数または5の倍数」を表すので、「2の倍数でも5の倍数でもない」は、「2の倍数または5の倍数」の否定として表すことができます。

否定

\downarrow
 $\text{not } ((i \bmod 2 = 0) \text{ or } (i \bmod 5 = 0))$

← 空欄b

2の倍数または5の倍数

以上から、空欄aが「 $(i \bmod 3 = 0) \text{ or } (i \bmod 7 = 0)$ 」、空欄bが「 $\text{not } ((i \bmod 2 = 0) \text{ or } (i \bmod 5 = 0))$ 」の組合せの
オが正解です。

○ 解答 ○

問05 オ

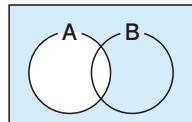


合格の力ギ

ド・モルガンの法則

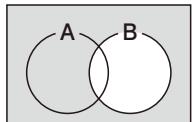
一般に、 $(\text{not } A) \text{ and } (\text{not } B)$ は、 $\text{not}(A \text{ or } B)$ と書き換えることができます。これをド・モルガンの法則といいます。

$\text{not } A$



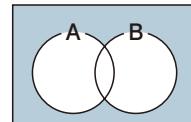
and

$\text{not } B$



=

$\text{not } (A \text{ or } B)$





06

次のプログラム中の と に入る正しい答えの組合せを、解答群の中から選べ。

元金 p 、年利 r の n 年後の元利合計は、次の式で表すことができる ($n \geq 0$)。

$$n \text{ 年後の元利合計} = p \times (1 + r)^n$$

関数 fv は、引数に元金 p 、年利 r 、年数 n を指定し、 n 年後の元利合計を返す。

(プログラム)

```
○実数型 : fv( 実数型 : p, 実数型 : r, 整数型 : n)
if ()
    return p
else
    return 
endif
```

解答群

	a	b
ア	n が 0 に等しい	$fv(p, r, n - 1) \times (1 + r)$
イ	n が 0 に等しい	$fv(p, r, n - 1) \times r$
ウ	n が 0 に等しい	$fv(p, r, n) \times (1 + r)$
エ	n が 0 に等しい	$fv(p, r, n) \times r$
オ	n が 1 以上	$fv(p, r, n - 1) \times (1 + r)$
カ	n が 1 以上	$fv(p, r, n - 1) \times r$
キ	n が 1 以上	$fv(p, r, n) \times (1 + r)$
ク	n が 1 以上	$fv(p, r, n) \times r$

※オリジナル問題

プログラムの解説

```
01 ○ 実数型 : fv( 実数型 : p, 実数型 : r, 整数型 : n)
02 if ()
03     return p
04 else
05     return 
06 endif
```

再帰的プログラム

関数や手続の処理の中で、自分自身を呼び出すプログラムを**再帰的プログラム**といいます。たとえば関数 a の処理の中でさらに関数 a が呼び出され、その処理の中でさらに関数 a が呼び出され…のように、何段にも入れ子になって関数が呼び出されるプログラムです。



問題解説

行番号 02：再帰的プログラムでは、「これ以上は自分自身を呼び出さない」という条件がからならず設定されます。この条件がないと、無限に自分自身を呼び出し続けることになってしまふからです。

元利合計は、年数が0の場合は $p \times (1 + r)^0 = p$ となり、元金pのままとなります。したがって空欄aは、年数nが0かどうかを調べる条件式が入ります。

02 if (n が 0 に等しい) ← 空欄a
03 return p

行番号 05：元利合計の計算式「n年後の元利合計 = $p \times (1 + r)^n$ 」は、次のように変形できます。

$$\text{n年後の元利合計} = p \times (1 + r)^n = \underline{p \times (1 + r)^{n-1}} \times (1 + r)$$

上の式の下線部分は、n-1年後の元利合計を表すので、

$$\text{n年後の元利合計} = n - 1 \text{年後の元利合計} \times (1 + r)$$

となります。「n-1年後の元利合計」は、関数fvを $\text{fv}(p, r, n-1)$ のように呼び出して求めることができます。行番号05は次のようにになります。

05 return fv(p, r, n-1) × (1 + r)
↑ 空欄b

以上から、空欄aが「nが0に等しい」、空欄bが「 $\text{fv}(p, r, n-1) \times (1 + r)$ 」の組合せのアが正解です。

解答

問06 ア



合格のカギ

再帰的関数と非再帰的関数

一般に、再帰的プログラムは非再帰的プログラムに書き換えることができます。たとえば、非再帰的な関数fvは、次のようにになります。

○実数型：fv(実数型：p, 実数型：r, 整数型：n)
実数型：ret
整数型：i
ret ← p
for (i を 1 から n まで 1 ずつ増やす)
 ret ← ret × (1 + r)
endfor
return ret



07

次の記述中の **a** と **b** に入れる正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まる。

関数 `bsearch` は、配列 `array` から引数 `key` を探索し、見つかった場合にはその要素番号を返す。また、見つからなかった場合には `0` を返す。配列 `array` の内容が `{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}` であるとき、関数 `bsearch` を `bsearch(6)` をのように呼び出した。この関数が戻り値 `7` を返すまでに、 α の行は **a** 回、 β の行は **b** 回実行される。

(プログラム)

大域： 整数型の配列： `array` $\leftarrow \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

```
○整数型： bsearch( 整数型： key )
    整数型： lo, hi, x
    lo  $\leftarrow 1$ 
    hi  $\leftarrow$  array の要素数
    while (lo が hi 以下)
        x  $\leftarrow (\text{lo} + \text{hi}) \div 2$  の商
        if (array[x] が key と等しい)
            return x
        elseif (array[x] が key より小さい)
            lo  $\leftarrow x + 1$  ← α
        else
            hi  $\leftarrow x - 1$  ← β
        endif
    endwhile
    return 0
```

解答群

	a	b
ア	0	1
イ	0	2
ウ	1	0
エ	1	1
オ	1	2
カ	2	0
キ	2	1
ク	2	2

※オリジナル問題

プログラムの解説

```

01 大域： 整数型の配列： array ← {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
02 ○整数型： bsearch( 整数型： key )
03   整数型： lo, hi, x
04   lo ← 1
05   hi ← array の要素数
06   while (lo が hi 以下)
07     x ← (lo + hi) ÷ 2 の商
08     if (array[x] が key と等しい)
09       return x
10     elseif (array[x] が key より小さい)
11       lo ← x + 1 ← α
12     else
13       hi ← x - 1 ← β
14     endif
15   endwhile
16   return 0 ← 見つからなかった場合

```

二分探索法

関数 `bsearch` は、[二分探索法](#)というアルゴリズムを使って配列を探索します。二分探索法は、あらかじめ整列済みの配列に対して、1回の探索ごとに探索範囲を半分にせばめていく探索アルゴリズムです。

行番号 04, 05：変数 `lo` は探索範囲の先頭、変数 `hi` は探索範囲の末尾の要素番号を指します。最初は配列全体を調べるので、`lo` の初期値は 1、`hi` の初期値は配列の末尾になります。

```

04   lo ← 1
05   hi ← array の要素数

```

行番号 06～09：探索範囲が残っている間、探索を繰り返します。探索範囲の真ん中の要素番号を変数 `x` に設定し、`array[x]` が目的の値に等しければ、`x` の値を返します。

```

06   while (lo が hi 以下)
07     x ← (lo + hi) ÷ 2 の商
08     if (array[x] が key と等しい)
09       return x

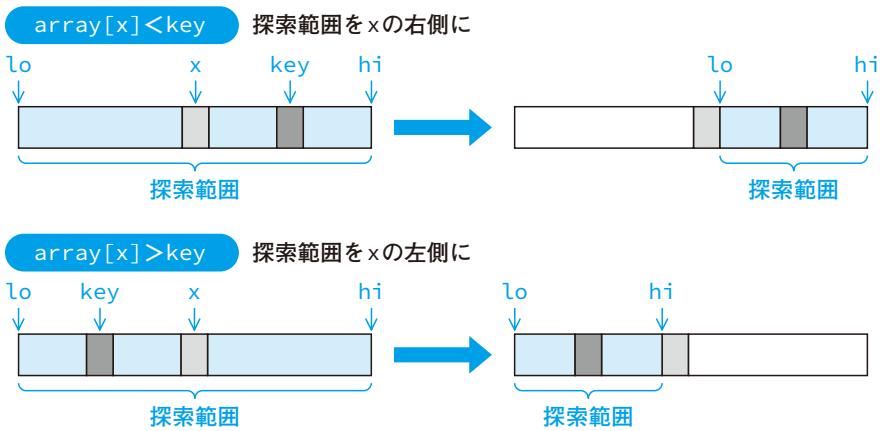
```

行番号 10～13：`array[x]` が目的の値より小さいなら、目的の値は `array[x]` より右側にあります。したがって、探索範囲の始点を `x` の1つ後ろに移動します。また、`array[x]` が目的の値より大きいなら、目的の値は `array[x]` より左側にあります。したがって、探索範囲の終点を `x` の1つ前に移動します。

```

10   elseif (array[x] が key より小さい)
11     lo ← x + 1 ← 探索範囲を x の右に
12   else
13     hi ← x - 1 ← 探索範囲を x の左に

```

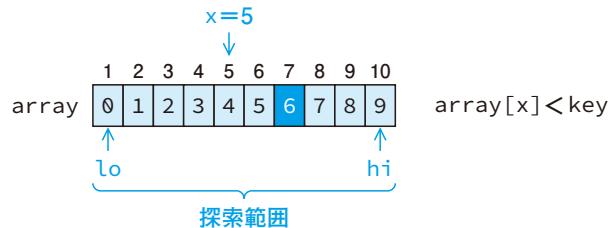


このように、探索範囲を半分にして、もう一度探索を行います。

問題解説

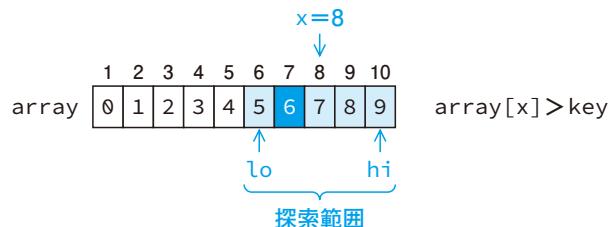
`bsearch(6)` を実行した場合の探索範囲は、次のようにになります。

① 1回目の探索 : $x \leftarrow (1 + 10) \div 2$



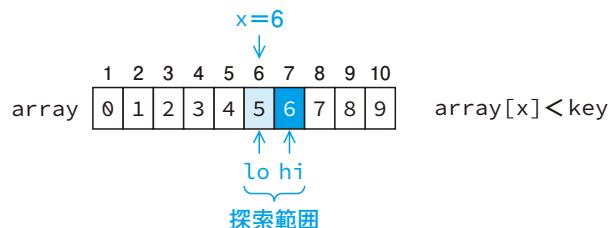
`array[x]` は 6 より小さいので、探索範囲の始点 `lo` が移動します (α)。

② 2回目の探索 : $x \leftarrow (6 + 10) \div 2$



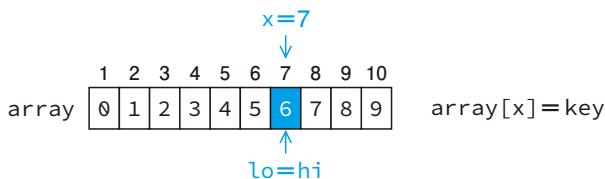
array[x] は 6 より大きいので、探索範囲の終点 hi が移動します (β)。

③ 3回目の探索 : $x \leftarrow (6 + 7) \div 2$



array[x] は 6 より小さいので、探索範囲の始点 lo が移動します（ α ）。

④ 4回目の探索 : $x \leftarrow (7 + 7) \div 2$



目的の値が見つかったので、探索を終了します。

以上から、 α の行は2回、 β の行は1回実行されます。正解は **キ** です。

○ 解答 ○

問07 **キ**



合格の力ギ

二分探索法の効率

たとえば、要素数が 1,024 個の配列を探索する場合、先頭から順番に探索する方法では、最悪の場合 1,024 回の比較が必要になります。一方、二分探索法では、1 回の探索ごとに探索範囲が半分になるため、 $1,024 \rightarrow 512 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ と、どんなに多くても 11 回の比較で目的の値に到達します。



08

次のプログラム中の に入る正しい答えの組合せを、解答群の中から選べ。

中置記法で $(a + b) \div (c - d) \times (e + f)$ と表される算術式が、図1の二分木に格納されている。手続 `traverse` は、引数として図1の二分木の“ \times ”の節への参照を受け取り、算術式を後置記法（逆ポーランド記法）で出力する。

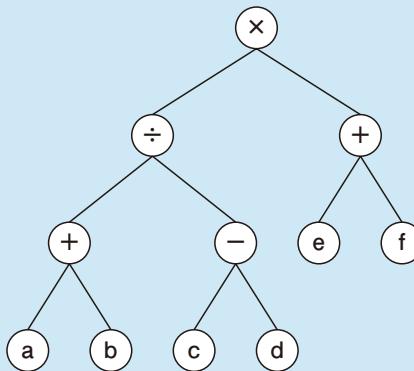


図1 二分木に格納された算術式

二分木の節は、クラス `Node` を用いて表現する。クラス `Node` の説明を図2に示す。`Node` 型の変数は、クラス `Node` のインスタンスの参照を格納する。

メンバ変数	説明
文字型 : <code>val</code>	節の値を格納する。
<code>Node</code> : <code>left</code>	節の左の子への参照を格納する。左の子がない場合は未定義となる。
<code>Node</code> : <code>right</code>	節の右の子への参照を格納する。左の子がない場合は未定義となる。

図2 クラス `Node` の説明

〔プログラム〕

```
O traverse (Node: node)
    if (node が 未定義)
        return
    else
        
    endif
```

解答群

- ア node.val を出力する
traverse(node.left)
traverse(node.right)
- イ node.val を出力する
traverse(node.right)
traverse(node.left)
- ウ traverse(node.left)
node.val を出力する
traverse(node.right)
- エ traverse(node.right)
node.val を出力する
traverse(node.left)
- オ traverse(node.left)
traverse(node.right)
node.val を出力する
- カ traverse(node.right)
traverse(node.left)
node.val を出力する

※オリジナル問題

プログラムの解説

```
01  O traverse (Node: node)
02      if (node が 未定義) ← 子がない節の場合
03          return
04      else
05          [ ]
06          ← 二分木の走査
07      endif
08  
```

二分木

複数の節（ノード）をツリー状に接続したデータ構造を**木構造**といいます。下位に接続された節を**子**、上位に接続された節を**親**といい、木構造の頂点にあるもっとも上位の節を**根**といいます。木構造のうち、2つ以上の子をもたないものをとくに**二分木**といいます。

二分木の走査

二分木を構成する各節を漏れなく訪ねることを走査といいます。二分木の走査には、①**間順走査**、②**前順走査**、③**後順走査**の3種類があります。ここでは間順走査について説明しましょう。

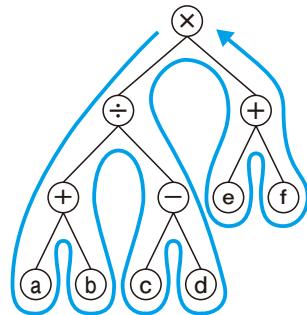
間順走査は、次のような順序で行います。

- ①現在の節の左側の子を根とする部分木を走査する
- ②現在の節の値を出力する
- ③現在の節の右側の子を根とする部分木を走査する

要は、①左の子→②自分→③右の子の順に走査します。

手順①と③は**再帰的**な手続きになっていることに注意しましょう。図1の二分木を上の手順に沿って走査すると、次のようにになります。

- ①“ \times ”の節の左側の子を根とする部分木を走査する
 - ①“ \div ”の節の左側の子を根とする部分木を走査する
 - ①“+”の節の左側の子を根とする部分木を走査する
 - ②“a”を出力する
 - ②“+”を出力する
 - ③“+”の節の右側の子を根とする部分木を走査する
 - ②“b”を出力する
- ②“ \div ”を出力する
- ③“ \div ”の節の右側の子を根とする部分木を走査する
 - ①“-”の節の左側の子を根とする部分木を走査する
 - ②“c”を出力する
 - ②“-”を出力する
 - ③“-”の節の右側の子を根とする部分木を走査する
 - ②“d”を出力する
- ②“ \times ”を出力する
- ③“ \times ”の節の右側の子を根とする部分木を走査する
 - ①“+”の節の左側の子を根とする部分木を走査する
 - ②“e”を出力する
 - ②“+”を出力する
 - ③“+”の節の右側の子を根とする部分木を走査する
 - ②“f”を出力する



以上から、値が出力される順序は $a + b \div c - d \times e + f$ となります。このように、間順走査した場合には一般に**中置記法**になります（中置記法では、必要に応じてカッコを補う必要があります）。

問題解説

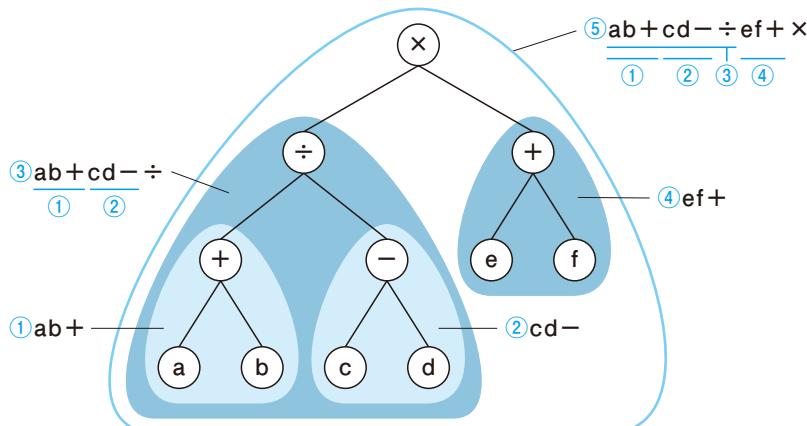
算術式 $(a + b) \div (c - d) \times (e + f)$ は、**後置記法**（逆ポーランド記法）で次のようにになります。

$ab + cd - \div ef + x$

部分木を左の子→右の子→自分の順に出力すれば、この順番が得られます。したがって空欄の処理は、

traverse(node.left) ← 左の子を走査
 traverse(node.right) ← 右の子を走査
 node.val を出力する ← この節の値を出力

となります。このような後順走査といいます。正解は **オ** です。

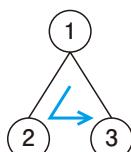


解答

問08 **オ**

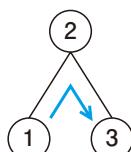
合格のカギ 二分木の走査

前順走査



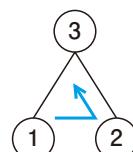
自分→左の子→右の子

間順走査



左の子→自分→右の子

後順走査



左の子→右の子→自分

解説のように、間順走査では中置記法になります。また、前順走査した場合には、前置記法（ポーランド記法）と呼ばれる算術式になりますが、この記法は使われていません。



問 09

次の記述中の [] に入る正しい答えを、解答群の中から選べ。

有向グラフの各頂点のラベルを順に出力するプログラムである。有向グラフの例を図に示す。
図はグラフの頂点を○で表し、頂点のラベルを○の中の文字で表している。

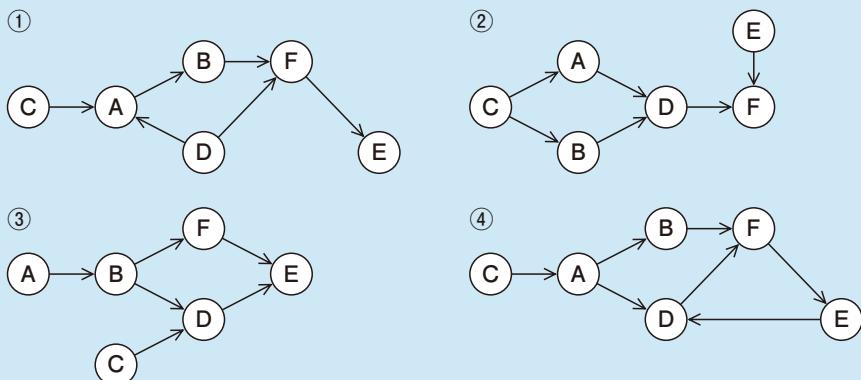


図 有向グラフの例

プログラムは、グラフの頂点をクラス `Vertex` のインスタンスで表し、そのラベルをクラス `Vertex` のメンバ変数 `label` で表す。このプログラムの出力結果が “CABDFE” となるグラフは、図の①～④のうち [] である。

解答群

ア ① イ ② ウ ③ エ ④

[プログラム]

```
文字列型の配列 : array[ グラフの頂点の数 ] /* 頂点の数を要素数とする配列 */
整数型 : count ← 1
Vertex: curr
while (count が グラフの頂点の数 以下)
    グラフから、自分を始点とする辺がひとつもない頂点を 1 つ選び、curr に格納する
    /* 見つからない場合、curr は 未定義とする */
    if (curr が 未定義の値 )
        エラーを出力して終了
    endif
    array[ グラフの頂点の数 - count + 1] ← curr.label
    グラフから curr を削除する
    count ← count + 1
endwhile
array の各要素を先頭から順に出力する
```

※オリジナル問題

プログラムの解説

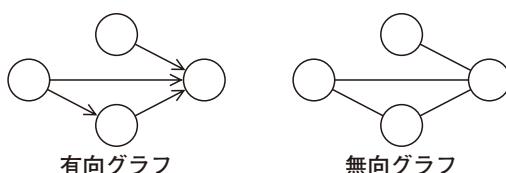
```

01 文字列型の配列： array[ グラフの頂点の数 ] /* 頂点の数を要素数とする配列 */
02 整数型： count ← 1
03 Vertex: curr
04 while (count が グラフの頂点の数 以下)
    グラフから、自分を始点とする辺がひとつもない頂点を 1 つ選び、 curr に格納する
    /* 見つからない場合、 curr は 未定義とする */
05     if (curr が 未定義の値 )
        エラーを出力して終了 ← 閉路が存在する場合
06     endif
07     array[ グラフの頂点の数 - count + 1] ← curr.label
08     グラフから curr を削除する
09     count ← count + 1
10 endwhile
11 array の各要素を先頭から順に出力する
12
13

```

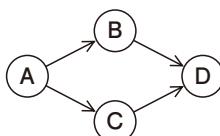
| 有向グラフ

グラフは、複数の頂点を辺（エッジ）で結んだデータ構造です。2つの頂点を結ぶ辺に、始点と終点の区別がある場合を**有向グラフ**、始点と終点の区別がない場合を**無向グラフ**といいます。

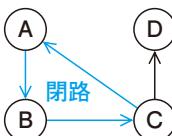


| トポロジカルソート

有向グラフの各頂点を、先行するものから順に出力することを**トポロジカルソート**といいます。たとえば、次のような有向グラフをトポロジカルソートすると、“ABCD”または“ACBD”が出力されます。



上のように、トポロジカルソートの結果は1通りとは限りません。また、次のように、閉路（ぐるっと回って元の頂点に戻ってくる経路）があるグラフは、トポロジカルソートができません。

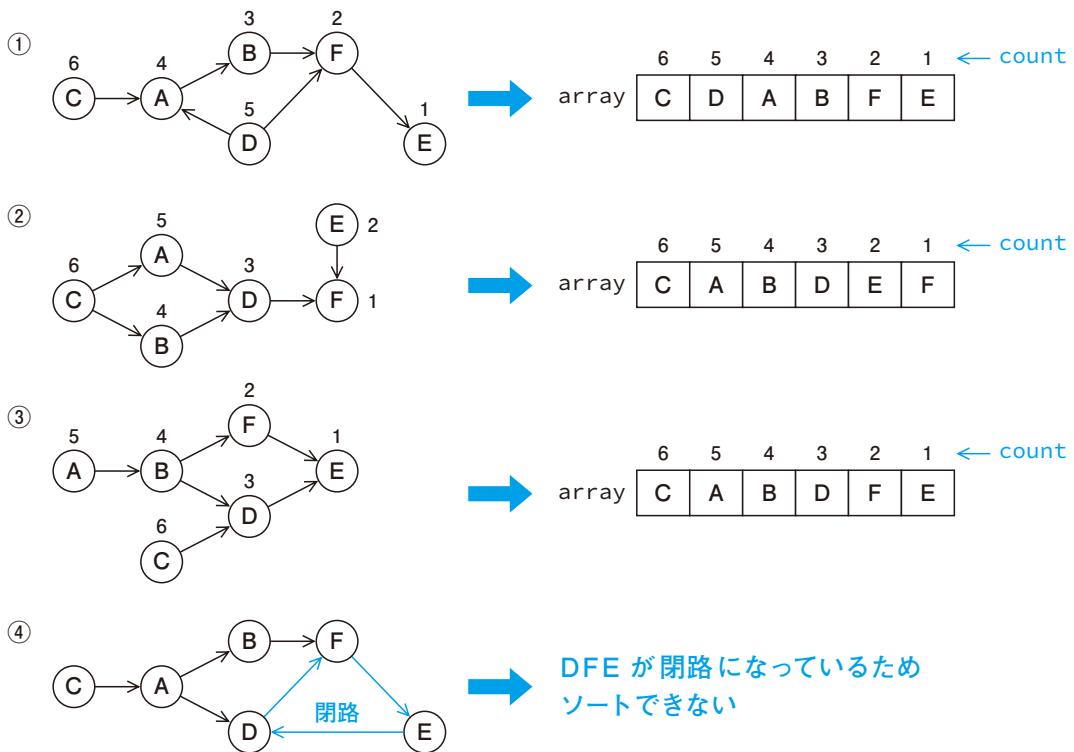


問題解説

本問のプログラムは、グラフをトポロジカルソートした結果を出力するプログラムです。プログラムは、次のような手順で頂点の順番を決めていきます。

- ①グラフの中から、始点となる辺が1つもない頂点を1つ選びます（行番号05）。複数ある場合はどれを選んでもかまいません。
- ②選んだ頂点のラベルを、配列arrayに後ろから順に追加します（行番号09）。
- ③選んだ頂点をグラフから削除します（行番号10）。
- ④以上の手順を、グラフから頂点がなくなるまで繰り返すと、配列arrayに頂点のラベルがトポロジカルソートの逆順に格納されます。

図の①～④のグラフを、上の手順にしたがってソートすると、次のようにになります。



以上から、結果が“CABDFE”となるグラフは③です。したがって、**ウ**が正解となります。

解答
問09 **ウ**



合格のカギ

グラフを使ったアルゴリズム

科目Aでよく出題されるアローダイアグラムは、有向グラフの一種です。トポロジカルソートは、アローダイアグラムから正しい工程の順序を見つける場合にも有用です。

なお、グラフの辺は二次元配列や単方向リストを使って表されます。これらについても学習しておきましょう。



10

次の記述中の **a** と **b** に入れる正しい答えの組合せを、解答群の中から選べ。

整数型のデータを格納するスタックとキューを、それぞれクラス **Stack** とクラス **Queue** で表現する。クラス **Stack** の説明を図 1、クラス **Queue** の説明を図 2 に示す。**Stack** 型、**Queue** 型の変数は、それぞれのクラスのインスタンスの参照を格納するものとする。

メソッド	説明
Stack()	空のスタックを作成するコンストラクタ。
push(整数型 : val)	引数 val をスタックに格納する。
整数型 : pop()	スタックからデータを取り出してその値を返す。

図 1 クラス **Stack** の説明

メソッド	説明
Queue()	空のキューを作成するコンストラクタ。
enq(整数型 : val)	引数 val をキューに挿入する。
整数型 : deq()	キューからデータを取り出してその値を返す。

図 2 クラス **Queue** の説明

次のプログラムを実行したとき、変数 **x** に代入される値は **a**、変数 **y** に代入される値は **b** となる。

〔プログラム〕

```
整数型 : x, y
Stack: stack ← Stack()
Queue: queue ← Queue()

stack.push(10)
stack.push(20)
queue.enq(stack.pop())
queue.enq(30)
stack.push(40)
stack.push(queue.deq())
x ← stack.pop()
y ← queue.deq()
```

解答群

	a	b
ア	10	20
イ	10	30
ウ	20	30
エ	20	40
オ	30	20
カ	30	40
キ	40	20
ク	40	30

※オリジナル問題

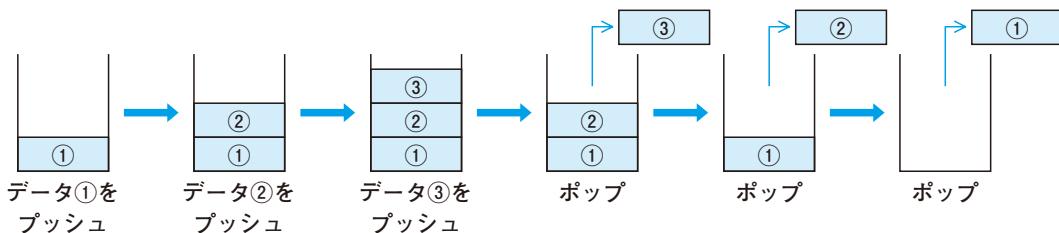
プログラムの解説

```
01 整数型 : x, y
02 Stack: stack ← Stack() ←———— スタックの作成
03 Queue: queue ← Queue() ←———— キューの作成

04 stack.push(10) ←———— スタックにデータをpush
05 stack.push(20) ←————
06 queue.enq(stack.pop()) ←———— キューにデータを挿入
07 queue.enq(30) ←————
08 stack.push(40) ←———— スタックからデータを取り出す
09 stack.push(queue.deq())
10 x ← stack.pop()
11 y ← queue.deq() ←———— キューからデータを取り出す
```

スタック

スタックとは、プッシュとポップという、2つの基本操作をもったデータ構造です。プッシュはスタックにデータを格納する操作、ポップはスタックからデータを取り出す操作です。このとき重要なのは、後に格納したデータから先に取り出されるという、後入れ先出し（LIFO：Last in First Out）の原則です。



キュー

スタックがデータを格納した順に積み上げていくのに対し、キューはデータを順に列のうしろに並べていくデータ構造です。データを列に入れる操作をエンキュー、データを列から取り出す操作をデキューといいます。キューでは、データは列の先頭から取り出されるので、先に格納したデータから先に取り出さ

れる先入れ先出し (FIFO : First in First Out) のデータ構造になります。

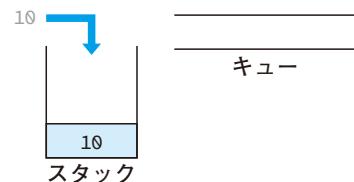


問題解説

プログラムを順にたどって、スタックとキューの内容がどのように変化するかをトレースしましょう。

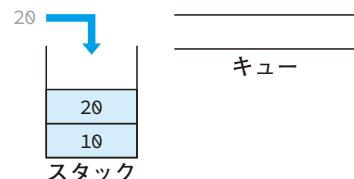
行番号 04：スタックに 10 を格納します。

`stack.push(10)`



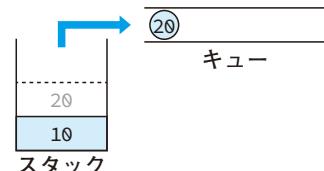
行番号 05：スタックに 20 を格納します。

`stack.push(20)`



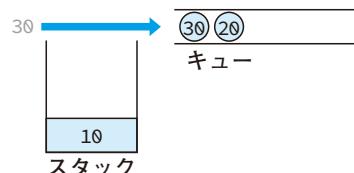
行番号 06：スタックからデータを取り出し、それをキューに格納します。

`queue.enq(stack.pop())`



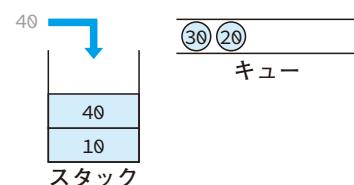
行番号 07：キューに 30 を格納します。

`queue.enq(30)`



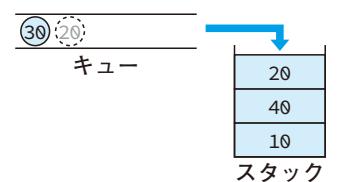
行番号 08：スタックに 40 を格納します。

`stack.push(40)`



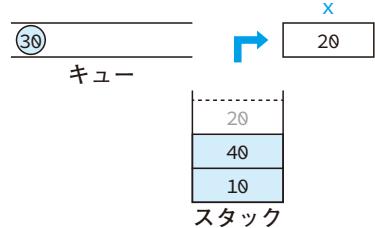
行番号 09：キューからデータを取り出し、それをスタックに格納します。

`stack.push(queue.deq())`



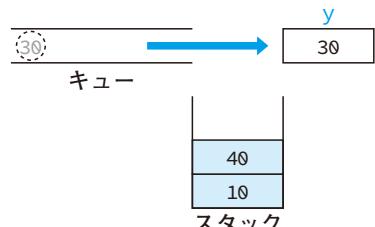
行番号 10：スタックからデータを取り出し、変数 x に代入します。

`x ← stack.pop()`



行番号 11：キューからデータを取り出し、変数 y に代入します。

`y ← queue.deq()`



以上から、変数 x には 20、変数 y には 30 がそれぞれ代入されます。正解は **ウ** です。

○ 解答 ○
問10 **ウ**

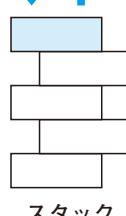


合格のカギ

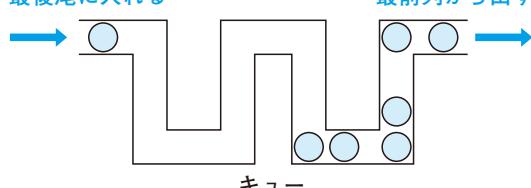
スタックとキュー

スタックは、データを入力した順に上に積み重ねていき、取り出すときは上から取り出していくイメージです。一方、キューは、ATM の順番待ちの列のように、データを到着した順番に列の最後尾に並べ、取り出すときは列の最前列から取り出します。

上に積み上げる 上から取り出す



最後尾に入れる





11

次のプログラム中の と に入れる正しい答えの組合せを、解答群の中から選べ。

関数 calcMean は、要素数が 1 以上の配列 dataArray を引数として受け取り、要素の値の平均を戻り値として返す。ここで、配列の要素番号は 1 から始まる。

[プログラム]

```
○ 実数型 : calcMean( 実数型の配列 : dataArray )
    実数型 : sum, mean
    整数型 : i
    sum ← 0
    for (i を 1 から dataArray の要素数 まで 1 ずつ増やす)
        sum ← 
    endfor
    mean ← sum ÷  /* 実数として計算する */
    return mean
```

解答群

	a	b
ア	sum + dataArray[i]	dataArray の要素数
イ	sum + dataArray[i]	(dataArray の要素数 + 1)
ウ	sum × dataArray[i]	dataArray の要素数
エ	sum × dataArray[i]	(dataArray の要素数 + 1)

※出典：IT パスポート試験プログラミング的思考力を問う擬似言語のサンプル問題

プログラムの解説

```
01 ○ 実数型 : calcMean( 実数型の配列 : dataArray )
02   実数型 : sum, mean
03   整数型 : i
04   sum ← 0
05   for (i を 1 から dataArray の要素数 まで 1 ずつ増やす )
06     sum ← [a]
07   endfor
08   mean ← sum ÷ [b] /* 実数として計算する */
09   return mean
```

平均値の計算

平均値とは、データの合計をデータの個数で割ったものです。プログラムは、行番号 04～行番号 07 でデータの合計を求め、その結果を変数 sum に代入します。

```
04   sum ← 0
05   for (i を 1 から dataArray の要素数 まで 1 ずつ増やす )
06     sum ← [a]
07   endfor
```

次に、データの合計をデータの個数で割って、変数 mean に代入します。

```
08   mean ← sum ÷ [b] /* 実数として計算する */
```

問題解説

空欄 a：配列 dataArray の先頭から末尾までの要素の合計は、

```
sum ← 0
sum ← sum + dataArray[1]
sum ← sum + dataArray[2]
:
}
} 繰り返し処理
```

のように、配列 dataArray の各要素を変数 sum に加えていきます。繰り返し処理の部分に for 構文を使うと、次のようにになります。

```
04   sum ← 0
05   for (i を 1 から dataArray の要素数 まで 1 ずつ増やす )
06     sum ← sum + dataArray[i]
07   endfor
      ↑ 空欄 a
```

以上から、空欄 a には「`sum + dataArray[i]`」が入ります。

空欄 b：平均値はデータの合計をデータの個数で割ったものです。データの合計は行番号 04～07 の処理で変数 sum に格納されています。また、データの個数とはこの場合配列 dataArray の要素数です。したがって、平均値を求める処理は次のようにになります。

08 mean \leftarrow sum \div dataArray の要素数

↑ 空欄 b

以上から、空欄 b には「dataArray の要素数」が入ります。

空欄 a が「sum + dataArray[i]」、空欄 b が「dataArray の要素数」なので、アが正解です。

解答

問11 ア



合格の力ギ

変数名はヒント

プログラムの変数名は、その変数の役割や値の意味がわかるように付けるのが一般的です。本問の場合も、sum（合計、和という意味）、mean（平均という意味）という変数名から、その変数の役割が推察できるようになっています。「sum という変数名だから、合計を求めるのだな」というように、変数名は問題を解くヒントになります。

よく使われる変数名

変数名	説明
cnt, count	カウンタ。繰返し処理で個数を数えるときに使う。
curr, next, prev	配列やリストで、現在の要素 (curr) や前の要素 (prev), 次の要素 (next) への参照を格納する。
i, j, …	繰返しごとに変化する変数。
tmp, temp	値を一時的に保存しておく変数。
len	配列の要素数、文字数など。
str, s	文字列データ。
ch, c	文字データ。
ret	関数の戻り値を格納する変数。
buf	文字列や配列のデータなどを一時的にたくわえておく変数。



12

次のプログラム中の a と b に入れる正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まる。

手続 change は、引数に指定された金額を受け取り、その金額に必要な 10,000 円札、5,000 円札、1,000 円札、500 円硬貨、100 円硬貨それぞれの最小枚数を出力する手続である。100 円未満の金額は入力されないものとする。

[プログラム]

```

○ change( 整数型 : money )
整数型の配列 : yen ← {10000,5000,1000,500,100}
整数 : tmp ← money
for (i を 1 から yen の要素数 まで 1 ずつ増やす)
    num ←  a
    yen[i] "円：" num "枚" を出力する
    tmp ←  b
endfor

```

解答群

	a	b
ア	tmp ÷ yen[i] の商	tmp + yen[i] × num
イ	tmp ÷ yen[i] の商	tmp - yen[i] × num
ウ	tmp mod yen[i]	tmp ÷ yen[i] の商
エ	tmp mod yen[i]	tmp + yen[i] × num
オ	tmp - yen[i] × num	tmp ÷ yen[i] の商
カ	tmp - yen[i] × num	tmp mod yen[i]

※オリジナル問題

プログラムの解説

```

01 ○ change( 整数型 : money )
02 整数型の配列 : yen ← {10000,5000,1000,500,100} ← 金種
03 整数 : tmp ← money
04 for (i を 1 から yen の要素数 まで 1 ずつ増やす)
05     num ←  a
06     yen[i] "円：" num "枚" を出力する ← 各金種の枚数を出力
07     tmp ←  b
08 endfor

```

金種計算

たとえば、168,300円の現金を用意するには、10,000円札が16枚、5,000円札が1枚、1,000円札が3枚、500円硬貨0枚、100円硬貨3枚が必要になります。このように、金額から必要な金種の枚数を求めることがあります。

168,300円の金種計算は、次のようにになります。

金額	金種	商	余り	枚数
① $168300 \div 10000 = 16$ 余り 8300				→ 16枚
② $8300 \div 5000 = 1$ 余り 3300				→ 1枚
③ $3300 \div 1000 = 3$ 余り 300				→ 3枚
④ $300 \div 500 = 0$ 余り 300				→ 0枚
⑤ $300 \div 100 = 3$ 余り 0				→ 3枚
	↑ tmp	↑ yen[i]		↑ num

プログラムは、金額を変数 tmp、金種を配列 yen の各要素に格納し、求めた枚数を変数 num に格納します。

問題解説

空欄 a：配列 yen には、10,000円、5,000円、1,000円、…のように、金種が順に格納されています。また、金額は tmp に代入されているので、金種ごとの枚数は

`num ← tmp ÷ yen[i] の商`

で求めることができます。以上から、**空欄 a** は「`tmp ÷ yen[i] の商`」となります。

空欄 b：1つの金種の枚数を求めたら、次の金種の枚数は、前の金種の枚数分の金額を差し引いた残りの金額から求めます。

`tmp ← tmp - yen[i] × num`

あるいは、金額 tmp を金種 yen[i] で割った余りでも同じことです。`mod` 演算子を使うと、

`tmp ← tmp mod yen[i]`

となります。以上から、**空欄 b** には「`tmp - yen[i] × num`」または「`tmp mod yen[i]`」のどちらかが入ります。

解答群には、**イ** が

a	b
<code>tmp ÷ yen[i] の商</code>	<code>tmp - yen[i] × num</code>

となっているので、**イ** が正解となります。

○ 解答 ○

問12 イ



合格の力ギ

プログラムは1通りではない

本問の空欄 b のように、同じ処理を行うプログラムは、何通りもあるのが一般的です。正解に当たる処理が解答群にない場合は、別の書き方がないかどうか考えてみましょう。



13

次のプログラム中の と に入る正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素は 1 から始まる。

関数 multiMatrix は、引数に指定した 2 つの行列 $m1$, $m2$ の積を返す関数である。引数に指定する行列及び 2 つの行列の積は、いずれも整数型の二次元配列で表現する。

行列 $m1$ が a 行 n 列、行列 $m2$ が n 行 b 列の行列であるとき、2 つの行列の積 $m1 \times m2$ は、一般に a 行 b 列の行列となる。 $m1$ の列数と $m2$ の行数が等しくなければ積を求めることはできない。 $m1 \times m2$ の積 x の i 行 j 列目の要素 $x[i, j]$ は、一般に次の式で計算することができる。

$$x[i, j] = \sum_{k=1}^n (m1[i, k] \times m2[k, j])$$

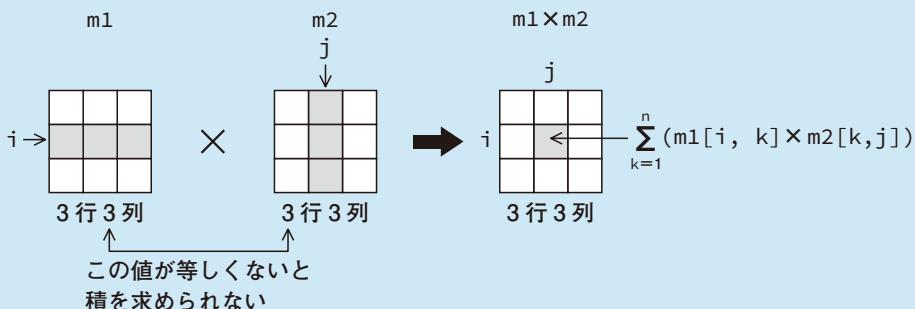


図 行列の積の説明

[プログラム]

○ 整数型の二次元配列 : multiMatrix(整数型の二次元配列 : $m1$,
整数型の二次元配列 : $m2$)

整数型 : i , j , k

整数型の二次元配列 : $x[m1$ の行数 , $m2$ の列数]

`if (a)`

```
for(i を 1 から  $m1$  の行数 まで 1 ずつ増やす)
    for(j を 1 から  $m2$  の列数 まで 1 ずつ増やす)
         $x[i, j] \leftarrow 0$ 
        for(k を 1 から  b まで 1 ずつ増やす)
             $x[i, j] \leftarrow x[i, j] + m1[i, k] \times m2[k, j]$ 
        endfor
    endfor
endfor
return x
endif
```

解答群

	a	b
ア	m1 の行数 が m2 の列数 に等しい	m1 の列数
イ	m1 の行数 が m2 の列数 に等しい	m2 の行数
ウ	m1 の列数 が m2 の行数 に等しい	m1 の列数
エ	m1 の列数 が m2 の行数 に等しい	m2 の列数
オ	m1 の列数 が m2 の行数 に等しくない	m1 の列数
カ	m1 の列数 が m2 の行数 に等しくない	m2 の行数
キ	m1 の列数 が m2 の行数 に等しくない	m1 の行数
ク	m1 の列数 が m2 の行数 に等しくない	m2 の列数

※オリジナル問題

プログラムの解説

```

01 ○ 整数型の二次元配列 : multiMatrix( 整数型の二次元配列 : m1,
02                               整数型の二次元配列 : m2)
03 整数型 : i, j, k
04 整数型の二次元配列 : x[ m1 の行数 , m2 の列数 ]           ← m1 × m2 の結果を格納
05 if ( [ a ] )
06     for(i を 1 から m1 の行数 まで 1 ずつ増やす)
07         for(j を 1 から m2 の列数 まで 1 ずつ増やす)
08             x[i,j] ← 0
09             for(k を 1 から [ b ] まで 1 ずつ増やす)
10                 x[i,j] ← x[i,j] + m1[i,k] × m2[k,j] ← m1 × m2 の i 行 j 列の
11                     要素を計算
12             endfor
13         endfor
14     return x
15 endif

```

行列の積

数学の行列の積の計算方法については問題文に説明がありますが、簡単な例を使って確認しておきましょう。たとえば、行列 m1、行列 m2 が、それぞれ次の内容とします。

$$m1 = \begin{array}{|c|c|c|} \hline 1 & 0 & 2 \\ \hline 3 & 1 & 4 \\ \hline 0 & 2 & 1 \\ \hline \end{array}, \quad m2 = \begin{array}{|c|c|c|} \hline 2 & 1 & 0 \\ \hline 1 & 3 & 2 \\ \hline 0 & 1 & 2 \\ \hline \end{array}$$

行列 m1 が 3 行 3 列、行列 m2 が 3 行 3 列なので、2 つの行列の積 m1 × m2 も、3 行 3 列の行列となります。また、m1 × m2 の 1 行 1 列目の要素 x[1,1] は、次のように計算します。

$$x[i,j] = 1 \times 2 + 0 \times 1 + 2 \times 0 = 2$$

$\begin{matrix} & \text{m1} \\ i \rightarrow & \begin{array}{|c|c|c|} \hline 1 & 0 & 2 \\ \hline 3 & 1 & 4 \\ \hline 0 & 2 & 1 \\ \hline \end{array} \end{matrix} \times \begin{matrix} & \text{m2} \\ j \downarrow & \begin{array}{|c|c|c|} \hline 2 & 1 & 0 \\ \hline 1 & 3 & 2 \\ \hline 0 & 1 & 2 \\ \hline \end{array} \end{matrix}$

$m1 \times m2$ の要素の個数は、「 $m1$ の行数 \times $m2$ の列数」となります。プログラムは `for` 構文を 2 重にして、要素の個数分の計算を実行します。また各要素の計算は、行番号 09 ~ 11 の繰返し処理で行っています。そのためプログラムは `for ~ endfor` を 3 重にした 3 重ループになります。

問題解説

空欄 a : $m1$ の列数と $m2$ の行数が等しくないと、行列の積は計算できません。**空欄 a** には、行列の積の計算を行うための条件が入るので、

`if (m1 の列数 が m2 の行数 に等しい)`
となります。

↑ 空欄 a

空欄 b : $m1 \times m2$ の i 行 j 列目の要素 $x[i, j]$ は、 $m1$ の i 行目の各要素と、 $m2$ の j 列目の各要素を順に掛け算して足し合わせたものです。プログラムでは、行番号 09 ~ 11 の繰返し処理でこの計算を行っています。

$x[i, j] \leftarrow x[i, j] + m1[i, k] \times m2[k, j]$

変数 k は、繰返し処理で 1 から $m1$ の 1 行の要素数 (= $m1$ の列数) まで 1 ずつ増やします。したがって**空欄 b** は

`for (k を 1 から m1 の列数 まで 1 ずつ増やす)`
となります。

↑ 空欄 b

以上から、**空欄 a** が「 $m1$ の列数 が $m2$ の行数 に等しい」、**空欄 b** が「 $m1$ の列数」の組合せの **□** が正解です。

なお、 $m1$ の列数と $m2$ の行数は等しいので、**空欄 b** は

`for (k を 1 から m2 の行数 まで 1 ずつ増やす)`

でも正解となりますが、**空欄 a** の正しい組合せが解答欄にありません。

○ 解答
問13 □



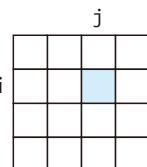
合格のカギ

二次元配列と配列の配列

二次元配列は、一般的なプログラム言語では「配列の配列」として表しますが、情報処理試験の擬似言語では二次元配列と配列の配列を区別しています。二次元配列の要素は `matrix[i, j]` のように表しますが、配列の配列は `matrix[i][j]` となることに注意してください。また、擬似言語の配列の配列では、要素数が異なる配列の配列もつくれます。

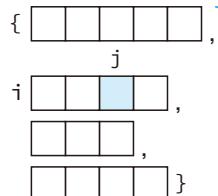
● 二次元配列

`matrix[i, j]`



● 配列の配列

`matrix[i][j]`



要素数が
異なって
いてもよい



14

次のプログラム中のに入れる正しい答えを、解答群の中から選べ。

任意の異なる 2 文字を c1, c2 とするとき、英単語群に含まれる英単語において、c1 の次に c2 が出現する割合を求めるプログラムである。英単語は、英小文字だけから成る。英単語の末尾の文字が c1 である場合、その箇所は割合の計算に含めない。例えば、図に示す 4 語の英単語 “importance”, “inflation”, “information”, “innovation” から成る英単語群において、c1 を “n”, c2 を “f” とする。英単語の末尾の文字以外に “n” は五つあり、そのうち次の文字が “f” であるものは二つである。したがって、求める割合は、 $2 \div 5 = 0.4$ である。c1 と c2 の並びが一度も出現しない場合、c1 の出現回数によらず割合を 0 と定義する。

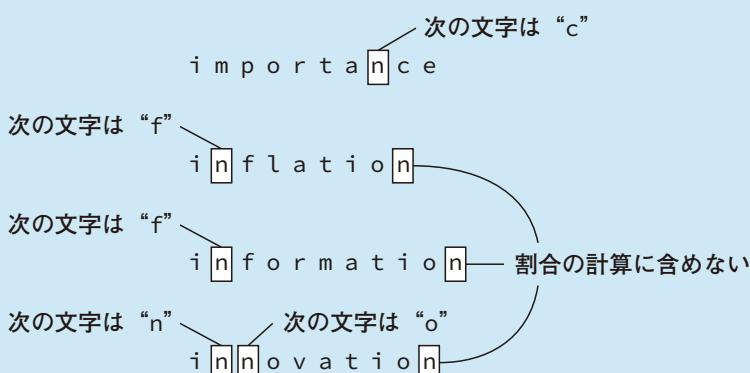


図 4 語から成る英単語群の例

プログラムにおいて、英単語群は Words 型の大域変数 words に格納されている。クラス Words のメソッドの説明を、表に示す。本問において、文字列に対する演算子 “+” は、文字列の連結を表す。また、整数に対する演算子 “÷” は、実数として計算する。

表 クラス Words のメソッドの説明

メソッド	戻り値	説明
freq(文字列型 : str)	整数型	英単語群中の文字列 str の出現回数を返す。
freqE(文字列型 : str)	整数型	英単語群の中で、文字列 str で終わる英単語の数を返す。

[プログラム]

```
大域 : Words: words /* 英単語群が格納されている */
```

```
/* c1 の次に c2 が出現する割合を返す */
```

```
○実数型 : prob( 文字型 : c1, 文字型 : c2 )
```

```
文字列型 : s1 ← c1 の 1 文字だけから成る文字列
```

```
文字列型 : s2 ← c2 の 1 文字だけから成る文字列
```

```

if (words.freq(s1 + s2) が 0 より大きい)
    return [ ]
else
    return 0
endif

```

解答群

- ア** $(\text{words.freq}(s1) - \text{words.freqE}(s1)) \div \text{words.freq}(s1 + s2)$
- イ** $(\text{words.freq}(s2) - \text{words.freqE}(s2)) \div \text{words.freq}(s1 + s2)$
- ウ** $\text{words.freq}(s1 + s2) \div (\text{words.freq}(s1) - \text{words.freqE}(s1))$
- エ** $\text{words.freq}(s1 + s2) \div (\text{words.freq}(s2) - \text{words.freqE}(s2))$

※サンプル問題より

プログラムの解説

```

01 大域 : Words: words /* 英単語群が格納されている */
02 /* c1 の次に c2 が出現する割合を返す */
03 ○ 実数型 : prob( 文字型 : c1, 文字型 : c2 )
04 文字列型 : s1 ← c1 の 1 文字だけから成る文字列
05 文字列型 : s2 ← c2 の 1 文字だけから成る文字列
06 if (words.freq(s1 + s2) が 0 より大きい)
07     return [ ]
08 else
09     return 0
10 endif

```

大域変数 `words` に格納されている英単語群を調べて、「`c1` の次に `c2` が出現する割合」を求めるプログラムです。プログラムは関数 `prob` を定義しているだけなので、プログラムを実行するときは、2 文字を引数に指定して、

```
rate ← prob('n', 'f')
```

のように関数 `prob` を呼び出す必要があります。上の例では、「`n` の次に `f` が出現する割合」が、実数型の変数 `rate` に代入されます。

問題解説

空欄には、関数 `prob` の戻り値である「`c1` の次に `c2` が出現する割合」を求める式が入ります。「`c1` の次に `c2` が出現する割合」は、大まかにいえば、

(c1 の次に c2 が出現する回数) ÷ (c1 の出現回数)

で求めることができます。ただし、問題文の説明に「英単語の末尾の文字が c1 である場合、その箇所は割合の計算に含めない」とあるので、c1 が末尾に出現する回数は、c1 の出現回数から差し引かなければなりません。したがって、求める式は次のようにになります。

(c1 の次に c2 が出現する回数) ÷ (c1 の出現回数 - c1 が末尾に出現する回数)

①

②

③

この式を、擬似言語で表します。

① c1 の次に c2 が出現する回数

引数に指定された文字 c1 と c2 は、行番号 04, 05 でそれぞれ文字列 s1 と s2 に格納されます。「c1 の次に c2 が出現する回数」は、クラス Words のメソッド freq に、s1 と s2 を連結した文字列を指定し、次のようにになります。

`words.freq(s1 + s2)`

② c1 の出現回数

同様に、「c1 の出現回数」は次のようにになります。

`words.freq(s1)`

③ c1 が末尾に出現する回数

「c1 が末尾に出現する回数」は、Words のメソッド freqE に s1 を指定し、

`words.freqE(s1)`

とします。

これらを組み立てると、空欄に入る式は次のようにになります。

`words.freq(s1 + s2) ÷ (words.freq(s1) - words.freqE(s1))`

以上から、正解は **ウ** です。

○ 解答 ○

問14 **ウ**



合格のカギ

計算式の組立て

計算式を組み立てる問題では、まず計算に用いる各項と演算子を決め、それから各項に変数や関数、メソッドを当てはめていきます。割合の計算では、分母と分子を逆にしないよう注意しましょう。



15

次の記述中の [] に入る正しい答えを、解答群の中から選べ。

配列を用いてヒープを実現するプログラムである。ヒープとは二分木であり、本問では、親は一つ又は二つの子をもち、親の値は子の値よりも常に大きいか等しいという性質をもつものとする。ヒープの例を図1に示す。図1において、丸は節を、丸の中の数値は各節が保持する値を表す。子をもつ節を、その子に対する親と呼ぶ。親をもたない節を根と呼び、根は最大の値をもつ。

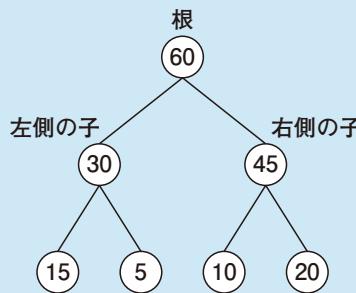


図1 ヒープの例

プログラムは、整数型の配列 `data` に格納されている n 個 ($n > 0$) のデータを、次の①～③の規則で整数型の配列 `heap` に格納する。配列の要素番号は1から始まるものとする。

- ①配列要素 `heap[i]` ($i = 1, 2, 3, \dots$) は、節に対応する。配列要素 `heap[i]` には、節が保持する値を格納する。
- ②配列要素 `heap[1]` は、根に対応する。
- ③配列要素 `heap[i]` ($i = 1, 2, 3, \dots$) に対応する節の左側の子は配列要素 `heap[i × 2]` に対応し、右側の子は配列要素 `heap[i × 2 + 1]` に対応する。子が一つの場合、左側の子として扱う。

図1のヒープの例に対応した配列 `heap` の内容を、図2に示す。

要素番号	1	2	3	4	5	6	7
heap	60	30	45	15	5	10	20

注記：矢印●→は、始点、終点の二つの配列要素に
対応する節が、親子関係にあることを表す。

図2 図1に対応する配列 `heap` の内容

プログラムの実行後、配列 `heap` の内容は [] となる。

〔プログラム〕

```
整数型の配列 : data ← {5, 10, 15, 20, 30}
整数型の配列 : heap ← {}
整数型 : i, parent, curr

for (i を 1 から data の要素数 まで 1 ずつ増やす)
    heap の末尾に data[i] の値 を追加する
    curr ← i
    while (curr は 1 より大きい)
        parent ← curr ÷ 2 の商
        if (heap[curr] は heap[parent] より大きい)
            heap[parent] と heap[curr] の値を交換する
            curr ← parent
        else
            break
        endif
    endwhile
endfor
```

解答群

- ア {5, 10, 15, 20, 30}
- イ {5, 10, 20, 15, 30}
- ウ {20, 30, 10, 5, 15}
- エ {20, 30, 10, 15, 5}
- オ {30, 20, 10, 5, 15}
- カ {30, 20, 10, 15, 5}
- キ {30, 15, 20, 5, 15}

※出典：基本情報技術者試験平成 30 年春期午後問題問 8 を改変

プログラムの解説

```
01 整数型の配列 : data ← {5, 10, 15, 20, 30}
02 整数型の配列 : heap ← {}
03 整数型 : i, parent, curr

04 for (i を 1 から data の要素数 まで 1 ずつ増やす)
05     heap の末尾に data[i] の値 を追加する ← データをヒープに追加
06     curr ← i
07     while (curr は 1 より大きい)
08         parent ← curr ÷ 2 の商 ← curr の親の要素番号を求める
09         if (heap[curr] は heap[parent] より大きい)
```

```

10      heap[parent] と heap[curr] の値を交換する ← 子の値のほうが大きい場合
11      curr ← parent
12  else
13      break
14  endif
15  endwhile
16 endfor

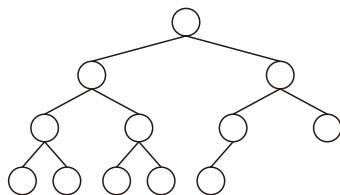
```

ヒープの特徴

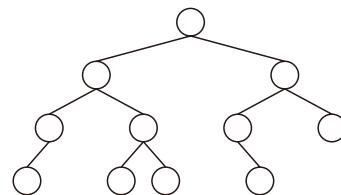
ヒープは、複数のデータを格納するデータ構造の一種で、次のような特徴をもっています。

①ヒープは二分木のデータ構造です。ただし、ヒープの二分木は、子を左側から順に詰めていく特徴があります。

○ ヒープの二分木

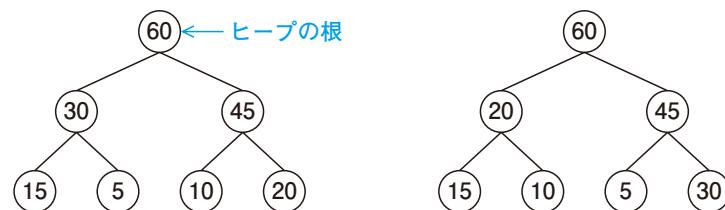


✗ ヒープの二分木ではない



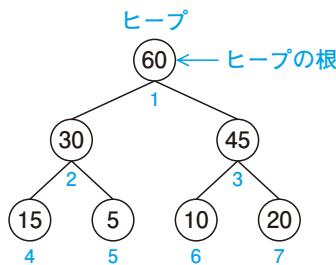
②親の値は、必ず子の値より大きいか等しくなければなりません。これを**ヒープの性質**（ヒープ条件）といいます。なお、子の値は左右どちらが大きくてもかまいません。

ヒープの例



上の例のように、同じデータを格納する場合でも、ヒープの格納順序は何通りもあります。ただし、いちばん大きい値は必ずヒープの根に格納されます。

③ヒープは、一般に配列を使って実現します。ヒープの各節に順番に番号を付けると、ヒープの各節と配列には、次のような対応関係があります。



配列による実装

1	60
2	30
3	45
4	15
5	5
6	10
7	20

親の要素番号 × 2
親の要素番号 × 2 + 1

問題解説

ヒープにデータを追加する手順は次のとおりです。

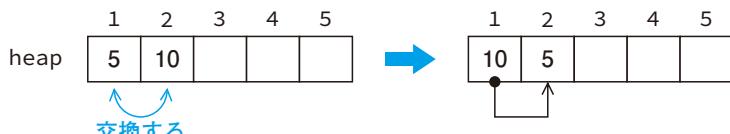
- ①配列 heap の末尾にデータを 1 個追加し, `heap[curr]` とします (行番号 05, 06)。
- ②`heap[curr]` の親の節に当たる要素番号 `parent` を求めます (行番号 08)。
- ③`heap[curr]` のほうが親のデータより大きい場合は、値を交換します (行番号 09, 10)。そうでなければ追加は終了です。
- ④親の要素番号を `curr` とし、②, ③を繰り返します (行番号 11)。`curr` が根になったら追加終了です。

上記の手順で、データを 5, 10, 15, 20, 30 の順に追加すると、配列 `heap` は次のようにになります。

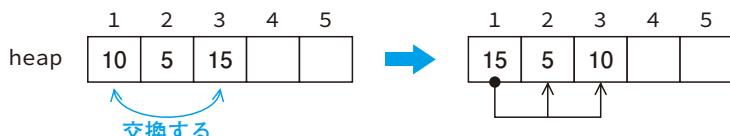
- ・データ 5 を追加

heap	1	2	3	4	5
5					

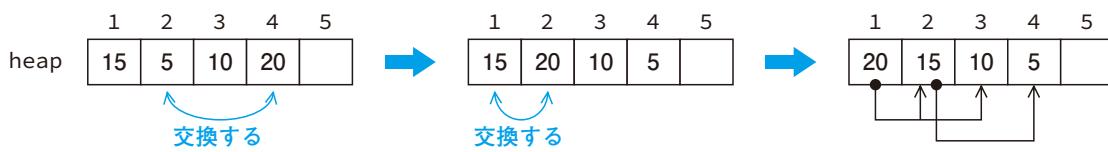
- ・データ 10 を追加



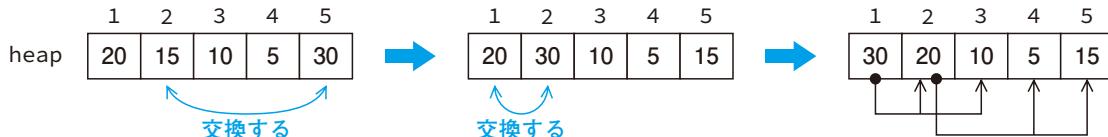
- ・データ 15 を追加



- ・データ 20 を追加



- ・データ 30 を追加



以上から、プログラムを終了すると、配列 `heap` の内容は {30, 20, 10, 5, 15} となります。正解は **オ** です。

○ 解答 ○

問15 **オ**



合格のカギ

ヒープソート

ヒープは、根にもつとも大きな値が格納されます。この性質を利用し、データをヒープに格納してから、根の値を順に取り出していく（根を取り出したら、その都度ヒープを再構成する）と、データが大きい順に整列されます。このアルゴリズムをヒープソートといいます。



問 16 次の記述中の a , b に入れる正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まる。

“listen”と“silent”的ように、1つの単語の文字を並べ替えてできる別の単語をアナグラムという。関数 chkAnagram は、引数に指定した2つの文字列が互いにアナグラムかどうかを調べ、結果を true または false で返す。

クラス Map は、文字列型のキーと整数型の値のペアを要素として格納するデータ構造である。クラス Map のメソッドを図に示す。Map 型の変数は、クラス Map のインスタンスの参照を格納するものとする。

メソッド	戻り値の型	説明
Map()	Map	コンストラクタ。中身が空のマップを作成し、その参照を返す。
get(文字列型 : key)	整数型	引数に指定した key をキーとする要素をマップから検索し、対応する値を返す。キーが存在しない場合は未定義の値を返す。
put(文字列型 : key, 整数型 : value)	なし	引数に指定した key と value をペアとする要素をマップに格納する。key をキーとする要素が既に存在する場合は、その要素の値を value で更新する。
equals(Map: map)	論理型	引数に指定したマップの構成要素が、自分自身と一致する場合は true、一致しない場合は false を返す。構成要素の順序は区別しない。

図 クラス Map の説明

[プログラム]

```
○論理型 : chkAnagram( 文字列型 : str1, 文字列型 : str2)
  Map: map1 ← Map(), map2 ← Map()
  整数型 : i
  if (str1 の文字数 と str2 の文字数 が等しくない)
    return false /* 文字数が異なる場合 */
  endif
  for (i を 1 から str1 の文字数 まで 1ずつ増やす)
    if (map1.get(str1[i]) が 未定義)
      map1.put(str1[i], 1)
    else
      map1.put(str1[i],  a )
    endif
```

```

if (map2.get(str2[i]) が 未定義 )
    map2.put(str2[i], 1)
else
    map2.put(str2[i], [ ] b [ ] )
endif
endfor
return map1.equals(map2)

```

解答群

	a	b
ア	map1.get(str1[i])	map2.get(str2[i])
イ	map1.get(str2[i])	map2.get(str1[i])
ウ	map1.get(str1[i] + 1)	map2.get(str2[i] + 1)
エ	map1.get(str2[i] + 1)	map2.get(str1[i] + 1)
オ	map2.get(str1[i])	map1.get(str2[i])
カ	map2.get(str2[i])	map1.get(str1[i])
キ	map2.get(str1[i] + 1)	map1.get(str2[i] + 1)
ク	map2.get(str2[i] + 1)	map1.get(str1[i] + 1)

※オリジナル問題

プログラムの解説

```

01 ○論理型 : chkAnagram( 文字列型 : str1, 文字列型 : str2)
02     Map: map1 ← Map(), map2 ← Map()
03     整数型 : i
04     if (str1 の文字数 と str2 の文字数 が等しくない)
05         return false /* 文字数が異なる場合 */
06     endif
07     for (i を 1 から str1 の文字数 まで 1ずつ増やす)
08         if (map1.get(str1[i]) が 未定義 )
09             map1.put(str1[i], 1)
10         else
11             map1.put(str1[i], [ ] a [ ] )
12         endif
13         if (map2.get(str2[i]) が 未定義 )
14             map2.put(str2[i], 1)
15         else
16             map2.put(str2[i], [ ] b [ ] )
17         endif
18     endfor
19     return map1.equals(map2)

```

マップクラス

クラス Map で表現するマップは、ハッシュ、連想配列、辞書（ディクショナリ）などとも呼ばれるデータ構造で、次のような特徴があります。

- 要素には、キーと値のペアを格納する。
- 各要素のキーは重複しない。
- 構成要素に順序はない。

要素の追加、または既にある要素の値の更新には、クラス Map のメソッド put を用います。また、指定したキーに対応する値を得るには、メソッド get を用います。

```
例 : map.put("a", 10) ← マップに要素 ("a", 10) を追加  
      map.put("a", 20) ← キーが "a" の要素の値を 20 に更新  
      n ← map.get("a") ← キーが "a" の要素の値を変数 n に代入
```

格納した値に要素番号を付けて管理するのが通常の配列だとすると、マップは格納した値にキーを付けて管理するデータ構造と考えることができます。

アナグラム

2つの文字列が互いにアナグラムかどうかを調べる方法は、何通りか考えられます。本問のプログラムは、2つの文字列について、文字ごとの出現回数を数え、文字と出現回数のペアをマップに記録します。たとえば “listen” であれば、

```
{"l", 1}, {"i", 1}, {"s", 1}, {"t", 1}, {"e", 1}, {"n", 1}
```

となります。また、 “silent” であれば

```
{"s", 1}, {"i", 1}, {"l", 1}, {"e", 1}, {"n", 1}, {"t", 1}
```

となります。プログラムは、行番号 07～12 で str1、行番号 13～17 で str2 のマップをつくっています。

2つのマップを比較し、両者の構成要素が一致していれば、アナグラムであると判定します。プログラムは行番号 19 で、メソッド equals を使って 2つのマップが等しいかどうかを調べています。

問題解説

str1 の各文字の出現回数を集計するために、str1 の各文字をキー、出現回数を値とする要素をマップ map1 に登録します。例として、str1 が “bbc” の場合で考えてみましょう。

まず先頭文字 “b” をキーとする要素が、既に map1 に登録されているかどうかを調べます（行番号 08）。map1 にはまだ何も登録されていませんから、map1.get(str1[i]) は未定義の値を返します。その場合、行番号 09 で “b” をキー、出現回数 1 を値とする要素を map1 に追加します。

```
08 if (map1.get(str1[i]) が 未定義)  
09   map1.put(str1[i], 1)
```

次に、2 文字目 “b” をキーとする要素が既に map1 に登録されているかどうかを調べます。先頭文字も “b” だったので、要素は既に登録されています。この場合、“b” をキーとする要素の値を 1 加算する必要があります。

既に登録されている要素の値は、メソッド get を使って map1.get(str1[i]) で取得できます。この

値に1を加算した値を、メソッドputを使って登録します。したがって、行番号11は次のような処理になります。

```
11 map1.put(str1[i], map1.get(str1[i]) + 1)
```

↑ 空欄a

str2についても、同様の手順でマップmap2を作成するので、行番号16の処理は次のようにになります。

```
16 map2.put(str2[i], map2.get(str2[i]) + 1)
```

↑ 空欄b

以上から、正解はウです。

解答

問16 ウ



合格のカギ アナグラム判定

アナグラムを判定する方法は、マップを利用するほかにもいくつかあります。

①2つの英単語の各文字をアルファベット順に並べ替え、両者を比較する方法。たとえば、“listen”と“silent”は、アルファベット順に並べ替えるとどちらも“eilnst”となり、アナグラムであることがわかります。

②単語を1文字ずつ調べ、一致した文字を単語から削除していく方法。最終的に文字数が0になれば、アナグラムであることがわかります。たとえば、単語“silent”から、l, i, s, t, e, nの順に文字を削除していくと、“silent”→“sient”→“sent”→“ent”→“en”→“n”→“”のように、すべての文字が消えるので、“listen”と“silent”はアナグラムです。



17

次の記述中の [a] ~ [c] に入る適切な答えの組合せを、解答群の中から選べ。

A 社は、関東の N 事業所で利用している営業支援システムを、関西の M 事業所でも利用するために、IPsec を利用した VPN の導入を検討している。

VPN の実現には、VPN ルータを利用する。IPsec では、データ受信側の VPN ルータがデータ送信側の VPN ルータを認証する仕組みの一つとして、RSA アルゴリズムを用いたデジタル署名を利用し、なりすまし及び改ざんの検知を行っている。その仕組みを図に示す。

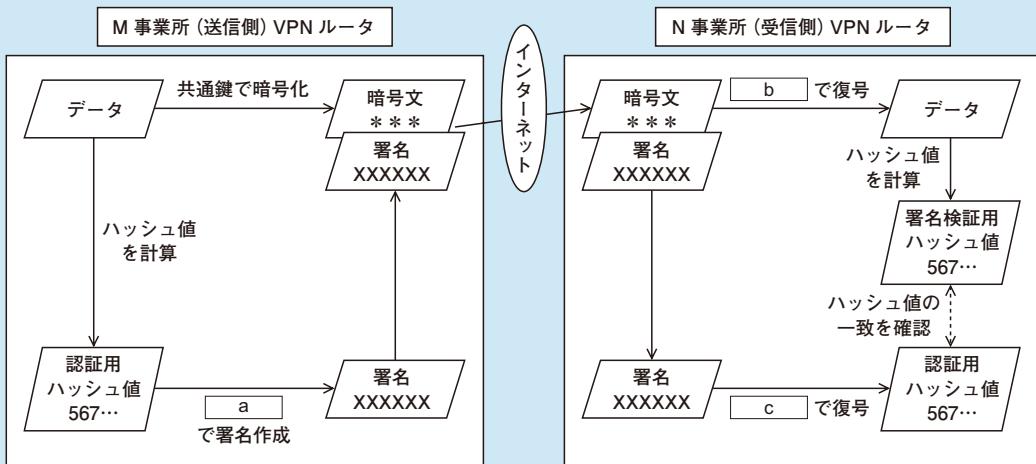


図 相手の VPN ルータを認証する仕組み

解答群

	a	b	c
ア	送信側の秘密鍵	共通鍵	送信側の公開鍵
イ	送信側の公開鍵	共通鍵	送信側の秘密鍵
ウ	受信側の秘密鍵	共通鍵	受信側の公開鍵
エ	受信側の公開鍵	共通鍵	受信側の秘密鍵
オ	共通鍵	受信側の秘密鍵	送信側の公開鍵
カ	共通鍵	送信側の公開鍵	受信側の秘密鍵
キ	共通鍵	受信側の秘密鍵	受信側の公開鍵
ク	共通鍵	受信側の公開鍵	受信側の秘密鍵

※出典：基本情報技術者試験平成 25 年度秋期午後問題問 4 を改変

問題解説

空欄 a, c : デジタル署名とは、送信データからハッシュ値を計算し、それを送信側の秘密鍵で暗号化したもののです。

これを復号するには、暗号化に使ったものとペアの鍵、すなわち送信側の公開鍵が必要です。

送信側の秘密鍵＝暗号化鍵

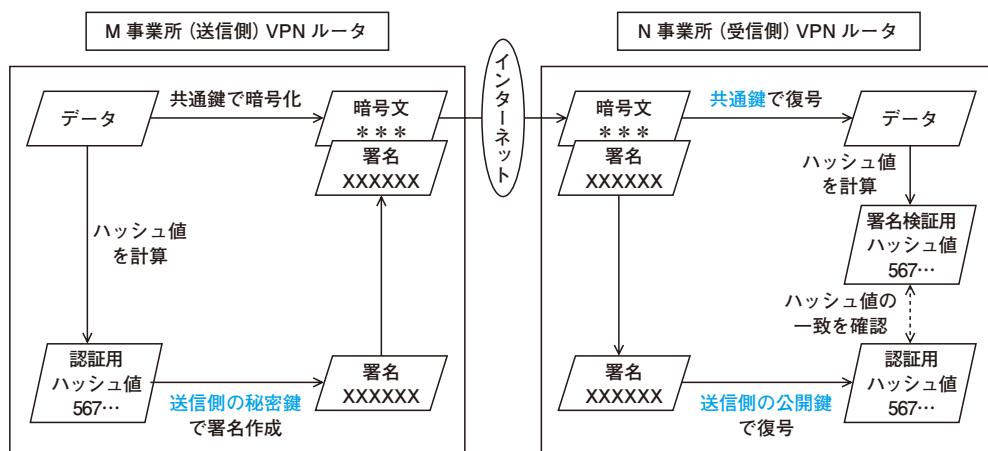
送信側の公開鍵＝復号鍵

秘密鍵というのは、所有者本人だけが保管しているものなので、それとペアの公開鍵でデジタル署名が復号できれば、デジタル署名が所有者本人で作成されたことを確認できます。

以上から、空欄**a**には「送信側の秘密鍵」、空欄**c**には「送信側の公開鍵」が入ります。

空欄b****：送信側のM事業所は、送信データを共通鍵で暗号化しています。共通鍵は、暗号化と復号に同一の鍵を使う暗号方式です。したがって、暗号文の復号も共通鍵で行います。**空欄**b****には「共通鍵」が入ります。

以上から、空欄**a**「送信側の秘密鍵」、空欄**b**「共通鍵」、空欄**c**「送信側の公開鍵」の**ア**が正解です。



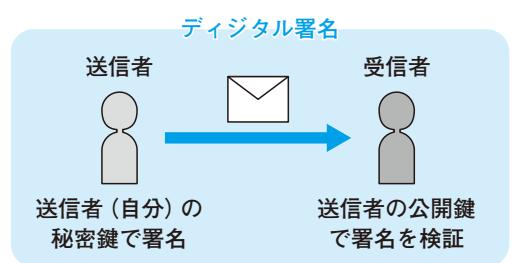
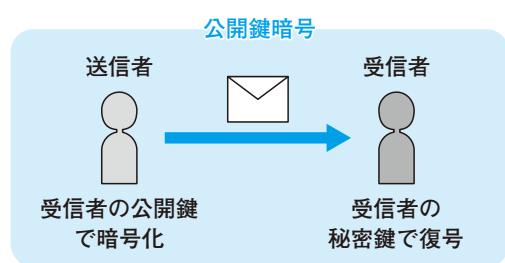
○ 解答 ○

問17 ア

合格のカギ

公開鍵と秘密鍵

公開鍵暗号やデジタル署名では、通信をやり取りする人同士が公開鍵と秘密鍵のペアを持っていることを前提とします。暗号通信は、メッセージを受信者の公開鍵で暗号化し、受信者の秘密鍵で復号します。デジタル署名では、送信者の秘密鍵で署名を行い、送信者の公開鍵で署名を確認します。まぎらわしいので注意してください。





18

次の記述中の [] に入る適切な答えの組合せを、解答群の中から選べ。

A 社が提供する Web サービスを利用するには、利用者が決めた利用者 ID とパスワードを、Web アプリケーションが動作するサーバに登録しておく必要がある。A 社の Web アプリケーションでは、利用者が Web アプリケーションにログインするときに、Web ブラウザから利用者 ID とパスワードがサーバに送信される。サーバは、受信した利用者 ID とパスワードを、照合することによって認証する。

利用者が決めたパスワードを、パスワードファイルに平文で保存しておくと、パスワードファイルの漏えいによって不正ログインが発生するおそれがある。そのための対策として、ハッシュ関数を用いパスワードのハッシュ値を計算（以下、ハッシュ化という）し、そのハッシュ値を保存する方式がある。この方式におけるログイン時の認証では、受信したパスワードから求めたハッシュ値を、パスワードファイルに保存されているハッシュ値と照合する。パスワードの保存の流れと、照合の流れを図に示す。

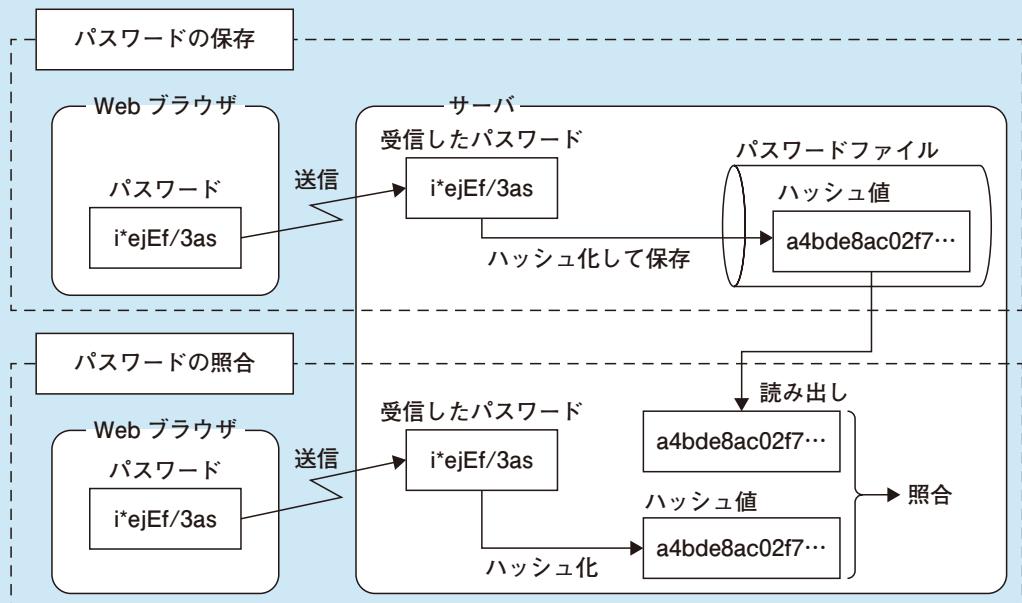


図 パスワードの保存と照合の流れ

ハッシュ化に用いるハッシュ関数は一般的に次のような特徴を備えているので、パスワードが一致していることの確認に用いることができる。

〔ハッシュ化に用いるハッシュ関数の特徴〕

- (1) パスワードの長さに関係なく、ハッシュ値は固定長になる。
- (2) []
- (3) ハッシュ値からパスワードを推測することが非常に困難である。
- (4) パスワードが1文字でも異なれば、ハッシュ値は大きく異なる。

解答群

- ア 異なるパスワードをハッシュ化すると、同じハッシュ値になる可能性が高い。
- イ 同一のパスワードをハッシュ化すると、同じハッシュ値になる。
- ウ パスワードをハッシュ化した結果のハッシュ値を再度ハッシュ化すると、元のパスワードになる。
- エ 秘密鍵を使用してハッシュ値から元のパスワードを復元できる。

※出典：基本情報技術者試験平成30年度春期午後問題問1を改変

問題解説

ハッシュ関数とは、入力されたデータを基に一定の計算を行い、決まったサイズのデータ（ハッシュ値）を出力する関数です。代表的なものに、アメリカのNSAが開発したSHA-256があります。

ハッシュ化に使用するハッシュ関数には、一般に次のような特徴があります。

- ①ハッシュ値は入力値の長さに関係なく固定長になる。
- ②入力値が同じなら、ハッシュ値も同じ値になる。
- ③ハッシュ値から元の入力値を復元することは非常に困難である。
- ④入力値が少しでも違うと、ハッシュ値はまったく異なる値になる。

解答群のうち、これらの特徴に当てはまるのはイの「同一のパスワードをハッシュ化すると、同じハッシュ値になる。」です。

- × ア ハッシュ値は入力値の長さに関係なく同じサイズになるため、異なる入力値が同じハッシュ値になる可能性はゼロではありません。ただし、その確率は非常に低いです。
- イ 正解です。
- × ウ ハッシュ値を再度ハッシュ化しても、元の入力値には戻りません。
- × エ ハッシュ値を元のパスワードに戻す鍵のようなものは存在しません。

○ 解答 ○

問18 イ



合格のカギ ソルト――

パスワードをそのままハッシュ化して保存する方法は、よく使われる単語などのパスワードをハッシュ値に変換したリストを事前に用意し、パスワードファイルと照合することで元のパスワードを知られるおそれがあります。この事前リストの対策として、元のパスワードにソルトと呼ばれるランダムな文字列を連結してハッシュ化する方法が知られています。



19

次の記述中の [a1] , [a2] 及び [b1] , [b2] に入る適切な答えの組合せを、解答群の中から選べ。

X社では、図に示すネットワークを構築し、インターネットへのWebサイトの公開とメールの送受信を行っている。

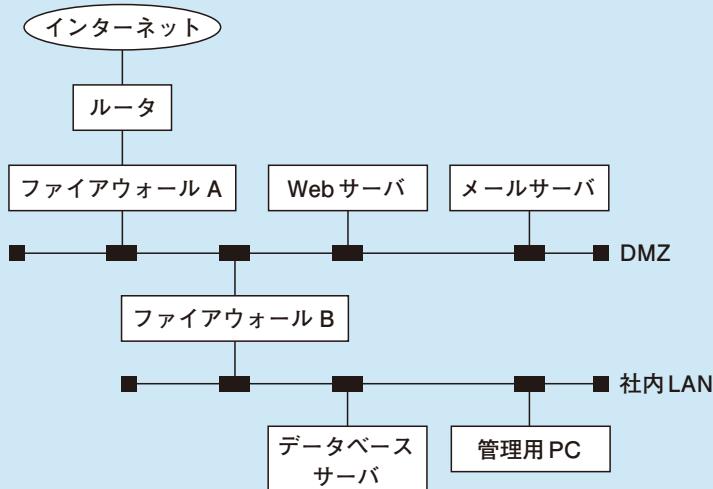


図 X社のネットワーク構成

X社のネットワークは二つのファイアウォールによって、DMZ 及び社内 LAN の二つのセグメントに分けられている。Web サーバ、メールサーバ及びデータベースサーバ（以下、DB サーバという）は、それぞれ表 1 に示す役割を果たしている。

表 1 Web サーバ、メールサーバ、DB サーバの役割

サーバ	役割
Web サーバ	Web サイトとして、自社の情報をインターネットに公開する。Web サーバ上では、社外との取引情報を処理するプログラムが動作する。このプログラムが利用するデータは DB サーバ上に格納される。
メールサーバ	社外とのメールの送受信を行う。また、取引先に対してメールを自動配信するプログラムが動作する。メール配信のためのデータは DB サーバ上に格納される。
DB サーバ	Web サーバ及びメールサーバで利用するデータを格納する。

社内 LAN に接続された管理用 PC からは、SSH を使った各サーバへのログイン操作と、メールサーバを介した外部とのメール送受信が可能である。管理用 PC から自社 Web サーバの参照はできるが、社外 Web サイトの利用は許可されていない。

ネットワーク上で使われるプロトコルとポート番号を表2に示す。

表2 プロトコルとポート番号

サービス	プロトコル	ポート番号
Web	HTTP	80
メール転送	SMTP	25
セキュアシェル（遠隔ログイン）	SSH	22
メール受信	POP3	110
DB アクセス	DB 専用	1999

ファイアウォールAとファイアウォールBのパケットフィルタリングの設定を、それぞれ表3、表4に示す。

フィルタリングの設定ルールは、送信元のIPアドレス、宛先のIPアドレス及び接続先ポート番号を指定して通信の許可／拒否を制御する。設定は上の行のルールから調べて、最初に条件が合致した行の動作を実行する。また、応答パケットについては動的フィルタリング機能によって自動的に許可されるので設定は不要なものとする。

表3 ファイアウォールAのフィルタリング設定

送信元	宛先	ポート番号	動作
任意	Webサーバ	80	許可
任意	メールサーバ	25	許可
a1	任意	a2	許可
任意	任意	任意	拒否

表4 ファイアウォールBのフィルタリング設定

送信元	宛先	ポート番号	動作
Webサーバ	DBサーバ	1999	許可
メールサーバ	DBサーバ	1999	許可
管理者用PC	b1	b2	許可
管理者用PC	メールサーバ	22	許可
管理者用PC	メールサーバ	25	許可
管理者用PC	Webサーバ	80	許可
管理者用PC	Webサーバ	22	許可
任意	任意	任意	拒否

解答群

	a1	a2	b1	b2
ア	DB サーバ	1999	メールサーバ	110
イ	Web サーバ	80	任意	任意
ウ	Web サーバ	80	DB サーバ	1999
エ	管理用 PC	22	メールサーバ	110
オ	管理用 PC	22	任意	任意
カ	メールサーバ	25	メールサーバ	110
キ	メールサーバ	25	DB サーバ	1999
ク	メールサーバ	25	任意	任意
ケ	メールサーバ	110	DB サーバ	1999
コ	メールサーバ	110	任意	任意

※出典：基本情報技術者試験平成21年度春期午後問題4を改変

問題解説

空欄 a1, a2 : ファイアウォール A は、X 社のネットワークが外部とやり取りする通信を制限します。各行の設定は、それぞれ次のような意味があります。

送信元	宛先	ポート番号	動作
任意	Web サーバ	80	許可
任意	メールサーバ	25	許可
a1	任意	a2	許可
任意	任意	任意	拒否

- ←外部から Web サーバへのアクセスを許可
- ←外部からメールサーバへのアクセスを許可
- ← a1 から外部へのアクセスを許可
- ←許可した通信以外すべて遮断

Web サーバ、メールサーバ、DB サーバ、管理用 PC のうち、X 社のネットワークから外部に向けた通信が必要なのは、メール送信を行うメールサーバだけです。メールの送信は SMTP で行うので、ポート番号は 25 が適切です。

以上から、空欄 a1 は「メールサーバ」、空欄 a2 は「25」が入ります。

空欄 b1, b2 : 空欄は、送信元が管理者 PC の場合の設定なので、管理者 PC の役割を問題文から確認しましょう。

社内 LAN に接続された管理者用 PC からは、SSH を使った各サーバへのログイン操作と、メールサーバを介した外部とのメール送受信が可能である。管理者用 PC から自社 Web サーバの参照はできるが、社外 Web サイトの利用は許可されていない。

以上の記述から、管理者用 PC と DMZ にあるサーバとは、

- ① Web サーバへのログイン操作
- ② メールサーバへのログイン操作
- ③ メールサーバを介したメール送信
- ④ メールサーバを介したメール受信
- ⑤ 自社 Web サーバの参照

の5種類の通信があることがわかります。これらを表3の各設定に当てはめると、次のようになります。

送信元	宛先	ポート番号	動作
Web サーバ	DB サーバ	1999	許可
メールサーバ	DB サーバ	1999	許可
管理者用 PC	b1	b2	許可
管理者用 PC	メールサーバ	22	許可
管理者用 PC	メールサーバ	25	許可
管理者用 PC	Web サーバ	80	許可
管理者用 PC	Web サーバ	22	許可
任意	任意	任意	拒否

← ?
← ②
← ③
← ⑤
← ①

以上から、空欄のある行は、④メールサーバを介したメール受信の設定であることがわかります。したがって、空欄b1は「メールサーバ」、空欄b2はメール受信プロトコルPOP3のポート番号「110」が入ります。

空欄aが「メールサーバ」「25」、空欄bが「メールサーバ」「110」となるので、正解は**力**です。

○ 解答 ○

問19 力



合格の力ギ

パケットフィルタリング

ファイアウォールの方式には、ホワイトリスト方式とブラックリスト方式の2種類があります。ホワイトリスト方式は、許可される通信だけを登録し、それ以外はすべて遮断する方式です。ブラックリスト方式はその逆で、遮断する通信だけを登録し、それ以外は許可する方式です。

ファイアウォールのパケットフィルタリングは、一般にホワイトリスト方式なので、問題文から必要な通信を抜き出し、まだリストにないものを空欄に入れます。



20

次の記述の [] に入る適切な答えを、解答群の中から選べ。

A 社は、口コミによる飲食店情報を収集し、提供する会員制サービス業者である。会員制サービスを提供するシステム（以下、A 社システムという）を図 1 に示す。

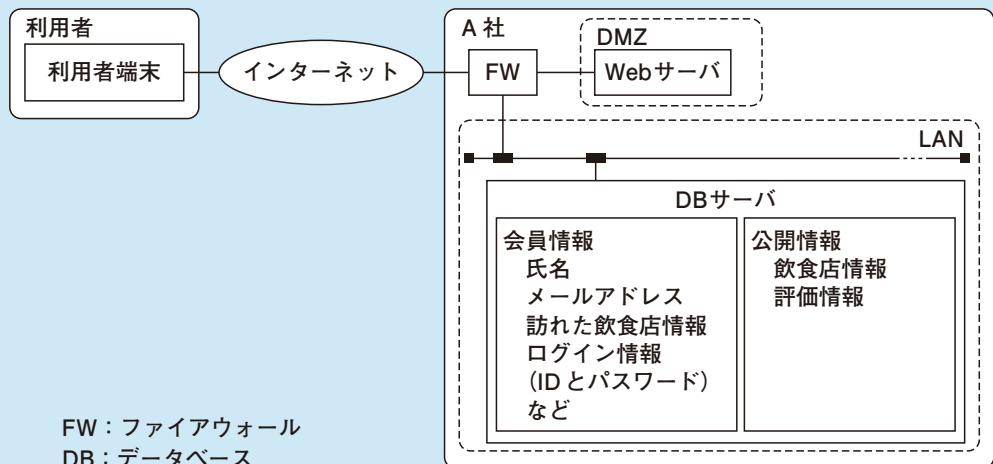


図 1 A 社システム

- (1) FW, Web サーバ及び DB サーバがあり、スマートフォンなどの利用者端末とはインターネットを介して接続されている。
- (2) Web サーバは DMZ に置かれており、DB サーバは LAN に置かれている。また、利用者端末から Web サーバへの接続には、セキュリティを考慮して TLS を用いている。
- (3) 会員登録を行った利用者（以下、会員という）には、ID とパスワードが発行される。
- (4) DB サーバには、会員情報（氏名、メールアドレス、訪れた飲食店情報、ログイン情報（ID とパスワード）など）と公開情報（飲食店情報、評価情報）が保管されている。
- (5) 会員は、公開情報を閲覧することができる。また、Web サーバにログインすることで、DB サーバに保管してある自分の会員情報と自らが書き込んだ公開情報の更新、及び新しい公開情報の追加が行える。
- (6) 非会員は、公開情報の閲覧だけができる。
- (7) 会員が Web サーバにログインするには、ID とパスワードが必要であり、A 社システムは DB サーバに保管してあるログイン情報を用いて認証する。
- (8) Web サーバ及び DB サーバでは、それぞれでアクセスログ（以下、ログという）が記録されており、システム管理者が定期的に内容を確認している。また、システム管理者は、通常、LAN から Web サーバや DB サーバにアクセスして、メンテナンスを行っている。
なお、外部から Telnet や SSH で Web サーバに接続して、インターネットを介したリモートメンテナンスが行えるようにしてあるが、現在はリモートメンテナンスの必要性はなくなっている。

ある日、システム管理者が、ログの確認において、通常とは異なるログが記録されているのを発見した。そのログを詳しく調査したところ、システム管理者以外の者が管理者IDと管理者パスワードを使ってWebサーバに不正侵入したことが明らかになった。

そこで、システム管理者は上司と相談し、会員制サービスを直ちに停止した。次に、今回の不正侵入に対する被害状況の特定と対策の検討を行った。不正侵入による被害状況を図2に示す。

- (一) Webサーバへの不正侵入があったことが確認された。秘密鍵への不正アクセスがあつたかは確認できなかった。
- (二) FWを経由し、Webサーバに不正侵入され、さらにそこからDBサーバに不正侵入された。
- (三) 一部の会員については会員情報が漏えいしたことが分かっているが、それ以外の会員については漏えいの有無を特定できていない。

図2 不正侵入による被害状況

図2中の項番（一）～（三）のそれぞれに対して、次のような対策の検討を行った。

- ・項番（一）の被害への対策として、公開鍵証明書の再発行手続を行い、秘密鍵は同じものを使用することにした。
- ・項番（二）の被害への対策として、リモートメンテナンス用ポートへのインターネットからのアクセスをFWで禁止し、TelnetやSSHのポートは閉じることにした。
- ・項番（三）の被害への対策として、管理者パスワードは変更し、漏えいした会員だけにパスワードの変更を依頼することにした。

以上のうち、適切な対策といえるのは [] である。

解答群

- ア 項番（一）の対策のみ
- イ 項番（二）の対策のみ
- ウ 項番（三）の対策のみ
- エ 項番（一）と項番（二）の対策
- オ 項番（一）と項番（三）の対策
- カ 項番（二）と項番（三）の対策
- キ 項番（一）、項番（二）、項番（三）の対策
- ク なし

※出典：基本情報技術者試験平成28年度春期午後問題問1を改変

問題解説

(一) Web サーバへの不正侵入があったことが確認された。秘密鍵への不正アクセスがあったかは確認できなかった。

利用者端末と Web サーバとの接続には、TLS (Transport Layer Security) とは、インターネット上での通信を暗号化するプロトコルです。また、TLS は公開鍵証明書にもとづいて、サーバを認証する機能も備えています。

Web サーバに保管されている秘密鍵は、利用者端末との暗号通信に用いるため、秘密鍵が第三者に漏えいすると、暗号通信が盗聴され、解読されてしまうおそれがあります。

今回の不正侵入では、秘密鍵への不正アクセスは確認できませんでしたが、不正アクセスがなかったとも確認できないため、秘密鍵は作り直したほうが安全です。秘密鍵を作り直す場合は、それとペアとなる公開鍵も作り直しになります。したがって、公開鍵証明書も再発行が必要です。

以上から、項番(一)の被害への対策として、「公開鍵証明書の再発行手続を行い、秘密鍵は同じものを使用することにした」は適切ではありません。

(二) FW を経由し、Web サーバに不正侵入され、さらにそこから DB サーバに不正侵入された。

侵入者はどうやって Web サーバや DB サーバに侵入したのでしょうか？ 問題文の次の記述がヒントになります。

なお、外部から Telnet や SSH で Web サーバに接続して、インターネットを介したリモートメンテナンスが行えるようにしてあるが、現在はリモートメンテナンスの必要性はなくなっている。

Telnet と SSH は、どちらも外部からサーバにログインして、サーバを遠隔操作するプロトコルです。これらのプロトコル用のポートが、不正侵入の入口になったことは十分に考えられます。とくに Telnet は古くからあるプロトコルで、通信を暗号化する機能がないため、セキュリティ上問題があります。

しかも、「現在はリモートメンテナンスの必要性はなくなっている」のですから、これらのポートは使用不可にしておくべきです。

以上から、項番(二)の被害への対策として、「リモートメンテナンス用ポートへのインターネットからのアクセスを FW で禁止し、Telnet や SSH のポートは閉じることにした」は適切な対策と言えます。

(三) 一部の会員については会員情報が漏えいしたことが分かっているが、それ以外の会員については漏えいの有無を特定できていない。

侵入者は管理者 ID と管理者パスワードを使って侵入しているので、管理者パスワードは当然変更しなければなりません。会員のパスワードについては全員のパスワードが漏えいしたとは限りませんが、漏えいしたパスワードが特定できない以上、全会員にパスワードの変更を依頼すべきです。

したがって、項番(三)の被害への対策として、「管理者パスワードは変更し、漏えいした会員だけにパスワードの変更を依頼することにした」は適切とは言えません。

解答

問20 1



合格のカギ

セキュリティ上の問題が発生して、受験者に対策を考えさせる問題です。このような問題では、最初のシステム概要に必ず解答のヒントがあります。

基本情報技術者

予想問題③

• 科目A 208

(全 60 問 試験時間 : 90 分)

• 科目B 246

(全 20 問 試験時間 : 100 分)



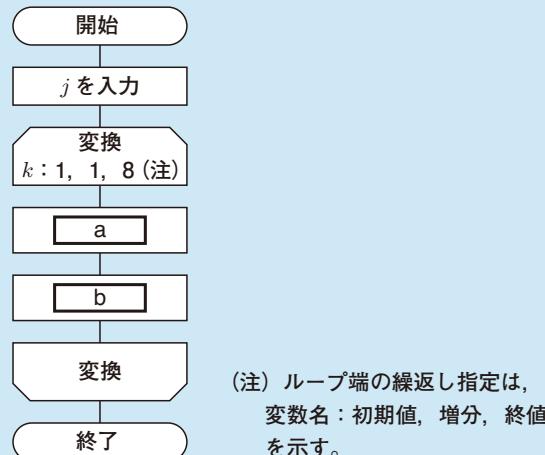
基本情報技術者試験 予想問題③

科目A



問 01

次の流れ図は、10進整数 j ($0 < j < 100$) を8桁の2進数に変換する処理を表している。2進数は下位桁から順に、配列の要素 NISHIN(1) から NISHIN(8) に格納される。流れ図の a 及び b に入る処理はどれか。ここで、 $j \text{ div } 2$ は j を2で割った商の整数部分を、 $j \text{ mod } 2$ は j を2で割った余りを表す。



	a	b
ア	$j \leftarrow j \text{ div } 2$	$\text{NISHIN}(k) \leftarrow j \text{ mod } 2$
イ	$j \leftarrow j \text{ mod } 2$	$\text{NISHIN}(k) \leftarrow j \text{ div } 2$
ウ	$\text{NISHIN}(k) \leftarrow j \text{ div } 2$	$j \leftarrow j \text{ mod } 2$
エ	$\text{NISHIN}(k) \leftarrow j \text{ mod } 2$	$j \leftarrow j \text{ div } 2$



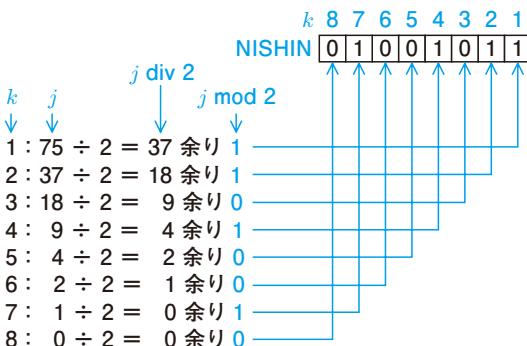
問 02

8ビットの値の全ビットを反転する操作はどれか。

- ア 16進表記 00 のビット列と排他的論理和をとる。
- イ 16進表記 00 のビット列と論理和をとる。
- ウ 16進表記 FF のビット列と排他的論理和をとる。
- エ 16進表記 FF のビット列と論理和をとる。

問01 10進→2進変換のアルゴリズム キホン! ■■■ 40%

10進数を2進数に変換するには、10進数を次々に2で割っていき、その余りを2進数の各桁に順番に格納していきます。たとえば、10進数の75を2進数に変換する場合は次のようにになります。



この処理は、変数 k の値を1から8まで1ずつ増やしながら、次の①、②を繰り返すことで実現できます。

- ① 10進数 j を2で割った余りを求め、結果を2進数の k 桁目に格納する。
- ② 10進数 j を2で割った商の整数部分を j に格納する。

流れ図の空欄aには、上の①の処理「NISHIN(k) \leftarrow $j \bmod 2$ 」が、空欄bには、②の処理「 $j \leftarrow j \div 2$ 」が入ります。以上から、正解は **工** です。

問02 ビット演算 キホン!

■■■ 80%

ビット列中の0のビットを1に、1のビットを0にするビット演算を考えます。「0 XOR 1」は1、「1 XOR 1」は0になるので、各ビットと1との**排他的論理和**(XOR)をとれば、各ビットが反転します。

01010101 ← 元のビット列(例)
 XOR 11111111 ← 16進数 FF
 10101010 ← 全ビットが反転

以上から、2進数の11111111(16進表記FF)のビット列と排他的論理和をとれば、全ビットが反転します。正解は **ウ** です。

- × **ア** 16進表記00との排他的論理和では、元のビット列と変わりません。
- × **イ** 16進表記00との論理和では、元のビット列と変わりません。
- **ウ** 正解です。
- × **エ** 16進表記FFとの論理和は、全ビットが1になります。

2進数	16進数
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

2進数	16進数
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

合格のカギ

覚えよう! 問01

10進数→2進数変換

10進数を次々に2で割っていき、その余りを下から順に並べる

$$\begin{array}{r} 10 \div 2 = 5 \text{ 余り } 0 \\ 5 \div 2 = 2 \text{ 余り } 1 \\ 2 \div 2 = 1 \text{ 余り } 0 \\ 1 \div 2 = 0 \text{ 余り } 1 \end{array}$$

↑
1010

2進数→10進数変換

2進数の各桁に2の $(n-1)$ 乗の重みを掛け、それらを合計する

2進数	1	0	0	1	1	1	0	0
重み	128	64	32	16	8	4	2	1

$10\text{進数 } 128+0+0+16+8+4+0+0 = 156$
 $(2^0=1)$

論理和(OR) 問02

2つのビットが両方とも0のとき0、それ以外は1になる論理演算。

$$\begin{array}{l} 0 \text{ OR } 0 \rightarrow 0 \\ 0 \text{ OR } 1 \rightarrow 1 \\ 1 \text{ OR } 0 \rightarrow 1 \\ 1 \text{ OR } 1 \rightarrow 1 \end{array}$$

排他的論理和(XOR) 問02

2つのビットが両方とも同じ値のとき0、それ以外は1になる論理演算。

$$\begin{array}{l} 0 \text{ XOR } 0 \rightarrow 0 \\ 0 \text{ XOR } 1 \rightarrow 1 \\ 1 \text{ XOR } 0 \rightarrow 1 \\ 1 \text{ XOR } 1 \rightarrow 0 \end{array}$$

○ 解答 ○

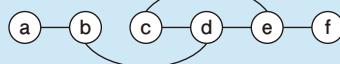
問01 **工** 問02 **ウ**

問 03

ノードとノードの間のエッジの有無を、隣接行列を用いて表す。ある無向グラフの隣接行列が次の場合、グラフで表現したものはどれか。ここで、ノードを隣接行列の行と列に対応させて、ノード間にエッジが存在する場合は 1 で、エッジが存在しない場合は 0 で示す。

$$\begin{array}{cccccc} & a & b & c & d & e & f \\ a & \left[\begin{array}{cccccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right] \end{array}$$

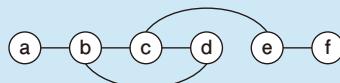
ア



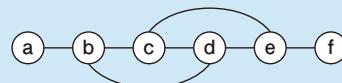
イ



ウ



エ



問 04

機械学習における教師あり学習の説明として、最も適切なものはどれか。

- ア 個々の行動に対しての善しあしを得点として与えることによって、得点が最も多く得られるような方策を学習する。
- イ コンピュータ利用者の挙動データを蓄積し、挙動データの出現頻度に従って次の挙動を推論する。
- ウ 正解のデータを提示したり、データが誤りであることを指摘したりすることによって、未知のデータに対して正誤を得ることを助ける。
- エ 正解のデータを提示せずに、統計的性質や、ある種の条件によって入力パターンを判定したり、クラスタリングしたりする。

問 05

10進法で5桁の数 $a_1 a_2 a_3 a_4 a_5$ を、ハッシュ法を用いて配列に格納したい。ハッシュ関数を $\text{mod } (a_1 + a_2 + a_3 + a_4 + a_5, 13)$ とし、求めたハッシュ値に対応する位置の配列要素に格納する場合、54321は配列のどの位置に入るか。ここで、 $\text{mod } (x, 13)$ は、 x を 13 で割った余りとする。

位置	配列
0	
1	
2	
⋮	⋮
11	
12	

ア 1

イ 2

ウ 7

エ 11

問03 グラフ

グラフとは、複数のノード（頂点）と、ノード間を結ぶエッジ（辺）で構成されるデータ構造です。また、エッジに方向性がないグラフを無向グラフといいます。

隣接行列は、ノード間のエッジの有無を二次元配列で表したもので、無向グラフでは、 $a - b$ 間にエッジが存在すれば、 $b - a$ にも必ずエッジが存在します。したがって、隣接行列は対角線をはさんで対称になります。

この隣接行列から、値が1（エッジが存在する）のノードの組合せを抜き出します。1は全部で12個あり、それぞれに対称となるエッジがあるので、エッジの数は6個になります。

$$a - b, b - c, b - d, c - d, c - e, e - f$$

これらのエッジを図で表すと、右の図のようになります。

解答群のうち、右の図と同じグラフはウなので、ウが正解です。

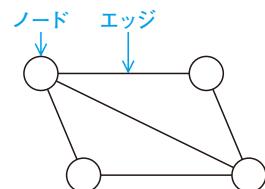
■ 10%

	a	b	c	d	e	f
a	0	1	0	0	0	0
b	1	0	1	1	0	0
c	0	1	0	1	1	0
d	0	1	1	0	0	0
e	0	0	1	0	0	1
f	0	0	0	0	1	0

合格のカギ

問03 グラフ

複数のノード（頂点）と、ノード間を結ぶエッジ（辺）で構成されるデータ構造。エッジに方向性がないグラフを特に無向グラフという。



問03

問04 教師あり学習と教師なし学習

■ 30%

教師あり学習とは、正解付きのデータを例題として与えて、一連のデータに共通する特徴を学習させ、未知のデータの正誤を判別できるようにする手法です。一方、教師なし学習は、正解を与えずに多数のデータを入力し、その中にある未知のパターンや関係性をコンピュータ自身に発見させる手法です。

- ア 強化学習の説明です。
- イ トランスクレシジョンの説明です。
- ウ 正解です。
- エ 教師なし学習の説明です。

問04 機械学習

人間が自然に行っている学習能力と同様の機能を、コンピュータにもたせるための技術。

問04

問04 強化学習

個々の行動に応じて得点を与え、得点が最も多く得られるような方策を発見させる手法。人工知能に将棋や囲碁などを学習させる場合に使われる。

問04

トランスクレシジョンの説明です。
観測されたデータから、新たな出力を予測する機械学習の手法。

問04

問05 ハッシュ法

■ 30%

$a_1 = 5, a_2 = 4, a_3 = 3, a_4 = 2, a_5 = 1$ として、 $\text{mod}(a_1 + a_2 + a_3 + a_4 + a_5, 13)$ を計算します。

$$\text{mod}(5 + 4 + 3 + 2 + 1, 13) = \text{mod}(15, 13) = 2 \quad \leftarrow 15\text{を}13\text{で割った余り}$$

以上から、54321は配列の2の位置に入ります。正解はイです。

この方式で配列に値を格納していくと、後で配列から値54321を検索する場合にも、同様の計算ですぐに配列の2番目から値を取り出せます。

位置	配列
0	12343
1	82121
2	54321
:	:
11	44556
12	55663

← 格納位置を計算で求めることができます

問05 ハッシュ法

データの格納位置を、ハッシュ関数によってデータ自体から計算する方法。データを検索する際には、同様の計算で格納位置をすばやく求めることができます。ただし、データによっては格納位置が重なる場合がある（衝突という）ため、その場合の処理手順が必要となる。

問05

○ 解答
問03 ウ 問04 ウ
問05 イ

問 06

配列 A が図 2 の状態のとき、図 1 の流れ図を実行すると、配列 B が図 3 の状態になった。図 1 の a に入れる操作はどれか。ここで、配列 A , B の要素をそれぞれ $A(i, j)$, $B(i, j)$ とする。

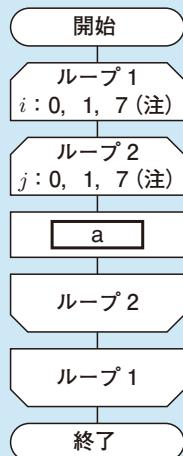


図 1 流れ図

		j							
		0	1	2	3	4	5	6	7
i	0	*	*	*	*	*	*	*	
	1	*							
	2	*							
	3	*	*	*	*				
	4	*							
	5	*							
	6	*							
	7	*							

図 2 配列 A の状態

		j							
		0	1	2	3	4	5	6	7
i	0	*	*	*	*	*	*	*	
	1	*	*	*	*	*	*	*	*
	2				*				*
	3					*			*
	4						*		*
	5							*	
	6								*
	7								

図 3 実行後の配列 B の状態

(注) ループ端の繰返し指定は、変数名：初期値、増分、終値を示す。

- ア $B(7 - i, 7 - j) \leftarrow A(i, j)$
 ウ $B(i, 7 - j) \leftarrow A(i, j)$

- イ $B(7 - j, i) \leftarrow A(i, j)$
 エ $B(j, 7 - i) \leftarrow A(i, j)$

問 07

A, C, K, S, T の順に文字が入力される。スタックを利用して、 S, T, A, C, K という順に文字を出力するために、最小限必要となるスタックは何個か。ここで、どのスタックにおいてもポップ操作が実行されたときには必ず文字を出力する。また、スタック間の文字の移動は行わない。

ア 1

イ 2

ウ 3

エ 4

問 08

自然数 n に対して、次のとおり再帰的に定義される関数 $f(n)$ を考える。 $f(5)$ の値はどれか。

$f(n)$: if $n \leq 1$ then return 1 else return $n + f(n - 1)$

ア 6

イ 9

ウ 15

エ 25

解説

問 06 二次元配列の操作

図 2 と図 3 を比べると、図 2 の内容を 90° 右に回転したものが図 3 です。配列 A と配列 B の対応する要素は、次のようになります。



20%

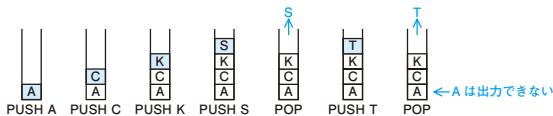
$$\begin{array}{ll}
 i & j \\
 A(0, 0) \rightarrow B(0, 7) & A(1, 0) \rightarrow B(0, 6) \\
 A(0, 1) \rightarrow B(1, 7) & A(1, 1) \rightarrow B(1, 6) \\
 A(0, 2) \rightarrow B(2, 7) & A(1, 2) \rightarrow B(2, 6) \\
 \vdots & \vdots \quad \vdots \quad \vdots \\
 A(0, 7) \rightarrow B(7, 7) & A(1, 7) \rightarrow B(7, 6)
 \end{array}$$

このように、配列 A の要素を $A(i, j)$ とすれば、対応する配列 B の要素は $B(j, 7-i)$ と表すことができます。以上から、空欄 **a** には「 $B(j, 7-i) \leftarrow A(i, j)$ 」が入ります。正解は **工** です。

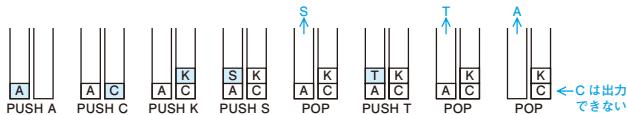
問07 スタックの個数

 60%

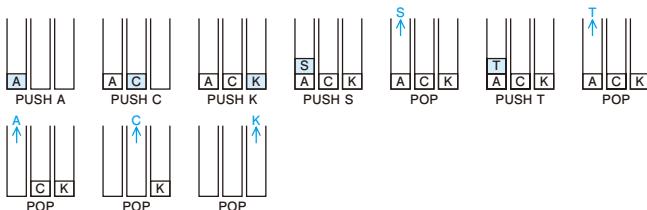
スタックが 1 個しかない場合は、次のように S, T までは出力できますが、A が出力できません。



スタックを 2 個にすると、今度は次のように S, T, A まで出力できるようになりますが、C が出力できません。



スタックを 3 個にすれば、次のようにすべての文字を出力できます。以上から、正解は **ウ** です。



覚えよう！

問07

スタックといえば

- データを順番に格納していく、取り出すときは後から格納したデータから取り出される後入れ先出し型のデータ構造

プッシュ：データをスタックに格納する

ポップ：スタックからデータを取り出す

問08 再帰的関数 キホン！

 60%

処理の中で自分自身を呼び出す関数を**再帰的関数**といいます。関数 $f(n)$ は、 n の値が 1 より大きければ $n + f(n - 1)$ を返し、 n の値が 1 以下のときは 1 を返します。したがって、 $f(5)$ の値は次のように計算できます。

$$\begin{aligned}
 f(5) &= 5 + f(4) \\
 &= 5 + 4 + f(3) \leftarrow n + f(n-1) \text{ を返す} \\
 &= 5 + 4 + 3 + f(2) \leftarrow n + f(n-1) \text{ を返す} \\
 &= 5 + 4 + 3 + 2 + f(1) \leftarrow n + f(n-1) \text{ を返す} \\
 &= 5 + 4 + 3 + 2 + 1 \leftarrow n \leq 1 \text{ なので } 1 \text{ を返す}
 \end{aligned}$$

以上から、正解は **ウ** です。

再帰的関数

問08

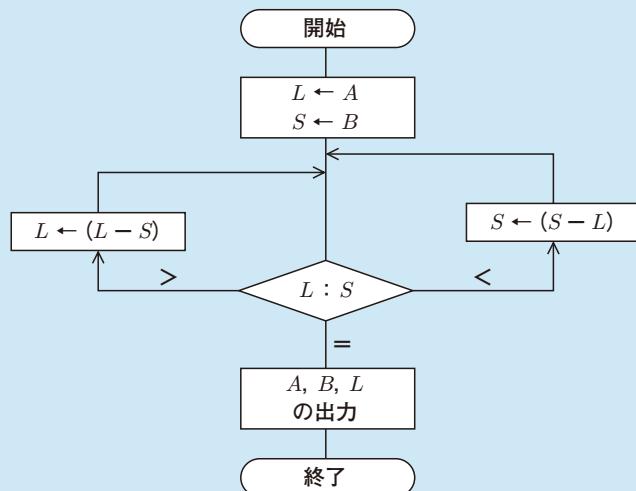
処理の中で自分自身を呼び出す関数。定義の中に、必ず底入れの規定が含まれる。この規定がないと無限に自分自身を呼び出してしまう。

 解答

 問06 工 問07 ウ
問08 ウ

問 09

次の流れ図は、2数 A, B の最大公約数を求めるユークリッドの互除法を、引き算の繰返しによって計算するものである。 A が 876, B が 204 のとき、何回の比較で処理は終了するか。



ア 4

イ 9

ウ 10

エ 11

問 10

複数のプロセスから同時に呼び出されたときに、互いに干渉することなく並行して動作することができるプログラムの性質を表すものはどれか。

ア リエントラント

イ リカーシブ

ウ リユーザブル

エ リロケータブル

問 11

1GHz のクロックで動作する CPU がある。この CPU は、機械語の 1 命令を平均 0.8 クロックで実行できることが分かっている。この CPU は 1 秒間に平均何万命令を実行できるか。

ア 125

イ 250

ウ 80,000

エ 125,000

問09 流れ図のトレース

■■■■■ 40%

ユークリッドの互除法は、2つの数の最大公約数を求める古典的なアルゴリズムです。変数 L , S の値の変化をトレースすると、次のようにになります。比較回数は全部で11回なので、正解は **工** です。

比較回数	L	S	$L : S$	次の処理
1	876	204	>	$L \leftarrow (L - S)$
2	672	204	>	$L \leftarrow (L - S)$
3	468	204	>	$L \leftarrow (L - S)$
4	264	204	>	$L \leftarrow (L - S)$
5	60	204	<	$S \leftarrow (S - L)$
6	60	144	<	$S \leftarrow (S - L)$
7	60	84	<	$S \leftarrow (S - L)$
8	60	24	>	$L \leftarrow (L - S)$
9	36	24	>	$L \leftarrow (L - S)$
10	12	24	<	$S \leftarrow (S - L)$
11	12	12	=	出力



ユークリッドの互除法

問09

2つの自然数 a , b ($a > b$) の最大公約数を計算する方法。 a を b で割った剰余 c を求め、除数 b を c で割った剰余、除数 c をその剰余で割った剰余…のように、剰余を求める計算を繰り返し、剰余が0になったときの除数を a と b の最大公約数とする。

$$\begin{aligned} 876 \div 204 &= 4 \text{ 余り } 60 \\ 204 \div 60 &= 3 \text{ 余り } 24 \\ 60 \div 24 &= 2 \text{ 余り } 12 \\ 24 \div 12 &= 2 \text{ 余り } 0 \end{aligned}$$

なお、本問では割り算の代わりに引き算を行っている。

問10 複数プロセスの実行

■■■■■ 20%

複数のプロセスから同時に呼び出されても、問題なく並行して動作することができるプログラムの性質を、リエンタント（再入可能）といいます。

- **ア** 正解です。
- × **イ** リカーシブ（再帰的）とは、関数定義の中で自分自身を呼び出す処理のことをいいます。
- × **ウ** リユーザブル（再使用可能）とは、いったん主記憶上に読み込まれたプログラムを、読み込み直さなくても再び実行できることをいいます。
- × **エ** リロケータブル（再配置可能）とは、主記憶上のどこのアドレスに配置しても実行できることをいいます。

リエンタント（再入可能）

問10

複数プロセスから同時に呼び出されても、問題なく動作するプログラムのこと。プログラムを手続き部分とデータ部分に分け、データ部分を各プロセスごとに独立させることで可能になる。

問11 CPU処理速度の計算 キホン!

■■■■■ 20%

クロック周波数が $1\text{GHz} = 10^9\text{Hz}$ ということは、1秒間に 10^9 クロックで動作するということです。1命令は平均 0.8 クロックで実行されるので、1秒間に実行できる命令数は、次のように計算できます。

$$10^9 \div 0.8 = 10^9 \div \frac{8}{10} = 10^9 \times \frac{10}{8} = 10^9 \times 1.25 = 125000 \times 10^4$$

万単位

以上から、1秒間に 125,000 万命令となります。正解は **工** です。

クロック周波数

問11

1秒間に発生するクロックの数。クロックとは、PCの各装置の動作のタイミングを合わせるための信号で、CPUはクロックに合わせて命令を実行する。

○ 解答 ○

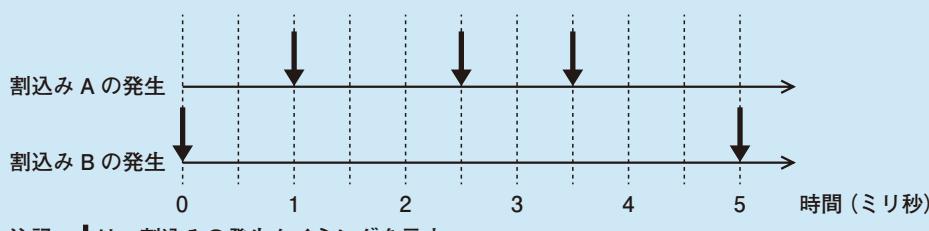
問09 **工** 問10 **ア**
問11 **工**



問 12

メイン処理、及び表に示す二つの割込み A、B の処理があり、多重割込みが許可されている。割込み A、B が図のタイミングで発生するとき、0 ミリ秒から 5 ミリ秒までの間にメイン処理が利用できる CPU 時間は何ミリ秒か。ここで、割込み処理の呼出し及び復帰に伴うオーバヘッドは無視できるものとする。

割込み	処理時間(ミリ秒)	割込み優先度
A	0.5	高
B	1.5	低



注記 ↓は、割込みの発生タイミングを示す。

ア 2

イ 2.5

ウ 3.5

エ 5



問 13

RAID の分類において、ミラーリングを用いることで信頼性を高め、障害発生時には冗長ディスクを用いてデータ復元を行う方式はどれか。

ア RAID1

イ RAID2

ウ RAID3

エ RAID4



問 14

2 台の処理装置から成るシステムがある。少なくともいずれか一方が正常に動作すればよいときの稼働率と、2 台とも正常に動作しなければならないときの稼働率の差は幾らか。ここで、処理装置の稼働率はいずれも 0.9 とし、処理装置以外の要因は考慮しないものとする。

ア 0.09

イ 0.10

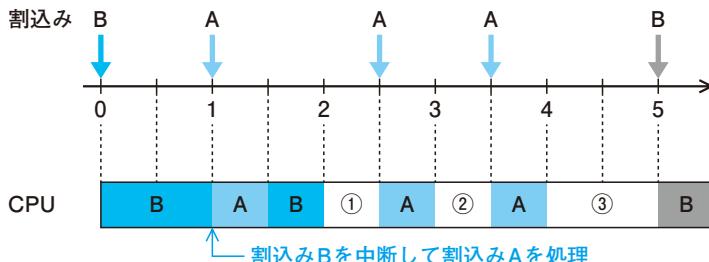
ウ 0.18

エ 0.19

問12 割込み処理

■■■ 40%

多重割込みが可能なので、割込み B の処理中に優先度の高い割込み A が発生すると、割込み B を中断して割込み A を先に処理します。0 ミリ秒から 5 ミリ秒までの CPU の処理は、次のようにになります。



メイン処理が CPU を利用できるのは、上図の①、②、③の部分です。この部分の CPU 時間の合計は $0.5 + 0.5 + 1.0 = 2$ ミリ秒なので、正解は ア です。

問13 RAIDの分類

■■■ 30%

RAID とは、複数の磁気ディスクを並列に接続して信頼性の向上やアクセスの高速化を図る技術です。RAID には RAID0、RAID1、RAID5などの種類がありますが、ミラーリング（ディスク 2 台に同一のデータを書き込む方式）によって、1 台に障害が発生してもデータを復元できるようにする方式は RAID1 です。以上から、正解は ア です。

問14 稼働率の計算 キホン！

■■■ 90%

「少なくともいずれか一方が正常に動作すればよいとき」の稼働率は、2 台の処理装置を並列に接続した場合なので、次のように計算できます。

$$1 - (1 - 0.9) \times (1 - 0.9) = 1 - 0.1 \times 0.1 = 1 - 0.01 = 0.99$$

2台とも故障する確率

一方、「2 台とも正常に動作しなければならないとき」の稼働率は、2 台の処理装置を直列に接続した場合なので、次のように計算できます。

$$0.9 \times 0.9 = 0.81$$

2台とも稼働する確率

2 つの稼働率の差は $0.99 - 0.81 = 0.18$ です。正解は ウ です。



問12 割込み

実行中の処理を中断して、CPU に強制的に別の処理を実行させること。

問13 RAIDの分類

RAID0	データを分散書き込みしてアクセスを高速化（ストライピング）
RAID1	ディスク 2 台に同一データを書き込む（ミラーリング）
RAID2	データと ECC を比特単位で分散書き込み
RAID3	データをバイト単位で分散書き込みし、parity を専用ディスクに書き込む
RAID4	データをブロック単位で分散書き込みし、parity を専用ディスクに書き込む
RAID5	データをブロック単位で分散書き込みし、parity データも分散して書き込む

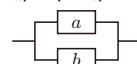
覚えよう！

直列システムの稼働率

$$a \times b \quad \boxed{a} \rightarrow \boxed{b}$$

並列システムの稼働率

$$1 - (1 - a) \times (1 - b)$$



○ 解答 ○

問12 ア 問13 ア
問14 ウ



問

15

冗長構成におけるデュアルシステムの説明として、適切なものはどれか。

- ア 2系統のシステムで並列処理をすることによって性能を上げる方式である。
- イ 2系統のシステムの負荷が均等になるように、処理を分散する方式である。
- ウ 現用系と待機系の2系統のシステムで構成され、現用系に障害が生じたときに、待機系が処理を受け継ぐ方式である。
- エ 一つの処理を2系統のシステムで独立を行い、結果を照合する方式である。



問

16

バックアップ方式の説明のうち、増分バックアップはどれか。ここで、最初のバックアップでは、全てのファイルのバックアップを取得し、OSが管理しているファイル更新を示す情報はリセットされるものとする。

- ア 最初のバックアップの後、ファイル更新を示す情報があるファイルだけをバックアップし、ファイル更新を示す情報は変更しないでそのまま残しておく。
- イ 最初のバックアップの後、ファイル更新を示す情報にかかわらず、全てのファイルをバックアップし、ファイル更新を示す情報はリセットする。
- ウ 直前に行ったバックアップの後、ファイル更新を示す情報があるファイルだけをバックアップし、ファイル更新を示す情報はリセットする。
- エ 直前に行ったバックアップの後、ファイル更新を示す情報にかかわらず、全てのファイルをバックアップし、ファイル更新を示す情報は変更しないでそのまま残しておく。



問

17

DRAMの特徴はどれか。

- ア 書込み及び消去を一括又はブロック単位で行う。
- イ データを保持するためのリフレッシュ操作又はアクセス操作が不要である。
- ウ 電源が遮断された状態でも、記憶した情報を保持することができる。
- エ メモリセル構造が単純なので高集積化することができ、ビット単価を安くできる。

解説

**問15 デュアルシステム**

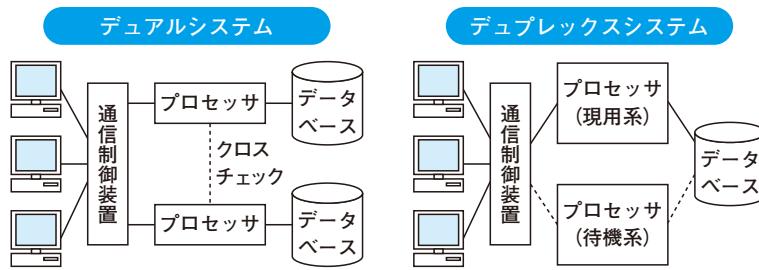
30%

デュアルシステムは、同じ処理を行うシステムを並列に稼働させ、一方に障害が発生しても処理を継続できるようにしたシステムです。2系統が正常に動

合格のカギ

作しているかどうかは互いに処理結果を照合することで確認します。

- A** **並行システム**の説明です。並列処理によって性能を上げるのは、冗長構成ではありません。
- I** 分散システムの説明です。負荷の分散は冗長構成ではありません。
- W** **デュプレックスシステム**の説明です。
- E** 正解です。



問16 バックアップ方式

30%

増分バックアップとは、前回行ったバックアップ以降に更新されたファイルだけをバックアップする方式です。

- A** 最初のバックアップ以降に更新されたファイル（最初のバックアップとの差分）を毎回バックアップする方式で、**差分バックアップ**の説明です。
- I** **フルバックアップ**の説明です。ふだんは差分バックアップや増分バックアップ方式でバックアップする場合でも、定期的にこの方式でフルバックアップをとらないとデータの復旧に時間がかかります。
- W** 正解です。バックアップ後に更新を示す情報をリセットするので、前回行ったバックアップ以降に更新されたファイルだけがバックアップされます。
- E** フルバックアップの説明ですが、ファイル更新を示す情報を変更しないので、バックアップ以後に更新されたファイルを判別できません。ただし、毎回フルバックアップを行う場合は、この方式でもかまいません。

問17 DRAM キホン！

80%

DRAMは、電源を切ると内容が消えてしまう揮発性メモリの一種です。構造が単純なので大容量化・高集積化が簡単にでき、ビット当たりの単価を安くできます。そのため、主に主記憶装置に用いられています。

- A** 書込みや消去をブロック単位で行うのは、**フラッシュメモリ**の特徴です。
- I** DRAMはデータを保持するために、一定時間ごとに内容を再書き込む**リフレッシュ**が必要です。
- W** 電源を切った後でも記憶した情報を保持できるのは、EPROM、フラッシュメモリといった不揮発性メモリの特徴です。
- E** 正解です。

冗長構成

問15

設備や装置を複数用意し、一部が故障しても処理を継続できるようにシステムを構成すること。

覚えよう！

問16

増分バックアップといえば

- 前回のバックアップ以後に更新されたファイルだけをバックアップ
- データ復旧には、前回行ったフルバックアップのデータと、それ以後のすべての増分バックアップデータが必要

差分バックアップといえば

- 前回のフルバックアップ以後に更新されたファイルをバックアップ
- データ復旧には、前回行ったフルバックアップのデータと、最新の差分バックアップデータ1回分が必要

覚えよう！

問17

DRAM といえば

- 構造が単純で比較的安価に高集積化が可能
- リフレッシュ動作が必要
- 主記憶装置に用いる

フラッシュメモリ

問17

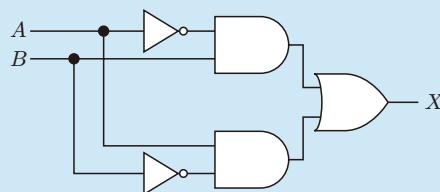
電源を切っても内容が消えない不揮発性メモリの一種。電気的に内容を消去・書換えできる。

○ 解答 ○

- | | | | |
|-----|----|-----|---|
| 問15 | 正解 | 問16 | ウ |
| 問17 | 正解 | | |

**問 18**

次の回路の入力と出力の関係として、正しいものはどれか。

**ア**

入力		出力
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

イ

入力		出力
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

ウ

入力		出力
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

エ

入力		出力
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

**問 19**

コードから商品の内容が容易に分かるようにしたいとき、どのコード体系を選択するのが適切か。

ア 区分コード**イ** 桟別コード**ウ** 表意コード**エ** 連番コード**問 20**

インタプリタの説明として、適切なものはどれか。

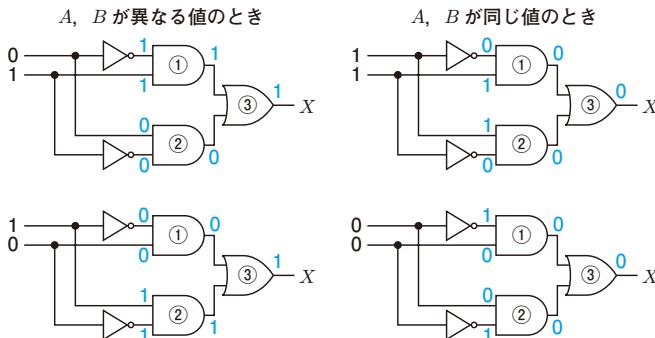
ア 原始プログラムを、解釈しながら実行するプログラムである。**イ** 原始プログラムを、推論しながら翻訳するプログラムである。**ウ** 原始プログラムを、目的プログラムに翻訳するプログラムである。**エ** 実行可能なプログラムを、主記憶装置にロードするプログラムである。

問18 論理回路

■ 90%

下図の論理積素子①の出力が 1 になるのは、A が 0 で B が 1 の場合です。また、論理積素子②の出力が 1 になるのは、A が 1 で B が 0 の場合です。どちらの場合でも、論理和素子③の出力は 1 になります。

一方、A、B ともに 1 の場合と、A、B ともに 0 の場合は、論理積素子①の出力も論理積素子②の出力も 0 になるので、論理和素子③の出力は 0 です。



以上から、A、B が異なる値のとき出力 X が 1、A、B が同じ値のとき出力 X が 0 になる イ が正解です。

問19 コード体系の選択

■ 20%

コードに意味のある記号や略称を含めて、商品の内容をコードから識別できるようにしたものをお表意コードといいます。たとえば、商品の色を示す「BK」「RD」「BL」などの記号をコードに含めるのは、表意コードの例です。

- × **ア** **区分コード**は、1000 番台は営業部、2000 番台は開発部、3000 番台は製造部のように、コードを複数にグループ分けしたコード体系です。
- × **イ** **桁別コード**は、上位 2 衔を学年、下位 4 衔を出席番号のように、コードの各桁に意味を持たせたコード体系です。
- **ウ** 正解です。
- × **エ** **連番コード**は、伝票番号などのように、発生順に連番を振っていくコード体系です。

問20 インタプリタ

■ 20%

プログラム言語は人間が記述しやすいように、人間の言語に近い単語や文法でできています。プログラム言語で記述したプログラムを**原始プログラム**（ソースプログラム、ソースコードともいう）といいます。

原始プログラムは、コンピュータが理解できる機械語に変換されて、はじめて実行可能なプログラムになります。インタプリタやコンパイラは、この変換を行うプログラムです。**インタプリタ**は、原始プログラムを機械語に逐語的に解釈しながら実行します。正解は **ア** です。



問18 論理回路

問18

論理否定素子 (NOT)

論理積素子 (AND)

論理和素子 (OR)

問18 論理否定 (NOT) 1のとき 0、0 のとき 1 になる。

問18 論理積 (AND) 2つのビットが両方とも 1 のとき 1、それ以外は 0 になる。

0 AND 0 → 0
0 AND 1 → 0
1 AND 0 → 0
1 AND 1 → 1

問18 論理和 (OR) 2つのビットが両方とも 0 のとき 0、それ以外は 1 になる。

0 OR 0 → 0
0 OR 1 → 1
1 OR 0 → 1
1 OR 1 → 1

覚えよう！ 問20

インタプリタといえば

- ソースプログラムを解釈しながら実行する

コンパイラといえば

- ソースプログラムを目的プログラムに翻訳

問20

参考 インタプリタは同時通訳、コンパイラは翻訳者の役割だね。

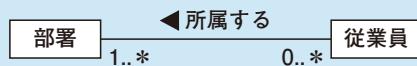


○ 解答

問18 イ 問19 ウ
問20 ア

問 **21**

UML を用いて表した図の概念データモデルの解釈として、適切なものはどれか。



- ア 従業員の総数と部署の総数は一致する。
- イ 従業員は、同時に複数の部署に所属してもよい。
- ウ 所属する従業員がいない部署の存在は許されない。
- エ どの部署にも所属しない従業員が存在してもよい。

問 **22**

“得点”表から、学生ごとに全科目の点数の平均を算出し、平均が 80 点以上の学生の学生番号とその平均点を求める。a に入れる適切な字句はどれか。ここで、実線の下線は主キーを表す。

得点 (学生番号, 科目, 点数)

[SQL 文]

```
SELECT 学生番号 , AVG( 点数 )
FROM 得点
GROUP BY a
```

- ア 科目 HAVING AVG(点数) >= 80
- イ 科目 WHERE 点数 >= 80
- ウ 学生番号 HAVING AVG(点数) >= 80
- エ 学生番号 WHERE 点数 >= 80

問 **23**

関係モデルにおいて、関係から特定の属性だけを取り出す演算はどれか。

- ア 結合 (join)
- イ 射影 (projection)
- ウ 選択 (selection)
- エ 和 (union)

問21 UML の概念データモデル

■■■ 20%

図は、従業員が部署に「所属する」関係であることを表しています。部署の多重度「1.. *」は、1人の従業員が必ず1つ以上の部署に所属することを示します。また従業員の多重度「0.. *」は、1つの部署に0人以上の従業員が所属することを示します。

- × **A** 従業員と部署の対応は1対1ではないので、総数は一致するとは限りません。
- **イ** 正解です。従業員は「1つ以上」の部署に所属するので、同時に複数の部署に所属できます。
- × **ウ** 1つの部署には「0人以上」の従業員が所属するので、所属する従業員がいない部署の存在も許されます。
- × **エ** 従業員は「1つ以上」の部署に所属するので、最低でも1つの部署に所属します。

問22 SQL文

■■■ 50%

平均は学生ごとに算出するので、GROUP BY句に学生を識別する学生番号を指定します。

GROUP BY 学生番号 ← 学生ごとにグループ化

次に、平均点が80点以上の学生のみ表示するために、HAVING句を使って、平均点が80点以上という条件を指定します。

HAVING AVG(点数) >= 80 ← 平均80点以上を取り出す

以上から、SQL文は次のようにになります。正解は**ウ**です。

```
SELECT 学生番号, AVG(点数)
FROM 得点
GROUP BY 学生番号 HAVING AVG(点数) >= 80 ← ウ
          空欄 a
```

問23 関係演算 キホン!

■■■ 50%

関係モデルにおいて、行・列からなる二次元の表を**関係**といい、行を**タブル**、列を**属性**といいます。「関係から特定の属性を取り出す」とは、表から特定の列を取り出す操作です。関係演算では、このような操作を**射影**といいます。

- × **ア** **結合**は、複数の表を共通する列をキーにして連結し、1つの表にまとめる関係演算です。
- **イ** 正解です。
- × **ウ** **選択**は、表から特定の行を取り出す関係演算です。
- × **エ** 和は、2つの表の**和集合**を求めます。

合格のカギ

覚えよう! 問21

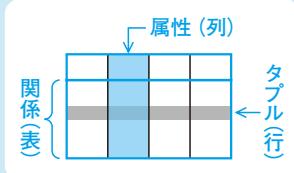
多重度の種類

0..1	ゼロまたは1
1	1つだけ
0.. *	0以上
1.. *	1以上

UML 問21

Unified Modeling Language : オブジェクト指向開発（設計）で利用する様々な種類の図式を規格化したもの。データモデルには、UMLのクラス図を利用する。

関係モデル 問23



覚えよう! 問23

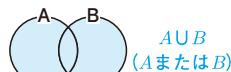
関係演算といえば

- 射影：特定の列を取り出す
- 選択：特定の行を取り出す
- 結合：複数の表を共通する列をキーにして結合する

覚えよう! 問23

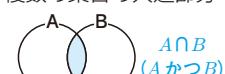
和集合といえば

- 複数の集合を合わせた集合



積集合といえば

- 複数の集合の共通部分



解答

- | | | | |
|-----|---|-----|---|
| 問21 | イ | 問22 | ウ |
| 問23 | イ | | |

**問 24**

10M ビット／秒の回線で接続された端末間で、平均 1M バイトのファイルを、10 秒ごとに転送するときの回線利用率は何%か。ここで、ファイル転送時には、転送量の 20%が制御情報として付加されるものとし、1M ビット = 10^6 ビットとする。

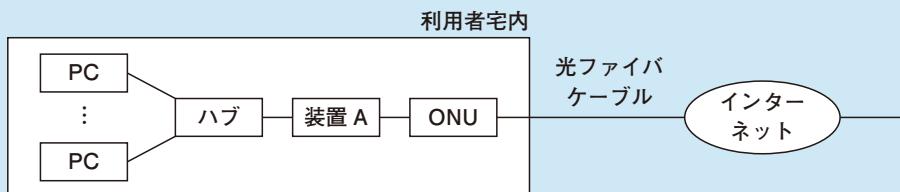
ア 1.2**イ** 6.4**ウ** 8.0**エ** 9.6**問 25**

メディアコンバータ、リピータハブ、レイヤ2スイッチ、レイヤ3スイッチのうち、レイヤ3スイッチだけがもつ機能はどれか。

- ア** データリンク層において、宛先アドレスに従って適切な LAN ポートにパケットを中継する機能
- イ** ネットワーク層において、宛先アドレスに従って適切な LAN ポートにパケットを中継する機能
- ウ** 物理層において、異なる伝送媒体を接続し、信号を相互に変換する機能
- エ** 物理層において、入力信号を全ての LAN ポートに対して中継する機能

**問 26**

LAN に接続されている複数の PC をインターネットに接続するシステムがあり、装置 A の WAN 側インターフェースには 1 個のグローバル IP アドレスが割り当てられている。この 1 個のグローバル IP アドレスを使って複数の PC がインターネットを利用するのに必要な装置 A の機能はどれか。



- ア** DHCP
- イ** NAPT (IP マスカレード)
- ウ** PPPoE
- エ** パケットフィルタリング

問24 回線利用率の計算 キホン!

■■■ 20%

10秒間に転送するデータ量は、ファイルが1Mバイトと、制御情報が1Mバイト×20%で、合わせて1.2Mバイトです。1バイト=8ビットなので、 $1.2\text{Mバイト} = 1.2 \times 8\text{Mビット} = 9.6\text{Mビット}$ となります。

一方、回線速度は10Mビット／秒なので、10秒間に $10\text{Mビット} \times 10\text{秒} = 100\text{Mビット}$ を転送できます。したがって回線利用率は、

$$\frac{9.6\text{Mビット}}{100\text{Mビット}} \times 100 = 9.6\%$$

となります。正解は**工**です。

問25 レイヤ3スイッチの機能

■■■ 80%

LAN間接続装置は、ネットワーク相互の通信をOSI基本参照モデルのどの階層で中継するかによって機能が異なります。メディアコンバータとリピータハブは物理層、レイヤ2スイッチはデータリンク層、レイヤ3スイッチはネットワーク層で通信を中継します。

- ア** レイヤ2スイッチの機能です。
- イ** 正解です。
- ウ** メディアコンバータの機能です。
- エ** リピータハブの機能です。

問26 LANとインターネットとの接続 キホン!

■■■ 30%

LANに接続した各端末に割り当てられているプライベートIPアドレスと、インターネットで利用するためのグローバルIPアドレスとを相互に変換する機能をNAT(Network Address Translation)といいます。特に、ポート番号と組み合わせることで、グローバルIPアドレス1個と、複数の端末のプライベートIPアドレスとを変換する機能をNAPT(IPマスクレード)といいます。

なお、図の装置Aはルータ、ONUは光回線終端装置です。

- ア** DHCP(Dynamic Host Configuration Protocol)は、ネットワークに接続した端末にIPアドレスを自動的に割り当てるプロトコルです。
- イ** 正解です。
- ウ** PPPoE(PPP over Ethernet)は、イーサネット上でPPPの機能を利用するためのプロトコルで、光回線などでインターネット接続プロバイダに接続する際のユーザ認証に利用されています。
- エ** パケットフィルタリングは、通過するパケットのIPアドレスやポート番号に応じて、通信を許可したり禁止したりする機能です。



覚えよう!

問25

OSI基本参照モデルとLAN間接続装置

OSI基本 参照モデル	LAN間 接続装置
第4～7層 その他	ゲートウェイ
第3層 ネットワーク層	ルータ
第2層 データリンク層	ブリッジ、 レイヤ2スイッチ
第1層 物理層	リピータハブ、 リピータ

問25

参考 「レイヤ2」「レイヤ3」は、OSI基本参照モデルの第2層、第3層という意味だよ。



④ メディアコンバータ [問25](#)
光ファイバとLANケーブルといった異なる伝送媒体同士を接続し、信号を相互に変換する装置。OSI基本参照モデルの物理層で信号を中継する。ONU(光回線終端装置)など。

⑤ NAPT(IPマスクレード)

問26

LAN内部のプライベートIPアドレスを、インターネット上のグローバルIPアドレスに変換する機能。ポート番号を組み合わせることで、1個のグローバルIPアドレスでも同時に複数の端末がインターネットを利用できる。

○ 解答 ○

問24 **工** 問25 **イ**
問26 **イ**



問 27

クライアント A がポート番号 8080 の HTTP プロキシサーバ B を経由してポート番号 80 の Web サーバ C にアクセスしているとき、宛先ポート番号が常に 8080 になる TCP パケットはどれか。

- ア A から B への HTTP 要求及び C から B への HTTP 応答
- イ A から B への HTTP 要求だけ
- ウ B から A への HTTP 応答だけ
- エ B から C への HTTP 要求及び C から B への HTTP 応答



問 28

攻撃者が用意したサーバ X の IP アドレスが、A 社 Web サーバの FQDN に対応する IP アドレスとして、B 社 DNS キャッシュサーバに記憶された。これによって、意図せずサーバ X に誘導されてしまう利用者はどれか。ここで、A 社、B 社の各従業員は自社の DNS キャッシュサーバを利用して名前解決を行う。

- ア A 社 Web サーバにアクセスしようとする A 社従業員
- イ A 社 Web サーバにアクセスしようとする B 社従業員
- ウ B 社 Web サーバにアクセスしようとする A 社従業員
- エ B 社 Web サーバにアクセスしようとする B 社従業員



問 29

マルウェアの動的解析に該当するものはどれか。

- ア 検体のハッシュ値を計算し、オンラインデータベースに登録された既知のマルウェアのハッシュ値のリストと照合してマルウェアを特定する。
- イ 検体をサンドボックス上で実行し、その動作や外部との通信を観測する。
- ウ 検体をネットワーク上の通信データから抽出し、さらに、逆コンパイルして取得したコードから検体の機能を調べる。
- エ ハードディスク内のファイルの拡張子とファイルヘッダの内容を基に、拡張子が偽装された不正なプログラムファイルを検出する。

解説

問27 プロキシサーバ

クライアント A が、HTTP プロキシサーバ B を経由して Web サーバ C にアクセスする場合の TCP パケットは、次の順でやり取りされます。

①A から B への HTTP 要求：宛先がプロキシサーバなので、宛先ポート番号は 8080 になります。また、送信元ポート番号は、クライアントの OS が自動的に割り当てた番号になります。

②B から C への HTTP 要求：宛先が Web サーバなので、宛先ポート番号は 80 になります。このとき、送信元ポート番号には、プロキシサーバが Web サーバとやり取りするために使うポート番号が使われます。



プロキシサーバ

問27

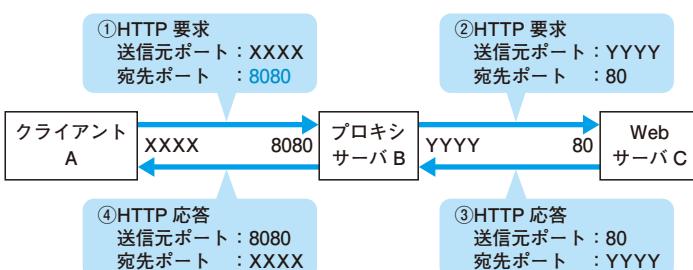
クライアントが外部サーバとやり取りする場合に、クライアントを代行して外部サーバとやり取りするサーバ。クライアントのセキュリティ向上や、外部とのアクセス管理などの目的で利用される。

合格のカギ

ポート番号

問27

インターネットのTCPやUDPプロトコルで、宛先や送信元アプリケーションの識別に使われる番号。代表的なアプリケーションについては標準のポート番号が決まっており、ウェルノンポート番号と呼ばれる。Webサーバのウェルノンポート番号は80。プロキシサーバは8080。



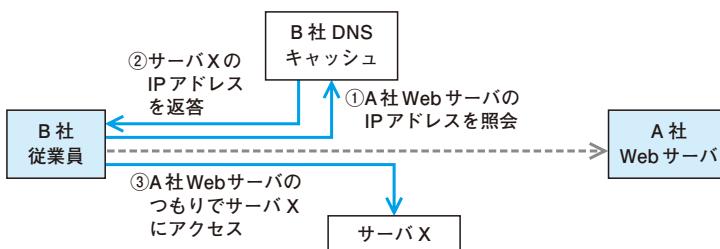
以上から、宛先ポート番号が常に8080になるTCPパケットは、①のAからBへのHTTP要求のみとなります。正解はイです。

問28 DNSキャッシュポイズニング攻撃

30%

B社の従業員が、WebブラウザにA社WebサーバのURLを入力すると、ブラウザはこのURLに対応するWebサーバのIPアドレスを、B社DNSキャッシュサーバに問い合わせます。するとB社DNSキャッシュサーバはA社Webサーバのものとして、サーバXのIPアドレスを返します。これにより、B社従業員はサーバXに誘導されます。このような攻撃をDNSキャッシュポイズニング攻撃といいます。

以上から、意図せずにサーバXに誘導されてしまう利用者は、A社WebサーバにアクセスしようとするB社従業員です。正解はイです。



問29 マルウェアの動的解析

20%

マルウェアの動的解析とは、検体（解析対象となるマルウェアのプログラムファイルなど）を実際に実行し、その動作や外部との通信を観測する解析手法です。実害を防ぐため、動的解析は**サンドボックス**と呼ばれる保護された環境下で行われます。

- ア ハッシュ値を利用した**パターンマッチング法**の説明です。
- イ 正解です。
- ウ 静的解析の説明です。
- エ 偽装された拡張子を検知してマルウェアを検出する手法です。マルウェアを動作させるわけではないので、動的解析ではありません。

DNSキャッシュポイズニング攻撃

問28

DNSキャッシュサーバに偽のドメイン情報を注入し、サーバの利用者を誤ったサイトに誘導する攻撃。

FQDN

問28

Fully Qualified Domain Name: 完全修飾ドメイン名。ドメイン名だけでなく、サブドメイン名やホスト名を省略せずに指定したもの。

マルウェア

問29

コンピュータウイルスやスパイウェア、ワームなどの悪質なプログラムの総称。

サンドボックス

問29

プログラムがシステムに悪影響を与えないように、隔離して構築された仮想環境のこと。

○ 解答 ○

問27 イ	問28 イ
問29 イ	

問 30 WPA3 はどれか。

- ア HTTP 通信の暗号化規格
- イ TCP/IP 通信の暗号化規格
- ウ Web サーバで使用するデジタル証明書の規格
- エ 無線 LAN のセキュリティ規格

問 31 メッセージに RSA 方式のデジタル署名を付与して 2 者間で送受信する。そのときのデジタル署名の検証鍵と使用方法はどれか。

- ア 受信者の公開鍵であり、送信者がメッセージダイジェストからデジタル署名を作成する際に使用する。
- イ 受信者の秘密鍵であり、受信者がデジタル署名からメッセージダイジェストを算出する際に使用する。
- ウ 送信者の公開鍵であり、受信者がデジタル署名からメッセージダイジェストを算出する際に使用する。
- エ 送信者の秘密鍵であり、送信者がメッセージダイジェストからデジタル署名を作成する際に使用する。

問 32 情報セキュリティにおいてバックドアに該当するものはどれか。

- ア アクセスする際にパスワード認証などの正規の手続が必要な Web サイトに、当該手続を経ないでアクセス可能な URL
- イ インターネットに公開されているサーバの TCP ポートの中からアクティブになっているポートを探して、稼働中のサービスを特定するためのツール
- ウ ネットワーク上の通信パケットを取得して通信内容を見るために設けられたスイッチの LAN ポート
- エ プログラムが確保するメモリ領域に、領域の大きさを超える長さの文字列を入力してあふれさせ、ダウンさせる攻撃



問30 WPA3

■■■ 20%

WPA (Wi-Fi Protected Access) は、無線 LAN のセキュリティ規格で、通信の暗号化やアクセスポイントへのユーザ認証方式などを規定したものです。WPA3は、現在幅広く使われている WPA2 の次世代規格です。

- × ア HTTP 通信の暗号化規格は、**HTTPS** です。
- × イ TCP/IP 通信の暗号化規格は、**IPsec** です。
- × ウ Web サーバで使用するデジタル証明書の規格は、**TLS** です。
- エ 正解です。

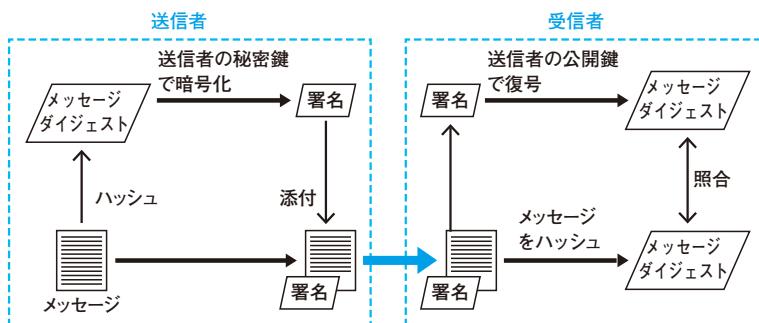
問31 デジタル署名の検証 キポン!

■■■ 50%

デジタル署名は、メッセージから生成したメッセージダイジェスト（ハッシュ値）を、送信者の秘密鍵によって暗号化したものです。メッセージの受信者は、デジタル署名を**送信者の公開鍵**によって復号し、メッセージダイジェストを得ます。復号に成功するのは正しい公開鍵が使われたときなので、これによりデジタル署名が本人のものであることが検証できます。

また、受け取ったメッセージ本体からメッセージダイジェストを生成し、これがデジタル署名から復号した内容と一致すれば、メッセージ本体が改ざんされていないことが確認できます。

以上から、デジタル署名の検証鍵は**送信者の公開鍵**で、受信者がデジタル署名からメッセージダイジェストを得る際に使用します。正解は **ウ** です。



問32 バックドア

■■■ 20%

バックドアは「裏口」という意味ですが、情報セキュリティ用語としては正规の手続きを経ない侵入経路を指します。たとえば、通常はユーザ認証しないとアクセスできない Web サイトに、ユーザ認証なしでアクセスできる URL などがバックドアに該当します。

- ア 正解です。
- × イ ポートスキャナの説明です。
- × ウ ネットワーク上の通信パケットを監視するツールを、**LAN アナライザ**などといいます。
- × エ バッファオーバフロー攻撃の説明です。

合格のカギ

IPsec

問30

IP パケットを暗号化して安全な通信を行うためのプロトコル。IPv4, IPv6 のどちらでも利用できるが、IPv6 には標準で IPsec の機能が組み込まれている。

TLS

問30

Transport Layer Security : インターネット上の利用者認証や暗号化、改ざんの検出など、安全な通信を実現するためのプロトコル。

覚えよう！

問31

デジタル署名といえば

- 送信者：メッセージのハッシュ値を送信者の秘密鍵で暗号化
- 受信者：送信者の公開鍵で署名を復号

問31

対策 メッセージから生成したハッシュ値のことをメッセージダイジェストというよ。



ポートスキャナ

問32

ポート番号を外部から順に指定してサーバにアクセスし、利用可能になっているサービスを探すソフトウェアのこと。不要なサービスが利用可能になると、不正アクセスされるおそれがあるため、ポートスキャナで不要なサービスが稼働していないことを確認する。

○ 解答 ○

- | | |
|-------|-------|
| 問30 エ | 問31 ウ |
| 問32 ア | |

問 33

検索サイトの検索結果の上位に悪意のあるサイトが表示されるように細工する攻撃の名称はどれか。

- ア DNS キャッシュポイズニング
ウ クロスサイトスクリプティング

- イ SEO ポイズニング
エ ソーシャルエンジニアリング

問 34

1台のファイアウォールによって、外部セグメント、DMZ、内部セグメントの三つのセグメントに分割されたネットワークがあり、このネットワークにおいて、Web サーバと、重要なデータをもつデータベースサーバから成るシステムを使って、利用者向けの Web サービスをインターネットに公開する。インターネットからの不正アクセスから重要なデータを保護するためのサーバの設置方法のうち、最も適切なものはどれか。ここで、Web サーバでは、データベースサーバのフロントエンド処理を行い、ファイアウォールでは、外部セグメントと DMZとの間、及び DMZ と内部セグメントとの間の通信は特定のプロトコルだけを許可し、外部セグメントと内部セグメントとの間の直接の通信は許可しないものとする。

- ア Web サーバとデータベースサーバを DMZ に設置する。
イ Web サーバとデータベースサーバを内部セグメントに設置する。
ウ Web サーバを DMZ に、データベースサーバを内部セグメントに設置する。
エ Web サーバを外部セグメントに、データベースサーバを DMZ に設置する。

問 35

パスワードリスト攻撃の手口に該当するものはどれか。

- ア 辞書にある単語をパスワードに設定している利用者がいる状況に着目して、攻撃対象とする利用者 ID を一つ定め、辞書にある単語やその組合せをパスワードとして、ログインを試行する。
イ パスワードの文字数の上限が小さい Web サイトに対して、攻撃対象とする利用者 ID を一つ定め、文字を組み合わせたパスワードを総当たりして、ログインを試行する。
ウ 複数サイトで同一の利用者 ID とパスワードを使っている利用者がいる状況に着目して、不正に取得した他サイトの利用者 ID とパスワードの一覧表を用いて、ログインを試行する。
エ よく用いられるパスワードを一つ定め、文字を組み合わせた利用者 ID を総当たりして、ログインを試行する。

問33 サイバー攻撃の手法 キホン!

■ 20%



特定のサイトが、検索サイトの検索結果の上位に表示されるようにサイトの内容を調整することを **SEO**（検索エンジン最適化）といいます。SEO の手法を使って、悪意のあるサイトを検索結果の上位に表示させる攻撃手法を **SEOポイズニング**といいます。

- × **A DNSキャッシュポイズニング**は、DNSサーバのDNSキャッシュに登録されているドメインの情報を改ざんし、そのDNSサーバの利用者を特定のWebサイトに誘導する攻撃です。
- **イ** 正解です。
- × **ウ クロスサイトスクリプティング**は、入力されたデータをそのまま画面に表示してしまうWebサイトに利用者を誘導し、利用者のWebブラウザで悪意のあるスクリプトを実行させる攻撃です。
- × **エ ソーシャルエンジニアリング**は、人の心理の隙や油断について機密情報を入手する攻撃手法です。

問34 DMZ キホン!

■ 20%

DMZは、外部ネットワークと直接やり取りする必要があるサーバ群を、内部セグメントから切り離して設置するセグメントです。

- × **A** DMZは部分的にしろ外部からのアクセスが許可されるので、重要なデータを含むデータベースサーバを置くべきではありません。
- × **イ** Webサーバを内部セグメントに設置すると、外部からのアクセスを受けられません。
- **ウ** 正解です。
- × **エ** Webサーバを外部セグメントに設置すると、ファイアウォールの保護が受けられません。

問35 パスワードリスト攻撃

■ 20%

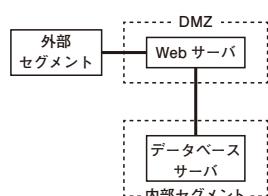
利用者にとって、パスワードをいくつも覚えるのは負担なので、1つのパスワードを複数のサイトで使いまわす利用者は多くいます。**パスワードリスト攻撃**はこの状況を利用して、あるサイトで不正に取得した利用者IDとパスワードのリストを用いて、別のサイトへ不正にログインしようとする攻撃です。

- × **ア 辞書攻撃**の手口です。
- × **イ ブルートフォース攻撃**（総当たり攻撃）の手口です。
- **ウ** 正解です。
- × **エ リバースブルートフォース攻撃**と呼ばれる手口です。

問34 DMZ

問34

DeMilitarized Zone : 非武装地帯という意味。社内 LAN とインターネットとの間に第三の領域を設け、社外に公開するサーバを社内 LAN から隔離したもの。外部から不正アクセスを受けても、社内 LAN に被害が及ばないようにする。



問35

参考 パスワードリスト攻撃への対策としては、パスワードの使いまわしをやめること。同じパスワードでも、サイト名を先頭や末尾に追加するだけで違うパスワードになるよ。



○ 解答 ○

- | | |
|-------|-------|
| 問33 イ | 問34 ウ |
| 問35 ウ | |

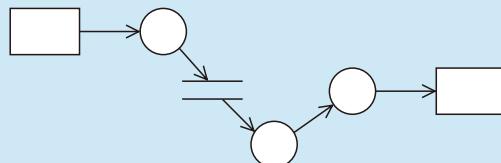
問 36 CAPTCHA の目的はどれか。

- ア Web サイトなどにおいて、コンピュータではなく人間がアクセスしていることを確認する。
- イ 公開鍵暗号と共に鍵暗号を組み合わせて、メッセージを効率よく暗号化する。
- ウ 通信回線を流れるパケットをキャプチャして、パケットの内容の表示や解析、集計を行う。
- エ 電子政府推奨暗号の安全性を評価し、暗号技術の適切な実装法、運用法を調査、検討する。

問 37 電子メールをドメイン A の送信者がドメイン B の宛先に送信するとき、送信者をドメイン A のメールサーバで認証するためのものはどれか。

- ア APOP
- イ POP3S
- ウ S/MIME
- エ SMTP-AUTH

問 38 図は、構造化分析法で用いられる DFD の例である。図中の “○” が表しているものはどれか。



- ア アクティビティ
- イ データストア
- ウ データフロー
- エ プロセス

問 39 モジュール結合度が最も弱くなるものはどれか。

- ア 一つのモジュールで、できるだけ多くの機能を実現する。
- イ 二つのモジュール間で必要なデータ項目だけを引数として渡す。
- ウ 他のモジュールとデータ項目を共有するためにグローバルな領域を使用する。
- エ 他のモジュールを呼び出すときに、呼び出したモジュールの論理を制御するための引数を渡す。

解説

問36 CAPTCHA

CAPTCHA（キャプチャ）は、Web サイトの認証などで、コンピュータではなく人間が入力していることを確認するための技術です。ランダムに並べた文字や数字の画像を歪めたり、一部を隠したりして表示し、それを利用者に判

合格のカギ

20%

読してもらいます。変形された文字はプログラムでは判読が難しいので、プログラムによる自動入力を排除できます。

- ア 正解です。
- イ ハイブリッド暗号方式の説明です。
- ウ パケットキャプチャの説明です。
- エ CRYPTREC の説明です。

合格のカギ

CAPTCHA の例 問36

CRYPTREC 問36

Cryptography Research and Evaluation Committees : 電子政府推奨暗号の安全性を評価・監視し、暗号技術の適切な実装法・運用法を調査・検討するプロジェクト。

S/MIME 問37

MIME（電子メールでバイナリデータなどを送信するための仕様）を拡張して、暗号化やデジタル署名などのセキュリティ機能を追加した規格。

問37 送信者の認証

メールサーバで、電子メールの送信時に送信者を認証する方式は、SMTP-AUTHです。インターネットの電子メールでは、メール送信にはSMTP、メール受信にはPOP3またはIMAP4というプロトコルが使われます。SMTP-AUTHは、送信プロトコルであるSMTPでユーザ認証を行うための拡張機能です。

- ア APOPは、電子メールをPOP3で受信するときに、ユーザ名やパスワードを暗号化してやり取りする方式です。
- イ POP3Sは、メールサーバとの通信をTLS(SSL)によって暗号化した電子メールの受信プロトコルです。
- ウ S/MIME(Secure MIME)は、電子メールに暗号化やデジタル署名などのセキュリティ機能を追加した規格です。
- エ 正解です。

問38 DFD(データフローダイアグラム)

DFD(Data Flow Diagram)は、業務過程をデータの流れに着目して図式化したもので、各記号には次の意味があります。

→	データフロー	データの流れを表す
○	プロセス	データに対する処理を表す
==	データストア	データを読み書きするファイルを表す
□	外部(データ源泉/データ吸収)	データの受け渡しを行うシステム外部を表す

以上から、“○”はプロセスを表します。正解はエです。

問39 モジュール結合度

モジュール同士の関連性の度合いをモジュール結合度といいます。一般に、モジュール結合度が弱いほど個々のモジュールの独立性が高く、保守しやすくなる特徴があります。

- ア モジュールが分割されていません。
- イ データ項目を引数で渡すのは、結合度の最も弱いデータ結合です。
- ウ グローバルな領域でデータを共有するのは、共通結合または外部結合です。
- エ 呼び出したモジュールの動作を引数で制御するのは、制御結合です。

モジュール結合度 問39


 解答

- | | | | |
|-----|---|-----|---|
| 問36 | ア | 問37 | エ |
| 問38 | エ | 問39 | イ |

問 40 テストで使用するスタブ又はドライバの説明のうち、適切なものはどれか。

- ア スタブは、テスト対象モジュールからの戻り値の表示・印刷を行う。
- イ スタブは、テスト対象モジュールを呼び出すモジュールである。
- ウ ドライバは、テスト対象モジュールから呼び出されるモジュールである。
- エ ドライバは、引数を渡してテスト対象モジュールを呼び出す。

問 41 単一の入り口をもち、入力項目を用いた複数の判断を含むプログラムのテストケースを設計する。命令網羅と判定条件網羅の関係のうち、適切なものはどれか。

- ア 判定条件網羅を満足しても、命令網羅を満足しない場合がある。
- イ 判定条件網羅を満足するならば、命令網羅も満足する。
- ウ 命令網羅を満足しなくても、判定条件網羅を満足する場合がある。
- エ 命令網羅を満足するならば、判定条件網羅も満足する。

問 42 XP (eXtreme Programming)において、プラクティスとして提唱されているものはどれか。

- ア インスペクション
- イ 構造化設計
- ウ ペアプログラミング
- エ ユースケースの活用

解説

問40 スタブとドライバ キホン!

20%

テスト対象のモジュールから呼び出される仮の下位モジュールをスタブといいます。スタブは、モジュールを上位モジュールから順に結合していくトップダウンテストで必要となります。

一方、テスト対象のモジュールを呼び出す仮の上位モジュールをドライバと

合格のカギ

いいます。ドライバは、モジュールを下位モジュールから順に結合していくボトムアップテストで必要となります。

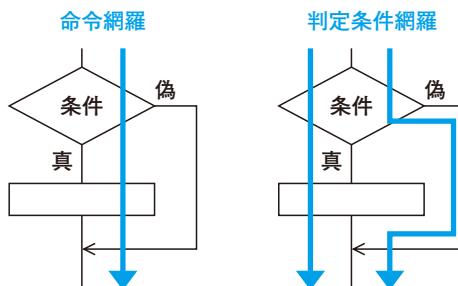
- ア テスト対象の下位モジュールからの戻り値の表示・印刷を行うのはドライバです。
- イ テスト対象モジュールを呼び出すモジュールはドライバです。
- ウ テスト対象モジュールから呼び出されるモジュールはスタブです。
- エ 正解です。ドライバは、引数を渡してテスト対象となる下位モジュールを呼び出します。

問41 命令網羅と判定条件網羅

■ 60%

命令網羅では、プログラムに含まれるすべての命令を、少なくとも1回は実行するようにテストケースを設計します。一方、判定条件網羅(分岐網羅)では、各判定条件の真偽を少なくとも1回実行するようにテストケースを設計します。

判定条件網羅ではすべての分岐を1回は通るので、判定条件網羅を満足すれば、命令網羅も満足します。



- ア 判定条件網羅を満足すれば、命令網羅も満足します。
- イ 正解です。
- ウ 命令網羅を満足しない場合は、通っていない分岐が必ずあるので、判定条件網羅も満足しません。
- エ 命令のない分岐がある場合、命令網羅を満足しても、判定条件網羅は満足しない場合があります。

問42 XP(eXtreme Programming)のプラクティス

■ 40%

XP(エクストリームプログラミング)は、少人数ですばやく開発をすすめていくソフトウェア開発手法です。XPではソースコードを重視し、開発途中の仕様変更にも柔軟に対応できるのが特徴です。

XPではプログラマが2人1組で作業するペアプログラミングの実践が提唱されています。ペアプログラミングでは、1人がコーディングをしている間、もう1人はコードをチェックするといった補助的な役割をして、コード品質の向上などの効果を狙います。正解はウです。

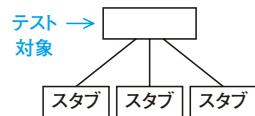


覚えよう!

問40

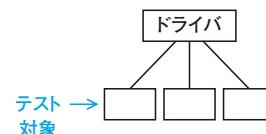
スタブといえば

- テスト対象モジュールから呼び出される仮の下位モジュール



ドライバといえば

- テスト対象モジュールを呼び出す仮の上位モジュール



覚えよう!

問41

命令網羅といえば

- すべての命令を少なくとも1回実行

判定条件網羅(分岐網羅)といえば

- 判定条件の真偽を少なくとも1回実行

条件網羅といえば

- 判定条件の個々の条件(A or Bの場合はAとB)の真偽を1回以上実行

判定条件／条件網羅といえば

- 判定条件(A or B)の真偽を少なくとも1回実行し、かつ、個々の条件(AとB)の真偽を1回以上実行

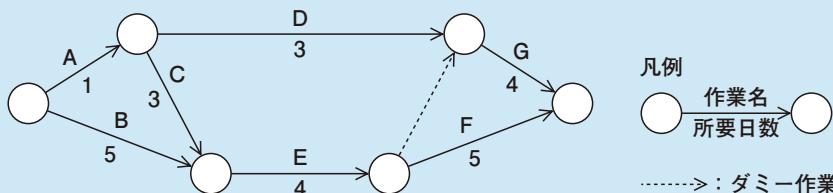
複数条件網羅といえば

- 個々の条件の真偽のすべての組合せをテストし、かつ、すべての命令を1回以上実行

○ 解答 ○

問40 エ 問41 イ
問42 ウ

問 43 あるプロジェクトの日程計画をアローダイアグラムで示す。クリティカルパスはどれか。



- ア A, C, E, F イ A, D, G ウ B, E, F エ B, E, G

問 44 ソフトウェア開発の見積方法の一つであるファンクションポイント法の説明として、適切なものはどれか。

- ア 開発規模が分かっていることを前提として、工数と工期を見積もる方法である。ビジネス分野に限らず、全分野に適用可能である。
イ 過去に経験した類似のソフトウェアについてのデータを基にして、ソフトウェアの相違点を調べ、同じ部分については過去のデータを使い、異なった部分は経験に基づいて、規模と工数を見積もる方法である。
ウ ソフトウェアの機能を入出力データ数やファイル数などによって定量的に計測し、複雑さによる調整を行って、ソフトウェア規模を見積もる方法である。
エ 単位作業項目に適用する作業量の基準値を決めておき、作業項目を単位作業項目まで分解し、基準値を適用して算出した作業量の積算で全体の作業量を見積もる方法である。

問 45 サービスマネジメントシステムにPDCA方法論を適用するとき、Actに該当するものはどれか。

- ア サービスの設計、移行、提供及び改善のためにサービスマネジメントシステムを導入し、運用する。
イ サービスマネジメントシステム及びサービスのパフォーマンスを継続的に改善するための処置を実施する。
ウ サービスマネジメントシステムを確立し、文書化し、合意する。
エ 方針、目的、計画及びサービスの要求事項について、サービスマネジメントシステム及びサービスを監視、測定及びレビューし、それらの結果を報告する。

問43 クリティカルパス キホン!

■■■ 80%

開始から終了までの各作業のうち、時間的余裕のない一連の作業経路を**クリティカルパス**といいます。クリティカルパス上にある各作業が1日でも遅れるとき、プロジェクト全体の遅れにつながります。

アローダイアグラムでは、開始から終了に至る複数の作業経路のうち、合計の所要日数が最も多い経路がクリティカルパスになります。解答群は、いずれも開始から終了までの作業経路なので、このうち所要日数の合計が最も多いものが正解です。

- × **ア** A, C, E, F : $1 + 3 + 4 + 5 = 13$ 日
- × **イ** A, D, G : $1 + 3 + 4 = 8$ 日
- **ウ** B, E, F : $5 + 4 + 5 = 14$ 日 ← **クリティカルパス**
- × **エ** B, E, G : $5 + 4 + 4 = 13$ 日



アローダイアグラム **問43**
プロジェクトの開始から終了までの作業を矢印の線で表し、作業と作業の結合点を○で表した図。ある結合点から出る作業は、その結合点に入るすべての作業の完了後に開始する。開始前に完了しなければならない作業が別の結合点に入る場合は、その結合点からダミー作業（所要日数0の作業）を出す。

問44 ファンクションポイント法 キホン!

■■■ 60%

ファンクションポイント法は、システムの機能を入出力データ数やファイル数、画面数などによって定量的に計測し、難易度などを考慮して点数化することで、システム規模を見積もる手法です。

- × **ア** 開発規模や予想行数などから、工数や工期を計算する見積もり手法には、**COCOMO**（ココモ）があります。
- × **イ** **類推法**の説明です。
- **ウ** 正解です。
- × **エ** **WBS法**または**標準タスク法**の説明です。

COCOMO **問44**
ベームが提唱した見積もり手法で、プログラムステップ数などで表される開発規模を基に、開発工数や開発期間を算出するもの。

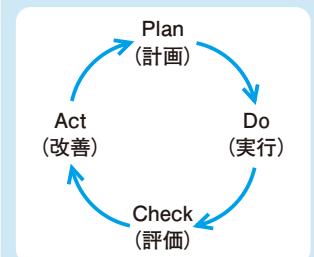
問45 PDCAサイクル

■■■ 20%

PDCAとは、Plan(計画)→Do(実行)→Check(評価)→Act(改善)という4つの段階を繰り返しながら、業務を継続的に管理・改善していくことです。

- × **ア** サービスマネジメントシステムを導入・運用するので、**Do(実行)**に該当します。
- **イ** 改善するための処置を実施するので、**Act(改善)**に該当します。
- × **ウ** サービスマネジメントシステムを確立・文書化するので、**Plan(計画)**に該当します。
- × **エ** サービスやマネジメントシステム自体を監視・測定・レビューするので、**Check(評価)**に該当します。

PDCAサイクル **問45**



○ 解答 ○

問43 **ウ** 問44 **ウ**
問45 **イ**

 **問 46** システムの移行計画に関する記述のうち、適切なものはどれか。

- ア** 移行計画書には、移行作業が失敗した場合に旧システムに戻す際の判断基準が必要である。
- イ** 移行するデータ量が多いほど、切替え直前に一括してデータの移行作業を実施すべきである。
- ウ** 新旧両システムで環境の一部を共有することによって、移行の確認が容易になる。
- エ** 新旧両システムを並行運用することによって、移行に必要な費用が低減できる。

 **問 47** システムテストの監査におけるチェックポイントのうち、最も適切なものはどれか。

- ア** テストケースが網羅的に想定されていること
- イ** テスト計画は利用者側の責任者だけで承認されていること
- ウ** テストは実際に業務が行われている環境で実施されていること
- エ** テストは利用者側の担当者だけで行われていること

 **問 48** 情報システム部が開発して経理部が運用している会計システムの運用状況を、経営者からの指示で監査することになった。この場合におけるシステム監査人についての記述のうち、最も適切なものはどれか。

- ア** 会計システムは企業会計に関する各種基準に準拠すべきなので、システム監査人を公認会計士とする。
- イ** 会計システムは機密性の高い情報を扱うので、システム監査人は経理部長直属とする。
- ウ** システム監査を効率的に行うために、システム監査人は情報システム部長直属とする。
- エ** 独立性を担保するために、システム監査人は情報システム部にも経理部にも所属しない者とする。



問46 システムの移行計画

■ 20%

現行の情報システムを、新しい情報システムに切り替えることを**移行**といいます。移行作業中にトラブルが起こった場合は、旧システムに戻すことも考えなければなりません。旧システムに戻すかどうかの判断基準は、移行計画の中で前もって決めておくべきです。

- ア 正解です。
- × イ データ量が多いほど移行作業に時間がかかるので、直前に一括して移行すると間に合わない可能性があります。更新頻度の少ないものは前もって移行しておくなど、段階的にすすめるべきです。
- × ウ 新旧両システムで環境の一部を共有した場合、問題の原因が新システムにあるのか、共有環境にあるのか確認が必要になるため、移行の確認が困難になる場合があります。
- × エ 新システムの移行後も旧システムを並行運用する場合は、両システムを連携させるための追加費用が必要となり、移行費用は高くなります。

問47 システムテストの監査

■ 20%

システムテストは、完成したシステムが要求事項を満たしているか、性能や機能に不足がないかなどを、総合的に検証するテストです。

- ア 正解です。要求事項を満たしているかどうかを検証するために、テストケースはすべての要求事項を網羅するように設計します。
- × イ システムテストは開発部門が行うテストなので、テスト計画は開発部門の責任者の承認を得ます。
- × ウ システムテストは、本番環境とは隔離された環境で実施します。
- × エ システムテストには、利用者側の担当者も参加すべきです。ただし、利用者側の担当者だけでは実施できません。

問48 システム監査人 キポン!

■ 40%

システム監査人は、独立・公正な立場で監査を実施できるように、被監査主体（監査される側）と身分上密接な利害関係をもたないようにしなければなりません。これをシステム監査人の**外観上の独立性**といいます。

- × ア 公認会計士は、会計が適切に行われているかどうかを監査します。会計システムの運用状況の監査は、公認会計士である必要はありません。
- × イ 会計システムは経理部が運用しているので、システム監査人を経理部長直属にすると利害関係が生じ、独立性が担保できません。
- × ウ 会計システムは情報システム部が開発しているので、システム監査人を情報システム部長直属にすると独立性が担保できません。
- エ 正解です。

監査人の独立性 問48
経済産業省が策定したシステム監査基準には、「システム監査人は、システム監査を実施するために、監査対象から独立していかなければならない。監査の目的によっては、被監査主体と身分上、密接な利害関係を有することがあってはならない。」と規定されている（外観上の独立性）。

○ 解答 ○

問46 ア 問47 ア
問48 エ

問 49 情報化投資において、リスクや投資価値の類似性でカテゴリ分けし、最適な資源配分を行う際に用いる手法はどれか。

- ア** 3C 分析
- イ** IT ポートフォリオ
- ウ** エンタープライズアーキテクチャ
- エ** ベンチマーкиング

問 50 自社の経営課題である人手不足の解消などを目標とした業務革新を進めるために活用する、RPA の事例はどれか。

- ア** 業務システムなどのデータ入力、照合のような標準化された定型作業を、事務職員の代わりにソフトウェアで自動的に処理する。
- イ** 製造ラインで部品の組立てに従事していた作業員の代わりに組立作業用ロボットを配置する。
- ウ** 人が接客して販売を行っていた店舗を、IC タグ、画像解析のためのカメラ、電子決済システムによる無人店舗に置き換える。
- エ** フォークリフトなどを用いて人の操作で保管商品を搬入・搬出していたものを、コンピュータ制御で無人化した自動倉庫システムに置き換える。

問 51 非機能要件の定義で行う作業はどれか。

- ア** 業務を構成する機能間の情報（データ）の流れを明確にする。
- イ** システム開発で用いるプログラム言語に合わせた開発基準、標準の技術要件を作成する。
- ウ** システム機能として実現する範囲を定義する。
- エ** 他システムとの情報授受などのインターフェースを明確にする。

問 52 バランススコアカードの内部ビジネスプロセスの視点における戦略目標と業績評価指標の例はどれか。

- ア** 持続的成長が目標であるので、受注残を指標とする。
- イ** 主要顧客との継続的な関係構築が目標であるので、クレーム件数を指標とする。
- ウ** 製品開発力の向上が目標であるので、製品開発領域の研修受講時間を指標とする。
- エ** 製品の製造の生産性向上が目標であるので、製造期間短縮日数を指標とする。

問49 情報化投資の手法 20%

ハイリスク／ハイリターンの投資と、ローリスク／ローリターンの投資を組み合わせて、全体としてリスクとリターンのバランスをとる資産運用の手法を、**ポートフォリオ**といいます。この手法を応用して、情報化投資への資源配分を最適化する手法を**ITポートフォリオ**といいます。

- A 3C分析**は、市場 (Customer), 競合 (Competitor), 自社 (Company) の3つの視点から現状を分析する経営手法です。
- イ** 正解です。
- ウ エンタープライズアーキテクチャ (EA)**は、組織をビジネス、データ、アプリケーション、テクノロジの4つの体系（アーキテクチャ）ごとに分析し、全体最適化を図る手法です。
- エ ベンチマー킹**は、自社の製品やサービスを、優良な競合企業と比較して、経営戦略の立案に役立てる経営手法です。

問50 RPA 20%

RPA (Robotic Process Automation) は、パソコンのアプリケーション画面などを、人間と同じように操作するソフトウェアロボットです。データ入力や照合といった定型作業を、人間の代わりにRPAが行うことで、企業にとっては人手不足解消や業務の効率化が期待できます。正解は**ア**です。

問51 非機能要件の定義  30%

情報システムの開発で行う要件定義のうち、業務に必要な機能を明らかにしたものをお**機能要件**といいます。これに対し、システムの性能や使いやすさ、開発方法、運用費用など、機能以外の様々な要素を**非機能要件**といいます。

- ア** 機能間のデータの流れは機能要件です。
- イ** 開発基準などは非機能要件です。
- ウ** システム機能として実現する範囲は機能要件です。
- エ** 他システムとのインターフェースは機能要件です。

問52 バランススコアカード 30%

バランススコアカードは、自社の経営戦略を①財務、②顧客、③内部ビジネスプロセス、④学習と成長の4つの視点から分析し、視点ごとに具体的な戦略目標や施策を策定する経営管理手法です。このうち内部ビジネスプロセスの視点では、財務や顧客の視点における目標を達成するために、業務プロセスや製造工程を改善していくことが目標となります。

- ア** 「財務」視点の目標・指標例です。
- イ** 「顧客」視点の目標・指標例です。
- ウ** 「学習と成長」視点の目標・指標例です。
- エ** 「内部ビジネスプロセス」の目標・指標例です。

 合格のカギ 覚えよう！ 問49

エンタープライズアーキテクチャといえば

- **ビジネスアーキテクチャ**：業務機能の構成
- **データアーキテクチャ**：業務機能に使われる情報の構成
- **アプリケーションアーキテクチャ**：業務機能と情報の流れをまとめたサービス群の構成
- **テクノロジアーキテクチャ**：各サービスを実現するための技術の構成

 覚えよう！ 問52

バランススコアカードといえば

- 財務
- 顧客
- 内部ビジネスプロセス
- 学習と成長

○ 解答 ○

- | | | | |
|-----|---|-----|---|
| 問49 | イ | 問50 | ア |
| 問51 | イ | 問52 | エ |



問 53

技術経営におけるプロダクトイノベーションの説明として、適切なものはどれか。

- ア 新たな商品や他社との差別化ができる商品を開発すること
- イ 技術開発の成果によって事業利益を獲得すること
- ウ 技術を核とするビジネスを戦略的にマネジメントすること
- エ 業務プロセスにおいて革新的な改革をすること



問 54

デジタルディバイドを説明したものはどれか。

- ア PCなどの情報通信機器の利用方法が分からなかったり、情報通信機器を所有していないなどして、情報の入手が困難な人々のことである。
- イ 高齢者や障害者の情報通信の利用面での困難が、社会的又は経済的な格差につながるないように、誰もが情報通信を利活用できるように整備された環境のことである。
- ウ 情報通信機器やソフトウェア、情報サービスなどを、高齢者・障害者を含む全ての人が利用可能であるか、利用しやすくなっているかの度合いのことである。
- エ 情報リテラシの有無やITの利用環境の相違などによって生じる、社会的又は経済的な格差のことである。



問 55

“かんばん方式”を説明したものはどれか。

- ア 各作業の効率を向上させるために、仕様が統一された部品、半製品を調達する。
- イ 効率よく部品調達を行うために、関連会社から部品を調達する。
- ウ 中間在庫を極力減らすために、生産ラインにおいて、後工程の生産に必要な部品だけを前工程から調達する。
- エ より品質が高い部品を調達するために、部品の納入指定業者を複数定め、競争入札で部品を調達する。



問 56

BCP（事業継続計画）の策定、運用に関する記述として、適切なものはどれか。

- ア ITに依存する業務の復旧は、技術的に容易であることを基準に優先付けする。
- イ 計画の内容は、経営戦略上の重要事項となるので、上級管理者だけに周知する。
- ウ 計画の内容は、自社組織が行う範囲に限定する。
- エ 自然災害に加え、情報システムの機器故障やマルウェア感染も検討範囲に含める。



問53 プロダクトイノベーション

■ 20%

プロダクトは「製品」、イノベーションは「革新」という意味で、革新的な新製品を開発することを**プロダクトイノベーション**といいます。

製品を革新するプロダクトイノベーションに対して、製造プロセスや業務プロセスなどのプロセスを革新することを**プロセスイノベーション**といいます。

- ア 正解です。
- イ イノベーションによる「価値の収益化」の説明です。
- ウ MOT（技術経営）の説明です。
- エ プロセスイノベーションの説明です。

問54 ディジタルディバイト

■ 30%

ディジタルディバイトとは、IT技術を利用できる人とできない人との間に生じる社会的・経済的な格差のことです。

- ア 情報弱者の説明です。
- イ 情報バリアフリー環境の説明です。
- ウ アクセシビリティの説明です。
- エ 正解です。

問55 かんばん方式

■ 20%

かんばん方式とは、工程で必要となる部品を前の工程から必要に応じて調達し、工程と工程の間に生じる中間在庫を極力減らす生産方式です。もともとトヨタ自動車がはじめた方式で、部品の補充を知らせる生産指示票が「かんばん」と呼ばれていたことからかんばん方式といいます。ジャストインタイム生産方式ともいいます。正解はウです。

問56 BCP（事業継続計画）の策定・運用

■ 20%

BCP（事業継続計画）とは、不測の事態が発生した場合に、必要な業務を早期に復旧・再開できるように立てておく計画のことです。想定される不測の事態としては、地震・火災・津波などの災害のほかにも、情報システムの故障やサイバー攻撃、マルウェア感染などについても検討します。

- ア 技術的に容易かどうかではなく、事業継続に不可欠な業務を優先的に復旧します。
- イ 緊急事態発生時に計画通り行動できるよう、事業継続計画の内容は組織全体に周知徹底が必要です。
- ウ 計画の内容によっては、警備会社やITサービスプロバイダといった他社との連携が必要です。
- エ 正解です。



問53

プロダクトイノベーション
といえば

- 革新的な新製品を開発
- 製造プロセスや業務プロセスを革新



問53

Management Of Technology:
技術経営。技術に立脚する事業を行う企業が、技術開発に投資してイノベーション（技術革新）を促進し、事業を継続的に発展させていく経営の考え方。

○ 解答 ○

問53 ア	問54 エ
問55 ウ	問56 エ

問 57 CIO の果たすべき役割はどれか。

- ア 各部門の代表として、自部門のシステム化案を情報システム部門に提示する。
- イ 情報技術に関する調査、利用研究、関連部門への教育などを実施する。
- ウ 全社的観点から情報化戦略を立案し、経営戦略との整合性の確認や評価を行う。
- エ 豊富な業務経験、情報技術の知識、リーダーシップをもち、プロジェクトの運営を管理する。

問 58 ROI を説明したものはどれか。

- ア 一定期間におけるキャッシュフロー（インフロー、アウトフロー含む）に対して、現在価値でのキャッシュフローの合計値を求めるものである。
- イ 一定期間におけるキャッシュフロー（インフロー、アウトフロー含む）に対して、合計値がゼロとなるような、割引率を求めるものである。
- ウ 投資額に見合うリターンが得られるかどうかを、利益額を分子に、投資額を分母にして算出するものである。
- エ 投資による実現効果によって、投資額をどれだけの期間で回収可能かを定量的に算定するものである。

問 59 売上高が 100 百万円のとき、変動費が 60 百万円、固定費が 30 百万円掛かる。変動費率、固定費は変わらないものとして、目標利益 18 百万円を達成するのに必要な売上高は何百万円か。

ア 108

イ 120

ウ 156

エ 180

問 60 ソフトウェアやデータに瑕疵がある場合に、^{かし}製造物責任法の対象となるものはどれか。

- ア ROM 化したソフトウェアを内蔵した組込み機器
- イ アプリケーションソフトウェアパッケージ
- ウ 利用者が PC にインストールした OS
- エ 利用者によってネットワークからダウンロードされたデータ

解説

問 57 CIO の役割

CIO（最高情報責任者）は、企業の情報資源や IT に関する上級役員で、組

合格のカギ

■ 50%


合格のカギ

織全体の情報化戦略やIT投資計画を経営戦略に基づいて立案します。

- ア 各利用者部門のシステム担当者の役割です。
- イ 情報システム部門の役割です。
- ウ 正解です。
- エ プロジェクトマネージャの役割です。

問58 ROI
 30%

ROI（投資利益率：Return On Investment）は、投資した金額に対してどれだけの利益が得られたかを表すもので、次のように算出します。

$$\text{ROI} = \frac{\text{利益額}}{\text{投資額}} \times 100 (\%)$$

解答群は、いずれも投資を評価するための指標です。

- ア NPV（正味現在価値）の説明です。
- イ IRR（内部収益率）の説明です。
- ウ 正解です。
- エ 回収期間法の説明です。

問59 売上高の計算 キホン！
 20%

売上高100百万円のとき、変動費が60百万円なので、**変動費率**（売上高に対する変動費の割合）は $60 / 100 = 0.6$ になります。

目標利益を達成するのに必要な売上高を x 百万円とすると、このときの**変動費**は、変動費率×売上高より、 $0.6x$ 百万円と表せます。また、**固定費**は売上高にかかわらず変わらないで30百万円です。

売上高から原価（変動費と固定費の合計）を差し引いたものが利益になるので、目標利益が18百万円のとき、次の式が成り立ちます。

$$\frac{x - (0.6x + 30)}{\text{売上高} - \text{原価}} = 18 \quad \rightarrow \quad 0.4x = 48 \quad \therefore x = 120$$

以上から、必要な売上高は120百万円になります。正解はイです。

問60 製造物責任法
 20%

製造物責任法（PL法）は、製造物の欠陥によって損害が生じた場合の製造業者の責任について定めた法律です。この法律でいう「製造物」とは「製造または加工された動産」であり、ソフトウェアやデータは製造物とはみなされません。ただし、ソフトウェアやデータをROMに組み込み、そのROMを部品とする製品は「製造物」となるので、製造物責任法の対象になります。

- ア 正解です。
- イ ソフトウェアパッケージは入れ物なので、対象とはなりません。
- ウ OSはソフトウェアなので、単体では製造物責任法の対象とはなりません。
- エ ダウンロードされたデータには実体がなく、製造物とはみなされません。


覚えよう！

問57

執行役員の種類

- CEO：最高経営責任者
- COO：最高執行責任者
- CFO：最高財務責任者
- CIO：最高情報責任者
- CTO：最高技術責任者


NPV

問58

現金流入の現在価値から、現金流出の現在価値を差し引いたもの。


覚えよう！

問59

変動費といえは

- 売上高に比例してかかる費用（材料費、運送費など）

$$\text{変動費率} = \frac{\text{変動費}}{\text{売上高}}$$

$$\text{変動費} = \text{変動費率} \times \text{売上高}$$

固定費といえは

- 売上高にかかわらず一定の費用（人件費、家賃など）


製造物責任法

問60

製造物の瑕疵（欠陥）によって損害が生じた場合の製造業者の損害賠償責任について規定した法律。被害者は製品に欠陥があったことを証明すれば、製造業者の過失の有無にかかわらず賠償を請求できる。

○ 解答

- | | | | |
|-----|---------------------------------------|-----|---------------------------------------|
| 問57 | <input checked="" type="checkbox"/> ウ | 問58 | <input type="checkbox"/> ウ |
| 問59 | <input checked="" type="checkbox"/> イ | 問60 | <input checked="" type="checkbox"/> ア |

基本情報技術者試験 予想問題③

科目B



問

01

次のプログラム中の a と b に入れる正しい答えの組合せを、解答群の中から選べ。

ある通信会社では、データ通信量が3GB以下の場合の料金は0円、3GBを超え5GB以下の場合の料金は1,000円である。5Gを超える場合の料金については、超過分の小数第一位を切り上げ、1GBにつき500円を1,000円に加算する。関数 fee は、データ通信量(GB)を表す0以上の実数値を引数として受け取り、料金を返す。

[プログラム]

○ 整数型 : fee(実数型 : qt)

```
整数型 : ret
if (qt が 3.0 以下)
    ret ← 0
elseif ( a )
    ret ← 1000
else
    ret ← 1000 +  b
endif
return ret
```

解答群

	a	b
ア	qt が 5.0 以下	((qt - 5.0) + 1.0) の整数部 × 500
イ	qt が 5.0 以下	((qt - 5.0) + 0.9) の整数部 × 500
ウ	qt が 5.0 より小さい	((qt - 5.0) + 1.0) の整数部 × 500
エ	qt が 5.0 より小さい	((qt - 5.0) + 0.9) の整数部 × 500
オ	(qt が 3.0 より大きい) and (qt が 5.0 より小さい)	((qt - 5.0) + 0.9) の整数部 × 500
カ	(qt が 3.0 より大きい) or (qt が 5.0 以下)	((qt - 5.0) + 1.0) の整数部 × 500

※オリジナル問題

プログラムの解説

```
01 ○ 整数型 : fee( 実数型 : qt )
02 整数型 : ret
03 if (qt が 3.0 以下) ← データ通信量が 3GB 以下の場合
04   ret ← 0
05 elseif ( [ ] a ) ← 3GB を超え, 5GB 以下の場合
06   ret ← 1000
07 else ← 5GB 超の場合
08   ret ← 1000 + [ ] b
09 endif
10 return ret
```

if ~ elseif ~ else 構文

if ~ elseif ~ else 構文

この構文では、条件式を上から順番に評価し、最初に真（true）になった条件式に対応する処理を実行します。真になった条件式以降の条件式は評価せず、対応する処理も実行しません。条件式がどれも真にならなかったときは、else に対応する処理 $n + 1$ を実行します。

```
if [ 条件式 1 ]
  処理 1 ← 条件式 1 が真のとき実行
elseif [ 条件式 2 ]
  処理 2 ← 条件式 2 が真のとき実行
  :
elseif [ 処理 n ]
  処理 n ← 条件式 n が真のとき実行
else
  処理 n + 1 ← 条件式 1 ~ n がいずれも真でないとき実行
endif
```

else と処理の組は、最後に 1 つだけ記述することができます。また必要なければ省略することもできます。

問題解説

行番号 03, 04 : 条件式「qt が 3.0 以下」が真のとき、変数 ret に 0 が代入されます。この処理は、問題文の「データ通信量が 3GB 以下の場合の料金は 0 円」に対応します。

```
03     if (qt が 3.0 以下)
04       ret ← 0
```

行番号 05, 06 : 空欄 a の条件式が真のとき、変数 ret に 1000 が代入されます。通信料金が 1,000 円になるのは、「データ通信量が 3GB を超え 5GB 以下の場合」なので、空欄の条件式は次のように書けます。

(qt が 3.0 より大きい) and (qt が 5.0 以下)

ただし、行番号 05 は行番号 03 の条件式が真でないときに限り実行評価されるので、「qt が 3.0 より

大きい」ことは明らかです。したがって、空欄 a に入る条件式は「qt が 5.0 以下」のみでかいません。

```
05  elseif (qt が 5.0 以下)
06      ret ← 1000
```

↑ 空欄 a

行番号 08：データ通信量が 5G を超える場合は、超過分の小数第一位を切り上げ、1GB につき 500 円を 1,000 円に加算します。たとえばデータ通信量が 5.6GB の場合、超過分は 0.6GB の小数第一位を切り上げ、1GB になります。この場合の通信料金は $1 \times 500 = 500$ 円を 1,000 円に加算し、1,500 円になります。

超過分の小数第一位を切り上げるには、**超過分に 0.9 を加算し、小数点以下を切り捨てます**（超過分に 1.0 を加算してしまうと、超過分が 1.0 ちょうどのとき、2.0 になってしまいます）。

((qt - 5.0) + 0.9) の整数部 × 500
超過分 加算 小数点以下を切り捨て

```
07  else
08      ret ← 1000 + ((qt - 5.0) + 0.9) の整数部 × 500
09  endif
```

↑ 空欄 b

以上から、空欄 a が「qt が 5.0 以下」、空欄 b が「((qt - 5.0) + 0.9 の整数部) × 500」のイが正解です。

○ 解答 ○
問 01 イ



合格の力ギ

小数点以下の四捨五入と切上げ

プログラム言語の計算では、実数値の小数点以下を切り捨てるのは比較的簡単で、実数値を整数に変換すれば、小数点以下は自動的に切り捨てられます。一方、四捨五入や切上げには次のような工夫が必要です。

①小数点以下の四捨五入

実数値に 0.5 を加算し、小数点以下を切り捨てます。

例：1.4 → 1.4 + 0.5 = 1.9 → 1
1.5 → 1.5 + 0.5 = 2.0 → 2

②小数第一位で切上げ

実数値に 0.9 を加算し、小数点以下を切り捨てます（注：小数第二位で切り上げる場合は、0.99 を加算します）。

例：1.2 → 1.2 + 0.9 = 2.1 → 2
2.0 → 2.0 + 0.9 = 2.9 → 2



02

次のプログラム中の と に入れる正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まる。

次のプログラムは、整数型の配列 array の要素の並びを逆順にする。

[プログラム]

整数型の配列 : array $\leftarrow \{1, 2, 3, 4, 5\}$

整数型 : right, left

整数型 : tmp

```
for (left を 1 から (array の要素数 ÷ 2 の商) まで 1 ずつ増やす)
    right  $\leftarrow$  
    tmp  $\leftarrow$  array[right]
    array[right]  $\leftarrow$  array[left]
      $\leftarrow$  tmp
endfor
```

解答群

	a	b
ア	array の要素数 - left	array[left]
イ	array の要素数 - left	array[right]
ウ	array の要素数 - left + 1	array[left]
エ	array の要素数 - left + 1	array[right]

※サンプル問題より

```

01 整数型の配列： array ← {1, 2, 3, 4, 5} ←----- 配列の宣言と初期化
02 整数型： right, left
03 整数型： tmp

```

```

04 for (left を 1 から (array の要素数 ÷ 2 の商) まで 1 ずつ増やす)
05   right ← [a]
06   tmp ← array[right]
07   array[right] ← array[left]
08   [b] ← tmp
09 endfor

```

値の交換

for 構文

01 | 配列の宣言と初期化

配列とは、複数の値をまとめて格納しておく値の入れ物です。1つの配列には、同じデータ型の値しか格納できません。

配列の宣言では、格納する値のデータ型と配列名を次のように記述します。

[データ型]の配列 : [配列名]

行番号 01 では、整数型の値を格納する `array` という名前の配列を宣言しています。また、行番号 01 では、配列の宣言と同時に値の代入も行っています。配列の内容は、

{[値 1], [値 2], [値 3], ...}

のように、{} 内にカンマで区切って記述します。

04 | for 構文

for 構文は、`for ~ endfor` の間の処理を、指定した条件にしたがって繰り返します。行番号 04 では次のようにになります。

繰返しごとに値を変化させる変数

```

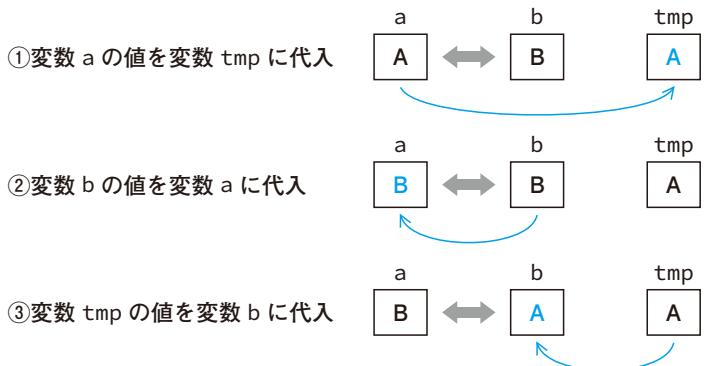
for (left を 1 から (array の要素数 ÷ 2 の商) まで 1 ずつ増やす)
    ↑      ↑      ↑
    初期値 終値 増分値

```

繰返し 1 回ごとに、変数 `left` の値を 1, 2, 3…のように 1 ずつ増やしていきます。`left` の値が「`array` の要素数 ÷ 2 の商」になったときが最後の繰り返しです。配列 `array` の要素数は 5 ですから、「`array` の要素数 ÷ 2 の商」は 2 になります。つまり、繰返し回数は全部で 2 回になります。

値の交換

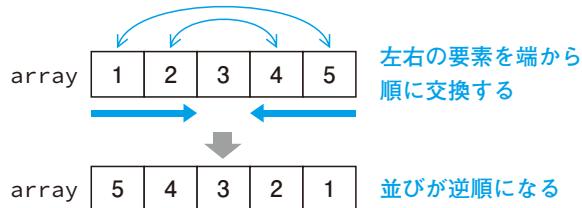
プログラムで 2 つの変数の値を交換するときは、よく使われる決まった処理があります。まず、変数を 1 つ用意します。このプログラムでは `tmp` という変数です。次に、値を交換したい一方の変数の値を `tmp` に移します。次に、もう一方の変数の値を、先ほど値を移しておいた変数に代入します。最後に、変数 `tmp` の値をもう一方の変数に代入すれば完了です。



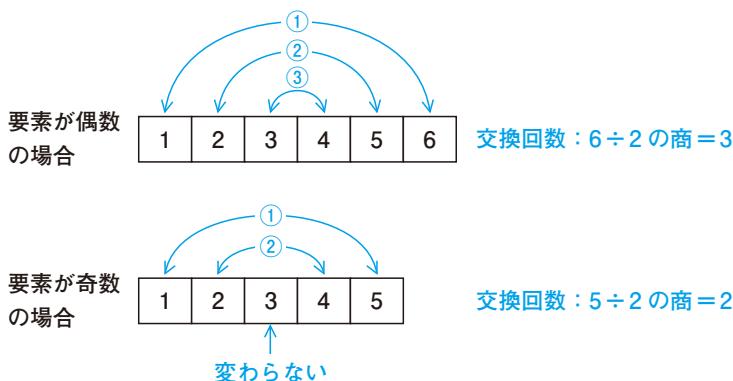
本問のプログラムでは、行番号 06～08 で値の交換を行っています。

問題解説

配列の要素の並びを逆順にするには、配列の左端と右端から同時にスタートし、どちらも真ん中へ進みながら値を交換していきます。



配列の要素数が偶数の場合は、最後に真ん中の 2 つの要素を交換して終了します。配列の要素数が奇数の場合は、真ん中の要素は逆順にしても変わりません。交換回数は「(array の要素数 ÷ 2) の商」になります。



空欄 a：交換する要素の要素番号は、左側を変数 `left`、右側を変数 `right` に設定します。変数 `left` が 1 から「(array の要素数 ÷ 2) の商」まで 1 ずつ増えていくのに対し、変数 `right` は array の要素数から「(array の要素数 ÷ 2) の商」まで、1 ずつ減っていきます。変数 `left` と変数 `right` の対応関係は次のようにになります。

`left` の値 : 1, 2, 3, ...,
`right` の値 : array の要素数, array の要素数 - 1, array の要素数 - 2, ...

上記から、変数 `right` の値に対応する変数 `left` の値から求めるには、次のようにすればよいことがわかります。

05

`right ← array の要素数 - left + 1 ← 空欄 a`

空欄 b：左側の `array[left]` と、右側の `array[right]` の値を交換します。値の交換は、次のような手順となります。

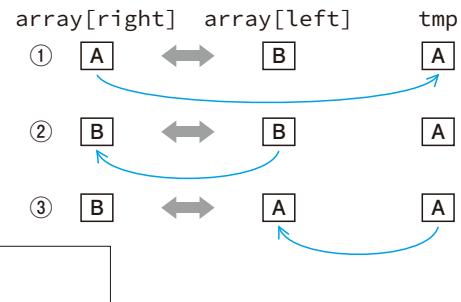
- ① `array[right]` の値を変数 `tmp` に代入する
- ② `array[left]` の値を `array[right]` に代入する
- ③ 変数 `tmp` の値を `array[left]` に代入する

擬似言語では、次のような処理になります。

```
06 tmp ← array[right]
07 array[right] ← array[left]
08 array[left] ← tmp
```

↑ 空欄 b

以上から、**空欄 b** には「`array[left]`」が入ります。



空欄 a に「`array の要素数 - left + 1`」、空欄 b に「`array[left]`」が入るので、正解は **ウ** です。

解答

問02 **ウ**



合格の力ギ

配列の要素番号

一般的なプログラム言語では、配列の要素番号は 0 から始まる場合が多いのですが、情報処理試験の擬似言語では 1 から始まる場合が多いので注意が必要です。問題文を必ず確認しましょう。

配列の要素番号の計算では、開始番号が「0」か「1」か、「+ 1」「- 1」は必要かなどに注意してください。たとえば配列の要素番号が 0 から始まる場合、本問のプログラムは次のようになります。

開始番号は 0

末尾の番号は要素数 - 1 になる

```
04 for (left を 0 から ((array の要素数 - 1) ÷ 2 の商) まで 1 ずつ増やす)
05   right ← array の要素数 - left - 1 ← 「+ 1」が「- 1」に
  :
```



03

次の記述中の **a**, **b** に入れる正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まる。

関数 max は、引数に指定した整数型の配列の中から、最大値を返す関数である。

プログラム 1 は、引数に指定した配列の要素数が最大 **a** のときエラーになる。一方、プログラム 2 は、引数に指定した配列の要素数が最大 **b** のときエラーになる。

[プログラム 1]

```
○ 整数 : max( 整数型の配列 : array )
整数 : i, ret
ret ← array[1]
i ← 2
do
    if (array[i] が ret より大きい)
        ret ← array[i]
    endif
    i ← i + 1
while(i が array の要素数 以下 )
return ret
```

[プログラム 2]

```
○ 整数 : max( 整数型の配列 : array )
整数 : i, ret
ret ← array[1]
i ← 2
while (i が array の要素数 以下 )
    if (array[i] が ret より大きい)
        ret ← array[i]
    endif
    i ← i + 1
endwhile
return ret
```

解答群

	a	b
ア	0	0
イ	0	1
ウ	1	0
エ	1	1

※オリジナル問題

プログラムの解説

[プログラム 1]

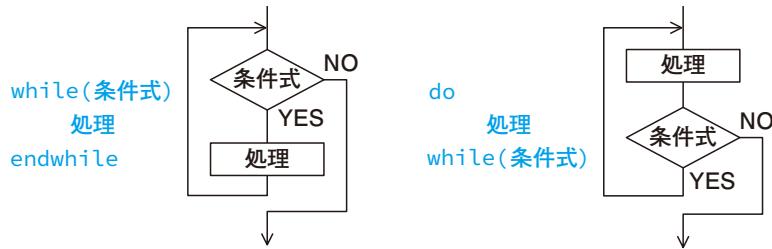
```
01 ○ 整数 : max( 整数型の配列 : array )
02 整数 : i, ret
03 ret ← array[1]
04 i ← 2
05 do
06     if (array[i] が ret より大きい)
07         ret ← array[i]
08     endif
09     i ← i + 1
10 while(i が array の要素数 以下 )
11 return ret
```

[プログラム 2]

```
01 ○ 整数 : max( 整数型の配列 : array )
02 整数 : i, ret
03 ret ← array[1]
04 i ← 2
05 while (i が array の要素数 以下 )
06     if (array[i] が ret より大きい)
07         ret ← array[i]
08     endif
09     i ← i + 1
10 endwhile
11 return ret
```

while ~ endwhile と do ~ while

while ~ endwhile と do ~ while は、どちらも条件式が真の間、処理を繰返し実行します。



2つの構文の違いは、while～endwhile が繰返し処理の入口で条件式を評価するのに対し、do～while は、繰返し処理の出口で条件式を評価することです。そのため、while～endwhile では繰返し処理が1度も実行されないことがあるのに対し、do～while では最低1回はかならず繰返し処理が実行されます。

最大値を求める

配列から最大値を求めるには、配列の要素を先頭から順に調べて、参照した値がこれまでより大きければ、その値を暫定最大値として記憶します。配列を最後まで調べ終わったときの暫定最大値が、実際の最大値となります。

問題解説

空欄a：プログラム1は、行番号03でarray[1]の内容を変数retに代入します。そのため、もし配列arrayの要素数が0だった場合には、この時点でエラーになります。

続いて行番号04で変数iに初期値2を代入してから、do～whileの繰返し処理に入ります。先ほど説明したように、do～whileの繰返し処理は最低でも1回はかならず実行されます。

行番号06では、条件式「array[i]がretより大きい」を評価します。変数iの初期値は2なので、初回はarray[2]が参照されます。したがって、配列arrayの要素数が1以下の場合はこの時点でエラーになります。

以上から、プログラム1では、配列arrayの要素数が「1以下」のときにエラーになります。

```
01 ○ 整数 : max( 整数型の配列 : array )
02   整数 : i, ret
03   ret ← array[1] ← 要素数0のときエラー
04   i ← 2
05   do
06     if (array[i] が ret より大きい) ← 要素数1のときエラー
07       ret ← array[i]
08     endif
09     i ← i + 1
10   while(i が array の要素数 以下 )
11   return ret
```

空欄b：プログラム2も、行番号03でarray[1]の内容を変数retに代入します。そのため、もし配列arrayの要素数が0だった場合には、この時点でエラーになります。

プログラムは変数iに初期値2を代入した後、while～endwhileの繰返し処理に入ります。ただし、

配列 array の要素数が 1 の場合、while の条件式「 i が array の要素数 以下」が偽になるため、繰り返し処理は 1 度も実行されません。この場合プログラムはエラーにならず、要素番号 1 の値を最大値として返します。

したがってプログラム 2 では、配列 array の要素数が「0」のときのみエラーになります。

```

01 ○ 整数 : max( 整数型の配列 : array )
02   整数 : i, ret
03   ret ← array[1] ← 要素数 0 のときエラー
04   i ← 2
05   while (i が array の要素数 以下) ← 要素数が 1 のときは繰り返し
06     if (array[i] が ret より大きい) 处理が実行されない
07       ret ← array[i]
08     endif
09     i ← i + 1
10   endwhile
11   return ret

```

以上から、空欄 a が「1」、空欄 b が「0」の **ウ** が正解です。

○ 解答

問03 **ウ**



合格のカギ 端の数

プログラミングで、端の数をどう処理するかはデリケートな問題です。条件式では「以上」とするか「より大きい」とするのか、配列の要素番号に +1 や -1 が必要かどうかなどに注意しましょう。

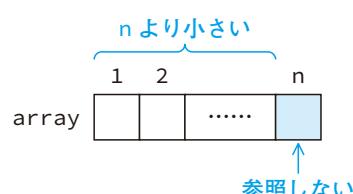
とくに擬似言語では、配列の要素番号が 1 から始まる場合が多いため注意が必要です。

●配列の要素番号が 1 から始まる場合

```

n ← array の要素数
i ← 1
✗ while (i が n より小さい)
    array[i] を参照
    i ← i+1
endwhile

```

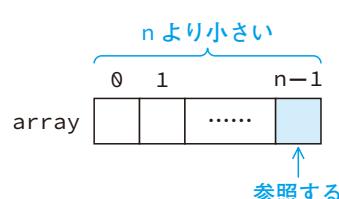


●配列の要素番号が 0 から始まる場合

```

n ← array の要素数
i ← 0
○ while (i が n より小さい)
    array[i] を参照
    i ← i+1
endwhile

```





04

次のプログラム中の a と b に入れる正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まる。

関数 dec2bin は、引数に指定した 0 以上の 10 進整数を 2 進数に変換し、その各桁を下位から順に配列 bin に格納する。たとえば、`dec2bin(10)` を実行すると、配列 bin の内容は 2 進数 `1010` の各桁を下位から順に格納し、`{0,1,0,1}` となる。

[プログラム]

```
大域 : 整数型の配列 : bin
○ 整数型の配列 : dec2bin( 整数型 : dec )
    整数 : num ← dec
    bin ← {}
    do
        bin の末尾に (num mod 2) の値を追加する
         a
    while ( b)
```

解答群

	a	b
ア	<code>num ← num mod 2</code>	num が 0 より大きい
イ	<code>num ← num mod 2</code>	num が 0 でない
ウ	<code>num ← num ÷ 2 の商</code>	num が 0 に等しい
エ	<code>num ← num ÷ 2 の商</code>	num が 0 でない
オ	<code>num ← num × 2</code>	num が 0 より大きい
カ	<code>num ← num × 2</code>	num が 0 に等しい

※オリジナル問題

プログラムの解説

```
01 大域 : 整数型の配列 : bin
02 ○ 整数型の配列 : dec2bin( 整数型 : dec )
03     整数 : num ← dec
04     bin ← {}
05     do
06         bin の末尾に (num mod 2) の値を追加する
07          a
08     while ( b)
```

10進数→2進数の変換

10進整数を2進数に変換するには、10進整数を0になるまで繰返し2で割り、その余りを下から順に並べます。

例：10進数の10を2進数に変換

$$\begin{array}{r} 10 \div 2 = 5 \text{ 余り } 0 \\ \downarrow \\ 5 \div 2 = 2 \text{ 余り } 1 \\ \downarrow \\ 2 \div 2 = 1 \text{ 余り } 0 \\ \downarrow \\ 1 \div 2 = 0 \text{ 余り } 1 \end{array}$$

下から
順に並べる

1010

上の例のように、10進数の10は2進数で「1010」になります。

なお、関数 `dec2bin` は2進数の各桁を下位から順に並べるため、`dec2bin(10)` を実行すると配列 `bin` の内容は {0, 1, 0, 1} になります。

問題解説

空欄a：10進整数 `num` を2進数に変換するには、繰返し処理で `num` の値を2で割ります。行番号07ではこの処理を行います。

07 `num ← num ÷ 2 の商 ← 空欄a`

空欄b：空欄の条件式が真の間、`do ~ while` の処理を繰り返します。繰返しは `num` の値が0になるまで続けます。つまり、`num` の値が0ではない間は、処理を繰り返します。したがって空欄の条件式は次のようにになります。

08 `while (num が 0 でない)`

↑
空欄b

以上から、空欄aが「`num ← num ÷ 2 の商`」、空欄bが「`num が 0 でない`」の **工** が正解です。

○ 解答 ○

問04 工

**合格のカギ**

繰返しの条件

`while ~ endwhile` や `do ~ while` には、繰返しを継続する条件を指定します。これを、繰返しの終了条件と勘違いする人がときどきいます。たとえば、

```
while (num が 0 でない)
    处理
endwhile
```

は、変数 `num` の値が0でない間、処理を繰り返します。言い換えると、変数 `num` の値が0になったら繰返しを終了します。



問 05

次の記述中の [a] と [b] に入る正しい答えの組合せを、解答群の中から選べ。

関数 `gcd` は引数に指定した 2 つの整数の最大公約数を返す関数、関数 `lcm` は引数に指定した 2 つの整数の最小公倍数を返す関数である。 $3, 6, 9$ の 3 つの数の最小公倍数を求めるために、`lcm(lcm(3, 6), 9)` を実行した。このとき、 α の行は [a] 回、 β の行は [b] 回実行される。

[プログラム 1]

○ 整数型 : `gcd(整数型 : a, 整数型 : b)`

```
整数型 : m, n, r
m ← a
n ← b
do
    r ← m mod n ← α
    m ← n
    n ← r
while (r が 0 でない)
return m
```

[プログラム 2]

○ 整数型 : `lcm(整数型 : a, 整数型 : b)`

```
return (a ÷ gcd(a, b) の商) × b ← β
```

解答群

	a	b
ア	3	1
イ	3	2
ウ	4	1
エ	4	2
オ	5	1
カ	5	2
キ	6	1
ク	6	2

※オリジナル問題

プログラムの解説

[プログラム 1]

```
01 ○ 整数型 : gcd( 整数型 : a, 整数型 : b )
02   整数型 : m, n, r
03   m ← a
04   n ← b
05   do
06     r ← m mod n ← α
07     m ← n
08     n ← r
09   while (r が 0 でない)
10   return m
```

[プログラム 2]

```
01 ○ 整数型 : lcm( 整数型 : a, 整数型 : b )
02   return (a ÷ gcd(a, b) の商) × b ← β
```

最大公約数

最大公約数とは、2つの整数に共通する約数のうち最大の数です。たとえば、18の約数は1, 2, 3, 6, 9, 18の6個、10の約数は1, 2, 5, 10の4個です。共通する約数1, 2のうち、最大の数2が最大公約数になります。

最大公約数は**ユークリッドの互除法**と呼ばれるアルゴリズムを用いて求めることができます。このアルゴリズムは、

- ① $a \div b$ の余り r を求める
- ② a を b の値に、 b を r の値に置き換える

という手順を繰り返すと、 r が 0 になったときの b が最大公約数になるというものです。たとえば、 $a = 18$, $b = 12$ の場合は、次のようになります。

$$\begin{array}{ccc} a & b & r \\ 18 & \div & 10 = 1 \text{ 余り } 8 \\ 10 & \div & 8 = 1 \text{ 余り } 2 \\ 8 & \div & 2 = 4 \text{ 余り } 0 \\ & \uparrow & \\ & \text{最大公約数} & \end{array}$$

本問の関数 `gcd` は、ユークリッドの互除法を用いて引数に指定した2つの整数の最大公約数を返します。

最小公倍数

最小公倍数とは、2つの整数に共通する倍数のうちの最小の数です。たとえば、6と9の最小公倍数は、6の倍数(6, 12, 18, 24, 30, 36, ...)と9の倍数(9, 18, 27, 36, 45, ...)に共通する数のうち、最小の数18になります。

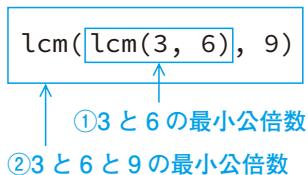
2つの数 a , b の最小公倍数は、 a , b の最大公約数を用いて次のように求めることができます。

$$a \div (\text{a と b の最大公約数}) \times b$$

たとえば 6 と 9 の最小公倍数は、6 と 9 の最大公約数 3 を使えば、 $6 \div 3 \times 9 = 18$ となります。本問の関数 `lcm` は、この方法を用いて引数に指定した 2 つの数の最小公倍数を返します。

問題解説

`lcm(lcm(3, 6), 9)` は、まず `lcm(3, 6)` を実行して 3 と 6 の最小公倍数 6 を求め、さらに 6 と 9 を引数に指定して関数 `lcm` を実行します。



`lcm(3, 6)` を実行すると、 β の行が 1 回実行されます。この行は、関数 `gcd(3, 6)` を呼び出して、3 と 6 の最大公約数を求めます。

β の行は r が 0 になるまで繰り返されるので、次のように 2 回実行されます。

```
m      n
r ← 3 mod 6 ← 3 ÷ 6 の余り 3 を r に代入
r ← 6 mod 3 ← 6 ÷ 3 の余り 0 を r に代入
```

`lcm(3, 6)` は、`gcd(3, 6)` の戻り値 3 を使って、3 と 6 の最小公倍数 $3 \div 3 \times 6 = 6$ を返します。

次に、`lcm(3, 6)` の戻り値 6 を使って `lcm(6, 9)` が実行され、2 回目の β の行が実行されます。この行は、関数 `gcd(6, 9)` を呼び出して、6 と 9 の最大公約数を求めます。

α の行は r が 0 になるまで繰り返されるので、次のように 3 回実行されます。

```
m      n
r ← 6 mod 9 ← 6 ÷ 9 の余り 6 を r に代入
r ← 9 mod 6 ← 9 ÷ 6 の余り 3 を r に代入
r ← 6 mod 3 ← 6 ÷ 3 の余り 0 を r に代入
```

`lcm(6, 9)` は、`gcd(6, 9)` の戻り値 3 を使って、6 と 9 の最小公倍数 $6 \div 3 \times 9 = 18$ を返します。

以上から、 α の行は合計 5 回、 β の行は合計 2 回実行されます。正解は **力** です。

問05 **力**



合格のカギ

プログラムのトレース

プログラムの処理に応じて、変数の値がどのように変化するか追跡することをトレースといいます。変数の変化は、プログラム中のある行に注目し、繰り返し 1 回ごとに表にするとわかりやすくなります。本問の場合は、 α の行に注目して、変数 m と n の変化をトレースします。



06

次のプログラム中の と に入れる正しい答えの組合せを、解答群の中から選べ。

手続 `delete` は、引数で与えられた文字を单方向リストから削除する手続である。单方向リストの各要素は、クラス `ListElement` を用いて表現する。クラス `ListElement` の説明を図に示す。`ListElement` 型の変数はクラス `ListElement` のインスタンスの参照を格納するものとする。

メンバ変数	型	説明
<code>val</code>	文字型	リストに格納する文字。
<code>next</code>	<code>ListElement</code>	リストの次の文字を保持するインスタンスの参照。初期状態は未定義である。

図 クラス `ListElement` の説明

单方向リストの先頭の要素の参照は、大域変数 `listHead` に格納されているものとする。リストが空のときは、`listHead` は未定義である。

〔プログラム〕

```

大域 : ListElement: listHead

○ delete( 文字型 : qVal )
    ListElement: prev, curr
    curr ← listHead
    while (curr.val が qVal と等しくない)
        if (curr.next が 未定義 )
            return /* 見つからなかった場合 */
        else
            prev ← curr
            
            endif
        endwhile
        if (curr が listHead と等しい)
            listHead ← listHead.next
        else
            
            endif
    
```

解答群

	a	b
ア	curr \leftarrow curr.next	prev.next \leftarrow curr
イ	curr \leftarrow curr.next	prev.next \leftarrow curr.next
ウ	curr \leftarrow curr.next	prev.next \leftarrow 未定義
エ	curr \leftarrow prev.next	prev.next \leftarrow curr
オ	curr \leftarrow prev.next	prev.next \leftarrow curr.next
カ	curr \leftarrow prev.next	prev.next \leftarrow 未定義

※オリジナル問題

プログラムの解説

```

01 ○ delete( 文字型 : qVal )
02     ListElement: prev, curr
03     curr  $\leftarrow$  listHead ← 先頭から順に探索
04     while (curr.val が qVal でない)
05         if (curr.next が 未定義 )
06             return /* 見つからなかった場合 */
07         else
08             prev  $\leftarrow$  curr
09             [a]
10         endif
11     endwhile
12     if (curr が listHead と等しい)
13         listHead  $\leftarrow$  listHead.next ← 先頭の要素を削除
14     else
15         [b] ← 先頭以外の要素を削除
16     endif

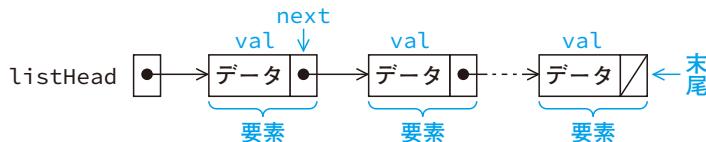
```

单方向リスト

单方向リスト（線形リスト）は、リストを構成する個々の要素（データの入れ物）を数珠つなぎにして、複数のデータをひと連なりにまとめたデータ構造です。個々の要素は、

- ①データを格納する部分 (val)
- ②次の要素の参照を格納する部分 (next)

で構成されています。前の要素の next に、次の要素の参照を格納し、次の要素の next にまた次の要素の参照を格納し——という操作を繰り返して、複数の要素をつなげていきます。



単方向リストに含まれる要素を参照するには、先頭要素から次の要素へと、目的の要素に到達するまでリストを順にたどっていきます。

クラスの定義

単方向リストの要素のような構造をもったデータは、整数型、文字型といった通常のデータ型では表現できません。そのため本問では、`ListElement` というクラスを用いてリストの要素を表現しています。

クラスとは、本来はオブジェクト指向プログラミングの用語ですが、ここでは複数のメンバ変数とメソッドを備えたデータ型の一種と考えてかまいません。クラスを定義するための構文は擬似言語には用意されていないので、使用するクラスについては問題文に説明があります。

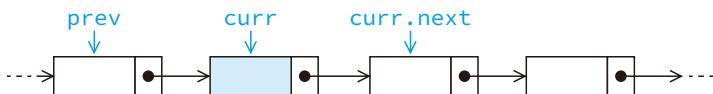
クラスを使用するには、通常のデータ型と同様に、

クラス名 : **変数名**

のように変数を宣言します。本問のプログラムでは、行番号02で`prev`、`curr`という2つの`ListElement`型の変数を宣言しています。

問題解説

行番号 04～11：プログラムは、単方向リストを先頭から順にたどって、削除する要素を探します。



変数 `curr` の要素が目的の要素でなかった場合は、次の要素を `curr` に設定します。行番号 09 の空欄 a にはこの処理が入ります。

次の要素は `curr` のメンバ変数 `next` に参照されているので、

`curr ← curr.next ← 次の要素を変数 curr に設定`

とします。

```

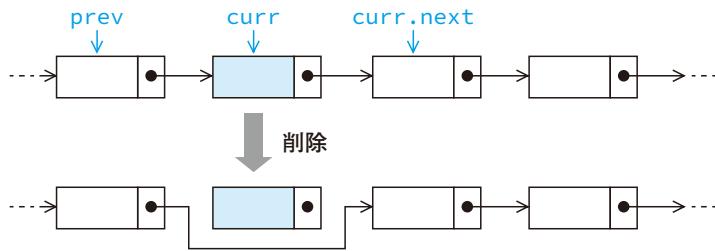
04     while (curr.val が qVal でない)
05         if (curr.next が 未定義)
06             return /* 見つからなかった場合 */
07         else
08             prev ← curr
09             curr ← curr.next ← 空欄 a
10         endif
11     endwhile
  
```

行番号 12～16：目的の要素が見つかった場合は、その要素をリストから削除します。目的の要素がリストの先頭にある場合は、次の要素を `listHead` に設定します。

```

12     if (curr が listHead と等しい)
13         listHead ← listHead.next
  
```

先頭要素以外の要素をリストから削除するには、削除する要素の1つ前の要素の `next` を、削除する要素の次の要素につけかえます。



削除したい要素は変数 `curr`, 1つ前の要素は変数 `prev` に設定されているので、現在は `curr` を指している `prev.next` を、`curr.next` につけかえれば、`curr` がリストからはずれます。

`prev.next ← curr.next ← 空欄 b`

以上から、空欄 **a** が「`curr ← curr.next`」、空欄 **b** が「`prev.next ← curr.next`」の **イ** が正解です。

○ 解答 ○
問06 イ



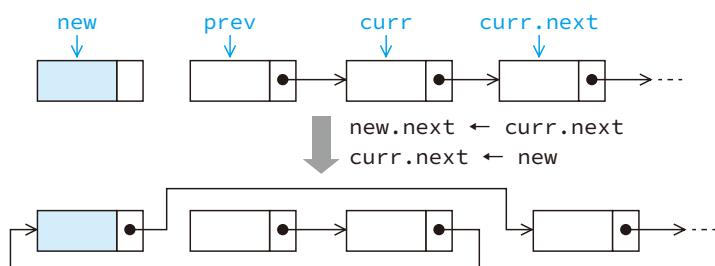
合格のカギ

単方向リストの操作

単方向リストに要素を挿入する場合は、次のようにになります。単方向リストは、このようにメンバ変数 `next` の書き換えによって、挿入や削除などの操作が行えます。

● `curr` の後に新しい要素 `new` を挿入する

```
new.next ← curr.next
curr.next ← new
```





07

次のプログラム中の **a** と **b** に入る正しい答えの組合せを、解答群の中から選べ。

手続 `reverse` は、スタックを用いて英単語のスペルを逆順にするプログラムである。スタックはクラス `Stack` を用いて表現する。クラス `Stack` の説明を図に示す。本問において、文字列に対する演算子 “+” は、文字列の連結を表す。

メソッド	説明
<code>push(文字列型 : s)</code>	文字列 <code>s</code> をスタックに格納する。
文字列型 : <code>pop()</code>	スタックから文字列を取り出す。
コンストラクタ	説明
<code>Stack(整数型 : n)</code>	大きさ <code>n</code> のスタックを生成する。

図 クラス `Stack` の説明

〔プログラム〕

```

○文字列型 : reverse( 文字列型 : word)
文字列型 : buf ← ""
Stack: stack ← Stack(word の文字数 )
整数型 : cnt ← 1
while ( [ a ] )
    stack.push(word の cnt 文字目の 1 文字から成る文字列 )
    cnt ← cnt + 1
endwhile
do
    cnt ← cnt - 1
    buf ← [ b ]
while (cnt が 1 より大きい)
return buf

```

解答群

	a	b
ア	cnt が word の文字数 以下	<code>buf + stack.pop()</code>
イ	cnt が word の文字数 以下	<code>stack.pop() + buf</code>
ウ	cnt が word の文字数 より小さい	<code>buf + stack.pop()</code>
エ	cnt が word の文字数 より小さい	<code>stack.pop() + buf</code>

※オリジナル問題

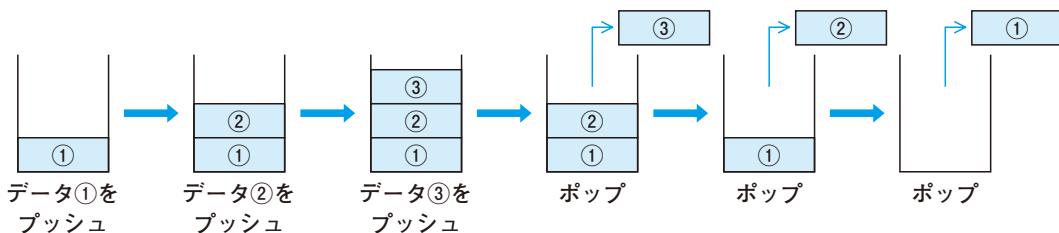
プログラムの解説

```
01 ○ 文字列型 : reverse( 文字列型 : word )
02 文字列型 : buf ← "" ← 初期値は空の文字列
03 Stack: stack ← Stack(word の文字数 )
04 整数型 : cnt ← 1
05 while (  a  )
06     stack.push(word の cnt 文字目から成る文字列 ) ← スタックに 1 文字
07         プッシュ
08 endwhile
09 do
10     cnt ← cnt - 1
11     buf ←  b ← スタックから取り出した文字を変数 buf に格納
12 while (cnt が 1 より大きい )
13 return buf
```

【スタック】

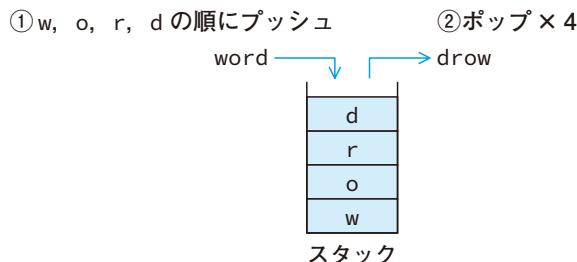
スタックは、プッシュとポップという、2つの基本操作をもったデータ構造です。プッシュはスタックにデータを格納する操作、ポップはスタックからデータを取り出す操作です。このとき重要なのは、後に格納したデータから先に取り出されるという、**後入れ先出し** (LIFO : Last In First Out) の原則です。

たとえば、データを1, 2, 3の順にスタックにプッシュします。次にスタックからデータをポップすると、取り出されるデータは3, 2, 1の順になります。格納するときはデータを一番上に積み上げ、取り出すときは一番上から取り出していくイメージです。



【文字列を逆順にする】

スタックを使うと、データの並びを簡単に逆順にできます。文字列を逆順にするには、文字列を1文字ずつスタックに格納し、次にスタックから1文字ずつ取り出して連結します。



問題解説

空欄 a : 行番号 05 ~ 08 は、文字列を 1 文字ずつスタックに格納する処理です。

```
05  while ( [ ] a )
06      stack.push(word の cnt 文字目の 1 文字から成る文字列 )
07      cnt ← cnt + 1
08  endwhile
```

スタックには、word の文字列を 1 文字目から順に格納します。したがって変数 cnt は、1 から (word の文字数) まで 1 ずつ増やします。変数 cnt が (word の文字数) を超えたら繰り返しを抜けるので、空欄 a の条件式は次のようにになります。

cnt が word の文字数 以下 ← 空欄 a

空欄 b : 行番号 09 ~ 12 は、スタックに格納した文字を取り出して、文字列 buf に格納します。

```
09  do
10      cnt ← cnt - 1
11      buf ← [ ] b
12  while (cnt が 1 より大きい)
```

スタックからデータを取り出すには、メソッド pop を使用します。文字は、word の末尾から順に出てくるので、これを文字列 buf の末尾に連結していくれば、word の逆順になります。

buf ← buf + stack.pop() ← 空欄 b

以上から、空欄 a が「cnt が word の文字数 以下」、空欄 b が「buf + stack.pop()」のアが正解です。

○ 解答 ○

問07 ア

予想問題
3

科目
B



合格のカギ スタックとキュー

スタックとキューは基本的なデータ構造なので、プログラム問題ではよく出題されると予想されます。スタックとキューの使い方を仕事にたとえると、入って来た仕事を入って来た順番に片付けていくのがキュー、新しい仕事が入ってきたら今の仕事を中断して、新しい仕事から順に片付けていくのがスタックです。



08

次のプログラム中の a と b に入れる正しい答えの組合せを、解答群の中から選べ。

構成する全ての節について、左の子のキー値は親のキー値より小さく、右の子のキー値は親のキー値より大きい二分木を二分探索木という。二分探索木の例を図 1 に示す。

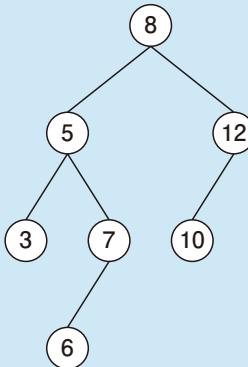


図 1 二分探索木の例

関数 `findNode` は、二分探索木の根の参照とキー値を引数として受け取り、二分探索からキー値を検索して、見つかったノードの参照を返す関数である。二分探索木の節は、クラス `Node` を用いて表現する。クラス `Node` の説明を図 2 に示す。`Node` 型の変数は、クラス `Node` のインスタンスの参照を格納する。

メンバ変数	型	説明
<code>val</code>	整数型	二分探索木に格納する整数。
<code>left</code>	<code>Node</code>	左の子への参照。空のときは未定義の値となる。
<code>right</code>	<code>Node</code>	右の子への参照。空のときは未定義の値となる。

図 2 クラス `Node` の説明

[プログラム]

```
○ Node: findNode(Node: root, 整数: key)
    Node: curr ← root
    while (curr.val が key ではない)
        if ( a)
            curr ← curr.left
        else
            curr ←  b
```

```

    endif
    if (curr が 未定義)
        return 未定義の値
    endif
endwhile
return curr

```

解答群

	a	b
ア	key が curr.val より小さい	curr.right
イ	key が curr.val より大きい	curr.right
ウ	key が curr.val より小さい	curr.left
エ	key が curr.val より大きい	curr.left

※オリジナル問題

プログラムの解説

```

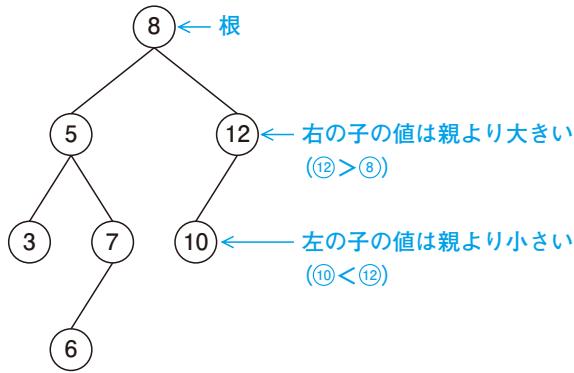
01 ○ Node: findNode(Node: root, 整数: key)
02     Node: curr ← root ←———— 根から探索
03     while (curr.val が key ではない)
04         if ([a])
05             curr ← curr.left ←———— 左の子に移動
06         else
07             curr ← [b]
08         endif
09         if (curr が 未定義)
10             return 未定義の値 ←———— 見つからなかった場合
11         endif
12     endwhile
13     return curr ←———— 見つかった場合

```

| 二分探索木

複数の節（ノード）をツリー状に接続したデータ構造を木構造といいます。下位に接続された節を子、上位に接続された節を親といい、木構造の頂点にあるもっとも上位の節を根といいます。

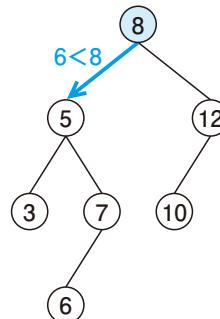
木構造のうち、2つ以上の子をもたないものをとくに二分木といいます。二分探索木は、二分木の各節に値を格納する際、「左の子の値は親の値より小さく、右の子の値は親の値より大きい」というルールを設けた二分木です。これにより、値の探索を効率的に行うことができます。



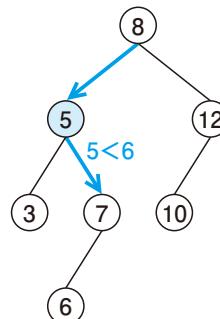
二分探索木から値を検索する

例として、上の二分探索木から値 6 を探索する手順を説明しましょう。

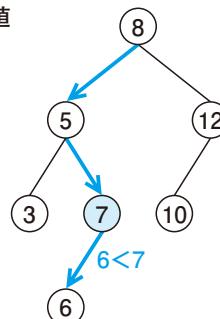
- ①二分探索木の根から出発します。 $6 < 8$ なので、目的の値は 8 の左側の部分木にあると考えられます。



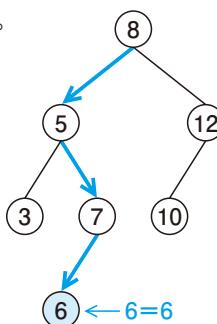
- ②8 の左の子の値は 5 です。 $5 < 6$ なので、目的の値は 5 の右側の部分木にあると考えられます。



- ③5 の右の子の値は 7 です。 $6 < 7$ なので、目的の値は 7 の左側の部分木にあると考えられます。



④7の左の子の値は目的の値6なので、探索成功です。



問題解説

空欄a：空欄の条件式が真のとき、変数currは変数currの左の子に設定されます。この処理は、目的の値がcurrの節の値より小さかった場合に実行します。したがって**空欄a**の条件式は「key が curr.val より小さい」です。

04	if (key が curr.val より小さい)
05	curr ← curr.left

↑ 空欄a

空欄b：行番号04の条件式が偽のときは、else以下の処理が実行されます。行番号04の条件式が偽になるのは「key が curr.val より大きい」場合です。この場合、目的の値は右側にあると考えられるので、変数currを右の子に設定します。

06	else
07	curr ← curr.right ← 空欄b

以上から、**空欄a**が「key が curr.val より小さい」、**空欄b**が「curr.right」のアが正解です。

解答

問08 ア



合格のカギ

二分探索木と二分探索法

二分探索木では、節が左右にバランスよく配置されていれば、1回の探索で探索範囲を半分にせばめることができます。この方式は、整列済みの配列から値を探査する二分探索法とよく似ています。二分探索法も基本的なアルゴリズムなので、内容を理解しておきましょう。



問 09

次の記述中の [] に入る正しい答えを、解答群の中から選べ。

4つの頂点をもつ無向グラフの隣接行列を、4行4列の二次元配列 `graph` で表現する。手続 `addEdge` は、この無向グラフに辺を追加する手続である。

```
addEdge(1, 2)
addEdge(1, 3)
addEdge(2, 4)
addEdge(3, 4)
addEdge(2, 3)
```

を実行したとき、二次元配列 `graph` が表すグラフは [] である。

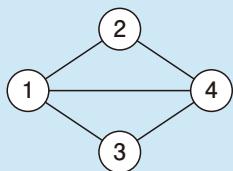
(プログラム)

```
大域： 整数型の二次元配列： graph ← {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}}
```

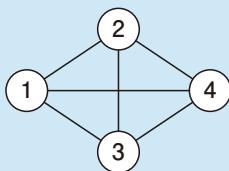
```
○ addEdge( 整数型 :t1, 整数型 :t2)
    graph[t1, t2] ← 1
    graph[t2, t1] ← 1
```

解答群

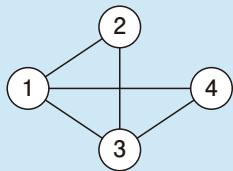
ア



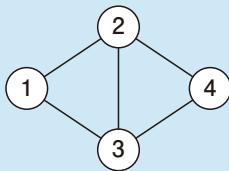
イ



ウ



エ



※オリジナル問題

プログラムの解説

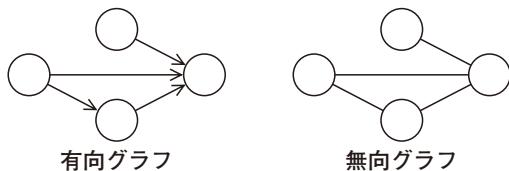
```

01 大域： 整数型の二次元配列： graph ← {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0},
                                         {0, 0, 0, 0}}
02 ○ addEdge( 整数型 :t1, 整数型 :t2)
03     graph[t1, t2] ← 1
04     graph[t2, t1] ← 1

```

グラフ

グラフは、複数の頂点を辺で結んだデータ構造です。2つの頂点を結ぶ辺（エッジ）に、始点と終点の区別がある場合を**有向グラフ**、始点と終点の区別がない場合を**無向グラフ**といいます。

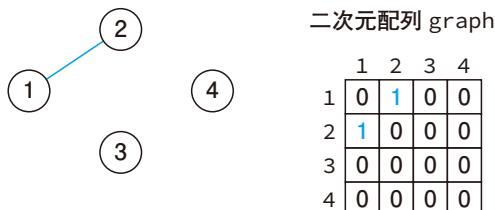


隣接行列

隣接行列は、グラフの頂点と頂点を結ぶ辺を表現する方法のひとつです。頂点の数をnとすると、隣接行列は、 $n \times n$ の二次元配列になります。

本問では、graphという 4×4 の二次元配列を隣接行列として使っています。たとえば、頂点1と頂点2の間を辺で結ぶ場合は、graph[1, 2]とgraph[2, 1]を1に設定します。

無向グラフでは、頂点1→頂点2と、頂点2→頂点1の両方向をつなげるので、上のようにgraph[1, 2]とgraph[2, 1]の両方を1に設定します。有向グラフの場合は、頂点1→頂点2ならgraph[1, 2]、頂点2→頂点1ならgraph[2, 1]だけを1に設定します。



問題解説

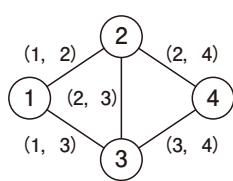
手続addEdgeは、2つの頂点の番号を引数に指定すると、その頂点の間を辺で結びます。本問では、手続addEdgeを次のように実行します。

```

addEdge(1, 2) ← 頂点1と頂点2を結ぶ
addEdge(1, 3) ← 頂点1と頂点3を結ぶ
addEdge(2, 4) ← 頂点2と頂点4を結ぶ
addEdge(3, 4) ← 頂点3と頂点4を結ぶ
addEdge(2, 3) ← 頂点2と頂点3を結ぶ

```

この結果できるグラフは、次のようにになります。正解は **エ** です。



二次元配列 graph

	1	2	3	4
1	0	1	1	0
2	1	0	1	1
3	1	1	0	1
4	0	1	1	0

○ 解答 ○

問09 **エ**



10

次の記述中の [] に入る正しい答えを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まる。

手続 `qsort` は、大域変数の配列 `data` に格納されたデータを、クイックソートを用いて昇順に整列する手続である。大域変数 `data` が

大域： 整数型の配列： `data` $\leftarrow \{2, 4, 5, 1, 3\}$

のように宣言されているとき、手続 `qsort` を `qsort(1, 5)` のように呼び出した。プログラムの α の行が 1 回目に実行された直後の配列 `data` の内容は、[] となる。

[プログラム]

```
○ 整数型： qsort( 整数型： left, 整数型： right )
    整数型： pivot, i, j
    if (left < right)
        pivot  $\leftarrow$  data[right]
        i  $\leftarrow$  left
        j  $\leftarrow$  right - 1
        do
            while (data[i] < pivot)
                i  $\leftarrow$  i + 1
            endwhile
            while (data[j] > pivot)
                j  $\leftarrow$  j - 1
            endwhile
            if (i \geq j)
                break /* ループから抜ける */
            endif
            data[i] の値 と data[j] の値を交換する
            i  $\leftarrow$  i + 1
            j  $\leftarrow$  j - 1
        while (true)
            data[right] の値 と data[i] の値を交換する   $\alpha$ 
            qsort(left, i - 1)
            qsort(i + 1, right)
        endif
```

解答群

- ア {2, 1, 5, 4, 3}
 イ {2, 3, 5, 1, 4}
 ウ {2, 1, 3, 4, 5}
 エ {2, 1, 5, 3, 4}
 オ {1, 2, 3, 4, 5}
 カ {1, 3, 2, 5, 4}

※オリジナル問題

プログラムの解説

```

01 ○ 整数型 : qsort( 整数型 : left, 整数型 : right)
02 整数型 : pivot, i, j
03 if (left < right)
04     pivot ← data[right] ←———— 基準値を設定
05     i ← left
06     j ← right - 1
07     do
08         while (data[i] < pivot) ←———— 左側に基準値より小さい値
09             i ← i + 1
10         endwhile
11         while (data[j] > pivot) ←———— 右側に基準値より大きい値
12             j ← j - 1
13         endwhile
14         if (i ≥ j)
15             break /* ループから抜ける */
16         endif
17         data[i] の値 と data[j] の値を交換する ←———— 左側にある基準値より大きい要素と
18             右側にある基準値より小さい要素を
19             交換
20         while (true)
21             data[right] の値 と data[i] の値を交換する ←———— α
22             qsort(left, i - 1)
23             qsort(i + 1, right)
24         endif

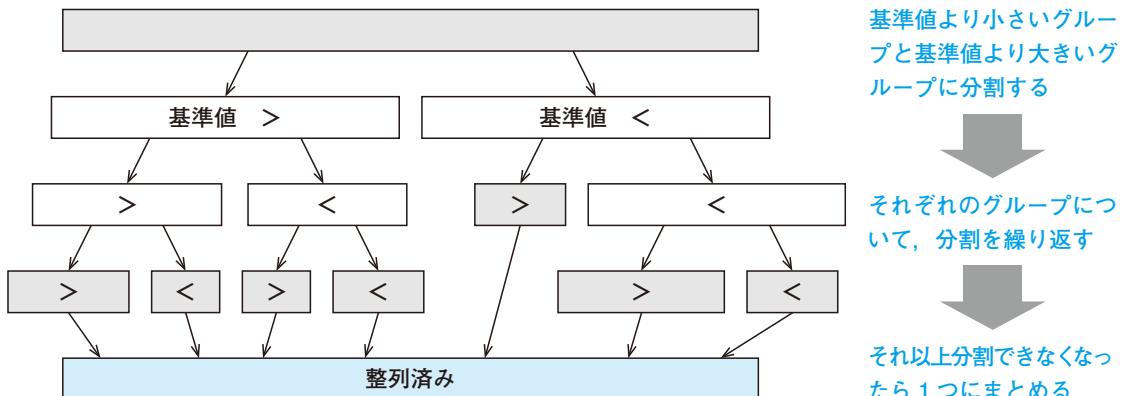
```

| クイックソート

データを昇順や降順に並べ替える整列アルゴリズムには、いくつかの種類があります。クイックソートは、その中でも効率が良く、実際によく使われているアルゴリズムのひとつです。

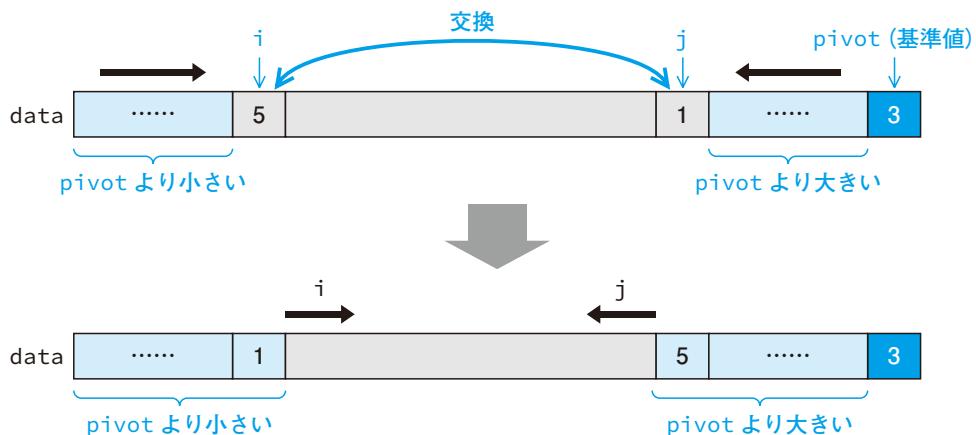
クイックソートのプログラムはやや複雑ですが、考え方自体はそれほど難しいものではありません。まず、配列から要素を1つ選び、その値を基準値とします。次に、基準値より小さいデータを配列の左側に、基準値より大きいデータを配列の右側に振り分けます。そして、配列の左側と右側とを、それぞれクイックソートを用いて整列します。

これ以上分割できなくなるまでこの作業を繰り返せば、配列全体が整列されます。

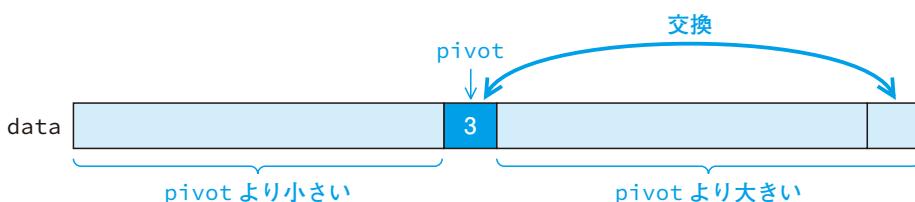


次に、プログラムをもう少し細かくみていきましょう。このプログラムでは、配列の右端の要素 `data[right]` の値を基準値 `pivot` とします（行番号 04）。

次に、配列を左端から中央に向かって順に調べ、値が `pivot` より大きい要素を見つけます（行番号 08 ~ 10）。一方、右端から中央に向かっても調べて、値が `pivot` より小さい要素を見つけます（行番号 11 ~ 13）。そして、見つかった 2 つの要素の値を交換します。



左側の i の位置と右側の j の位置が交差するまでこの作業を続けると、`pivot` より値が小さい要素は左側へ、大きい要素は右側に振り分けられます。最後に、基準値 `data[right]` を真ん中に置けば振り分け完成です（行番号 21）。



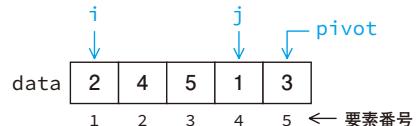
この後、振り分けた2つの部分に対して、それぞれクイックソートを実行します（行番号22, 23）。このプログラムでは、手続 `qsort` を再帰的に呼び出してこの作業を行っています。

問題解説

プログラムを順に追いながら、配列 `data` の内容の変化をトレースしていきましょう。

行番号04～06：手続 `qsort(1, 5)` を呼び出すと、引数 `left` には1、引数 `right` には5が設定されます。行番号04～06で、変数 `pivot` には `data[right]` の値3が、変数 `i` と `j` にはそれぞれ1と4が設定されます。

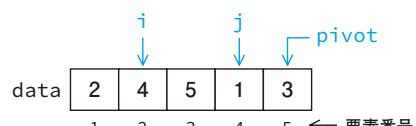
```
04  pivot ← data[right]
05  i ← left
06  j ← right - 1
```



行番号08～13：配列を左端から順に調べて、値が `pivot` 以上の要素を探します。`data[2]` の値が3より大きいので、`i` の値は2になります。

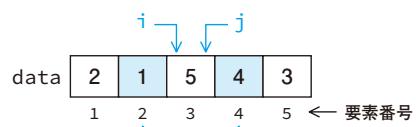
次に配列を右端 (`data[5]` は除く) から順に調べて、値が `pivot` 以下の要素を探します。`data[4]` の値が3より小さいので、`j` の値は4になります。

```
08  while (data[i] < pivot)
    i ← i + 1
10 endwhile
11 while (data[j] > pivot)
12     j ← j - 1
13 endwhile
```



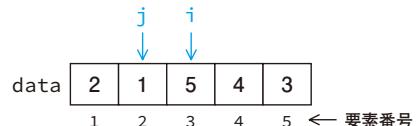
行番号17～20：`data[i]` の値と `data[j]` の値を交換します。次に、`i` と `j` の値を1ずつずすめ、行番号08に戻ります。

```
17  data[i] の値と data[j] の値を交換する
18  i ← i + 1
19  j ← j - 1
20  while (true)
```



行番号08～13：`data[i]` は3より大きいので、`i` の値は3で止まります。一方、`j` の値は2にすすみます。

```
08  while (data[i] < pivot)
    i ← i + 1
10 endwhile
11 while (data[j] > pivot)
12     j ← j - 1
13 endwhile
```

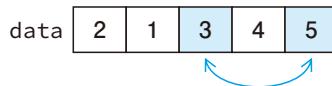


行番号 14～16： i のほうが j より大きくなつたので、繰返し処理を抜けます。

```
14 if (i ≥ j)  
15     break /* ループから抜ける */  
16 endif
```

行番号 21：`data[right]` の値と `data[i]` の値を交換します。配列 `data` は次のようにになります。

21 `data[right]` の値 と `data[i]` の値を交換する $\leftarrow \alpha$



以上から、1回目の α の行の実行が終わると、配列 `data` の内容は {2, 1, 3, 4, 5} になっています。正解は **ウ** です。

解答

問10 **ウ**



合格のカギ クイックソート

クイックソートについては、まず考え方をきちんと理解しておくことが重要です。クイックソートのように、問題を細かく分割し、分割したそれぞれを解決することで全体を解決する手法を分割統治法といいます。



11

次のプログラム中の と に入れる正しい答えの組合せを解答群の中から選べ。なお、配列の要素番号は 1 から始まる。

関数 `stdev` は、要素数が 1 以上の配列 `dataArray` を引数として受け取り、要素の標準偏差を戻り値として返す。ここで、実数型の配列 `dataArray` の各要素の値を x_1, x_2, \dots, x_n (n は `dataArray` の要素数) とすると、標準偏差は次の式で求めることができる。

$$\text{標準偏差} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2 - \left(\frac{1}{n} \sum_{i=1}^n x_i \right)^2}$$

〔プログラム〕

```

○ 実数型： stdev( 実数型の配列： dataArray )
    実数型： sum, sum2, var
    整数型： i, n
    n  $\leftarrow$  dataArray の要素数
    sum  $\leftarrow$  0
    sum2  $\leftarrow$  0
    for ( i を 1 から n まで 1 ずつ増やす )
        sum  $\leftarrow$  sum + dataArray[i]
        sum2  $\leftarrow$  sum2 + 
    endfor
    var  $\leftarrow$  (sum2  $\div$  n) - 
    return ( var の平方根 )

```

解答群

	<code>a</code>	<code>b</code>
ア	<code>dataArray</code> [<code>i</code>] \times 2	(<code>sum</code> の 2乗) \div <code>n</code>
イ	<code>dataArray</code> [<code>i</code>] \times 2	(<code>sum</code> \div <code>n</code>) の 2乗
ウ	<code>dataArray</code> [<code>i</code>] の 2乗	(<code>sum</code> の 2乗) \div <code>n</code>
エ	<code>dataArray</code> [<code>i</code>] の 2乗	(<code>sum</code> \div <code>n</code>) の 2乗

※オリジナル問題

プログラムの解説

```

01 ○ 実数型： stdev( 実数型の配列： dataArray )
02     実数型： sum, sum2, var
03     整数型： i, n
04     n  $\leftarrow$  dataArray の要素数
05     sum  $\leftarrow$  0
06     sum2  $\leftarrow$  0
07     for ( i を 1 から n まで 1 ずつ増やす )
08         sum  $\leftarrow$  sum + dataArray[i]   ← 合計値を求める
09         sum2  $\leftarrow$  sum2 +            ← 二乗和を求める
10     endfor
11     var  $\leftarrow$  (sum2  $\div$  n) -    ← 分散を求める
12     return ( var の平方根 )

```

分散と標準偏差

分散と標準偏差は、どちらもデータの散らばり具合を表す数値です。個々の要素と平均値との差を偏差といい、各要素の偏差の2乗を合計したものを平方和といいます。平方和を要素の個数で割ったものが分散です。

つまり分散とは、各要素の偏差の2乗の平均値です。ただし、この方法で分散を求めるには、まず各要素の平均値を求め、次に平方和を求めるという手順になるため、繰返し処理が2回必要になります。本問のプログラムは、繰返し処理を1回で済ますため、次のような分散を求める式を使っています。

$$\text{分散} = \frac{1}{n} \sum_{i=1}^n (x_i)^2 - \left(\frac{1}{n} \sum_{i=1}^n x_i \right)^2$$

プログラムは、変数 `sum` に個々の要素の合計 ($\sum_{i=1}^n x_i$) を、変数 `sum2` に個々の要素の2乗の合計 ($\sum_{i=1}^n (x_i)^2$) を格納します。行番号 07～10 の繰返し処理で `sum` と `sum2` の値を計算したら、行番号 11 でこれらを使って分散を計算し、変数 `var` に格納します。

標準偏差は分散の平方根ですから、行番号 12 で `var` の平方根を計算し、その値を戻り値とします。

問題解説

空欄 a：行番号 07～10 の繰返し処理で、`sum`、`sum2` という2つの項目を集計しています。`sum` には `dataArray[i]` の各要素の合計、`sum2` には各要素の2乗の合計を格納します。したがって行番号 09 の処理は次のようにになります。

09 `sum2 ← sum2 + dataArray[i] の2乗`
 ↑
 空欄 a

空欄 b：問題文の計算式を参考にして、分散を計算する式を組み立てます。

$\frac{1}{n} \sum_{i=1}^n (x_i)^2$ は、各要素の2乗の合計を `dataArray` の要素数 `n` で割ったものなので、「`sum2 ÷ n`」と書けます。また、 $\left(\frac{1}{n} \sum_{i=1}^n x_i \right)^2$ は、各要素の合計を `n` で割ったものの2乗なので、「`(sum ÷ n) の2乗`」と書けます。以上から、行番号 11 の処理は次のようにになります。

11 `var ← (sum2 ÷ n) - (sum ÷ n) の2乗`
 ↑
 空欄 b

空欄 a が「`dataArray[i] の2乗`」、空欄 b が「`(sum ÷ n) の2乗`」なので、正解は **I** です。

○ 解答 ○
問11 **I**



合格の力ギ

数式をプログラムにする

数学などで用いる数式は、ほぼそのままプログラムの計算式になります。ただし、数式の総和記号 Σ は、プログラムでは繰返し処理で値を求めます。数式の各項が、プログラムのどの変数に対応するかを確認しましょう。



12

次のプログラム中の と に入る正しい答えの組合せを解答群の中から選べ。ここで、要素番号は 1 から始まるものとする。

関数 `searchMatrix` は、整列済みの二次元配列から、引数に指定した値を探索する。要素が見つかったときは、その行番号と列番号を整数型の配列で返す。見つからなかったときは `{0, 0}` を返す。

ここで、整列済みの二次元配列とは、次のような条件を満たす二次元配列とする。また、整列済みの二次元配列の例を図に示す。

①ある要素の右側にある同じ行の要素の値は、左側の要素の値以上である。

②ある要素の下側にある同じ列の要素の値は、上側の要素の値以上である。

```
matrix ← {{ 1, 2, 2, 4, 5, 5, 9},
           { 2, 2, 3, 5, 6, 7, 9},
           { 2, 4, 6, 7, 8, 10, 12},
           { 6, 8, 8, 10, 11, 12, 14},
           { 7, 9, 11, 13, 15, 15, 19}}
```

図 整列済みの二次元配列の例

〔プログラム〕

```
大域： 整数型の二次元配列： matrix /* 整列済みのデータが格納されている */
○ 整数型の配列： searchMatrix( 整数型： key )
    i ← 1
    j ← matrix の列数
    while ((i が matrix の行数 以下) and (j が 1 以上))
        if (matrix[i, j] が key より小さい)
            
        elseif (matrix[i, j] が key に等しい)
            return {i, j}
        else
            
    endwhile
    return {0, 0}
```

解答群

	a	b
ア	$i \leftarrow i + 1$	$j \leftarrow j + 1$
イ	$j \leftarrow j + 1$	$i \leftarrow i + 1$
ウ	$i \leftarrow i + 1$	$j \leftarrow j - 1$
エ	$j \leftarrow j - 1$	$i \leftarrow i + 1$
オ	$i \leftarrow i - 1$	$j \leftarrow j + 1$
カ	$j \leftarrow j + 1$	$i \leftarrow i - 1$
キ	$i \leftarrow i - 1$	$j \leftarrow j - 1$
ク	$j \leftarrow j - 1$	$i \leftarrow i - 1$

※オリジナル問題

プログラムの解説

```
01 大域 : 整数型の二次元配列 : matrix /* 整列済みのデータが格納されている */
02 ○ 整数型の配列 : searchMatrix( 整数型 : key )
03   i ← 1
04   j ← matrix の列数 } 右上から探索をスタートする
05   while ((i が matrix の行数 以下) and (j が 1 以上))
06     if (matrix[i, j] が key より小さい)
07       a
08     elseif (matrix[i, j] が key に等しい)
09       return {i, j}
10     else
11       b
12   endwhile
13   return {0, 0}
```

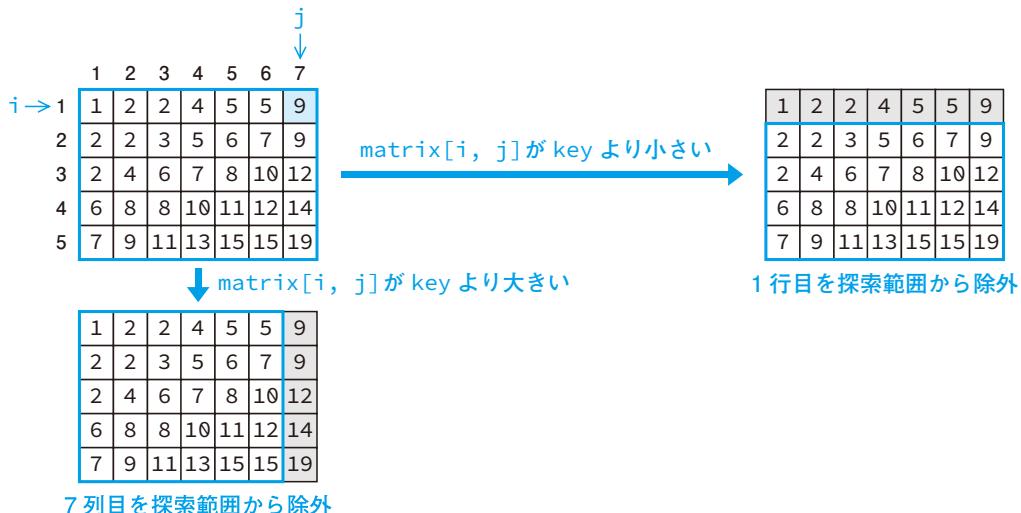
整列済み二次元配列の探索

次のような二次元配列 matrix を例に説明します。

```
matrix ← {{ 1, 2, 2, 4, 5, 5, 9},
           { 2, 2, 3, 5, 6, 7, 9},
           { 2, 4, 6, 7, 8, 10, 12},
           { 6, 8, 8, 10, 11, 12, 14},
           { 7, 9, 11, 13, 15, 15, 19}}
```

目的の値を探索する出発点は、整列済みの二次元配列の右上の要素になります。この要素は 1 行目の右端にあるので、同じ行の中でもっとも大きく、また、同じ列の中でもっとも小さい値です。

したがって、この値が目的の値 key より小さいなら、目的の値は同じ行にはありません。また、この値が目的の値より大きいなら、目的の値は同じ列にはありません。



このように、1回の探索で、1行または1列を探索範囲から除外することができます。この作業を、目的の値が見つかるまで繰り返します。

問題解説

空欄 a：要素 `matrix[i, j]` が、目的の値より小さい場合、目的の値は *i* 行目には存在しないので、*i* の値を 1 増やし、次の行を探索します。

```
10     if (matrix[i, j] が key より小さい)
11         i ← i + 1 ← 空欄 a
```

空欄 b：要素 `matrix[i, j]` が、目的の値より大きい場合、目的の値は *j* 列目には存在しないので、*j* の値を 1 減らし、次の列を探索します。

```
14     else
15         j ← j - 1 ← 空欄 b
```

○ 解答 ○

問12 ウ

以上から、空欄 a が「`i ← i + 1`」、空欄 b が「`j ← j - 1`」の
ウが正解です。



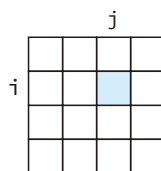
合格の力ギ

二次元配列と配列の配列

二次元配列は、一般的なプログラム言語では「配列の配列」として表しますが、情報処理試験の擬似言語では二次元配列と配列の配列を区別しています。二次元配列の要素は `matrix[i, j]` のように表しますが、配列の配列は `matrix[i][j]` となることに注意してください。また、擬似言語の配列の配列では、要素数が異なる配列の配列もつくれます。

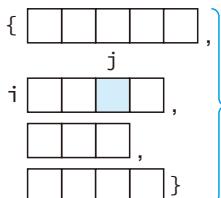
● 二次元配列

`matrix[i, j]`



● 配列の配列

`matrix[i][j]`



要素数が
異なって
いてもよい



13

次のプログラム中の と に入る正しい答えの組合せを解答群の中から選べ。

関数 str2num は、0～9 の数字から成る文字列を引数として受け取り、これを 10 進数の数値に変換して返す。プログラム中の“str の i 文字目の文字”は、文字列 str の先頭を 1 文字目とする i 文字目の文字を表す。また、“ch の ascii コード”は、文字データ ch に対応する ASCII コード（整数値）を表す。

(プログラム)

```

○ 整数型 : str2num( 文字列型 : str )
整数型 : ret, i, num
文字型 : ch
ret ← 0
for (i を 1 から str の文字数 まで 1 ずつ増やす)
    ch ← str の i 文字目の文字
    num ← (ch の ascii コード) - ()
    ret ← 
endfor
return ret

```

解答群

	a	b
ア	'0' の ascii コード	ret + num
イ	'0' の ascii コード	ret × 10 + num
ウ	'1' の ascii コード	ret + num
エ	'1' の ascii コード	ret × 10 + num
オ	'9' の ascii コード	ret + num
カ	'9' の ascii コード	ret × 10 + num

※オリジナル問題

プログラムの解説

```

01 ○ 整数型 : str2num( 文字列型 : str )
02 整数型 : ret, i, num
03 文字型 : ch
04 ret ← 0
05 for (i を 1 から str の文字数 まで 1 ずつ増やす)
06     ch ← str の i 文字目の文字
07     num ← (ch の ascii コード) - ()
08     ret ← 
09 endfor
10 return ret

```

ASCII コード

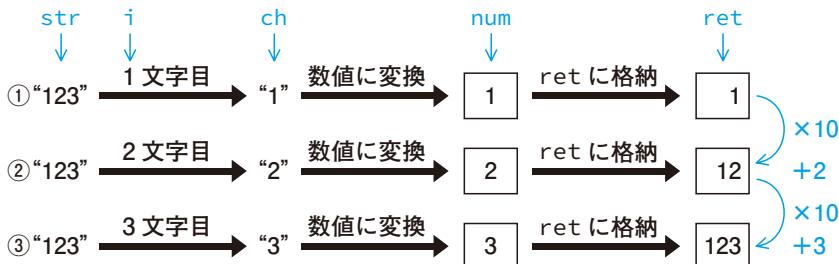
アルファベットや数字などの英数字には、**ASCII コード**と呼ばれる 7 ビットの数値が割り当てられています。ASCII コードは、数字の '0' が 48, '1' が 49, '2' が 50, … のように順番に割り当てられているので、たとえば、

('1' の ascii コード) – ('0' の ascii コード)

は $49 - 48 = 1$ となります。プログラムは、これをを利用して数字を数値に変換します。

問題解説

プログラムは、数字から成る文字列を 1 文字ずつ取り出して数値に変換し、変数 `ret` に格納します。たとえば、文字列 “123” を数値データに変換する手順は、次のようにになります。



空欄 a：文字 `ch` を、1 衝の数値データに変換して変数 `num` に格納します。`ch` の ASCII コードから数字 '0' の ASCII コードを引くと、数字に対応する 1 衝の数値が得られます。

07 `num ← (ch の ascii コード) – ('0' の ascii コード)`

↑ 空欄 a

空欄 b：変換した 1 衝の数値 `num` を、変数 `ret` に格納します。その際、すでに格納されている数値は 10 倍して、位を 1 つ上げます。行番号 08 の処理は次のようにになります。

08 `ret ← ret × 10 + num`

↑ 空欄 b

○ 解答 ○

問13 イ

合格のカギ 文字列と整数の変換

関数 `str2num` の逆で、数値データを文字列に変換する関数 `num2str` は、たとえば次のようにになります。

○ 文字列型： `num2str(整数型 : num)`

文字列型の配列： `array ← {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9"}`

整数型： `n ← num`

文字列型： `str ← ""`

```

if (n が 0 に等しい)
    return "0"
endif
while (n が 0 より大きい)
    str ← array[(n mod 10) + 1] + str
    n ← n ÷ 10 の商
endwhile
return str
  
```



14

次の記述中の [] に入る正しい答えを解答群の中から選べ。ここで、配列の要素番号は 1 から始まるものとする。

手続 insertHash は、引数に指定された数値を、整数型の配列 hashArray に格納する手続である。
hashArray の要素数は 11 で、初期値はすべての要素が未定義であるものとする。この手続を、

```
insertHash(3) → insertHash(23) → insertHash(14) → insertHash(12)  
→ insertHash(15)
```

の順に実行した直後の配列 hashArray の内容は、 [] となる。

[プログラム]

```
大域： 整数型の配列： hashArray[11] /* 要素の初期値は未定義とする */
```

```
○ insertHash( 整数型 : key )  
    hash ← (key mod hashArray の要素数) + 1  
    step ← 5 - (key mod 5)  
    while (hashArray[hash] が 未定義 でない )  
        hash ← hash + step  
        hash ← ((hash - 1) mod hashArray の要素数) + 1  
    endwhile  
    hashArray[hash] ← key
```

解答群

- ア { 未定義 , 23 , 未定義 , 14 , 3 , 未定義 , 未定義 , 15 , 未定義 , 12 , 未定義 }
- イ {12 , 未定義 , 未定義 , 15 , 未定義 , 14 , 未定義 , 3 , 23 , 未定義 , 未定義 }
- ウ {3 , 12 , 未定義 , 14 , 未定義 , 15 , 未定義 , 未定義 , 未定義 , 未定義 , 23}
- エ { 未定義 , 未定義 , 12 , 未定義 , 3 , 未定義 , 23 , 15 , 14 , 未定義 , 未定義 }
- オ { 未定義 , 23 , 未定義 , 3 , 14 , 未定義 , 未定義 , 12 , 未定義 , 15 , 未定義 }

※オリジナル問題

```

01 大域： 整数型の配列： hashArray[11] /* 要素の初期値は未定義とする */

02 ○insertHash( 整数型 : key )
03   hash ← (key mod hashArray の要素数) + 1 ← ハッシュ値を求める
04   step ← 5 - (key mod 5)
05   while (hashArray[hash] が 未定義 でない) ← 衝突が発生した場合
06     hash ← hash + step
07     hash ← ((hash - 1) mod hashArray の要素数) + 1 ← ラップアラウンド
08   endwhile
09   hashArray[hash] ← key ←———— ハッシュテーブルに値を格納

```

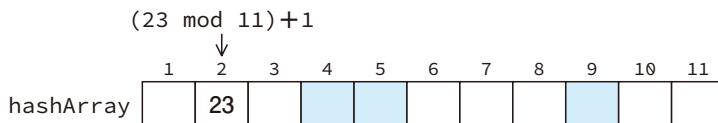
ハッシュテーブル

ハッシュテーブルとは：値の格納位置を、その値から算出したハッシュ値によって決めるデータ構造をハッシュテーブルといいます。手続 `insertHash` は、引数に指定した値から格納位置を算出し、配列 `hashArray` に値を格納します。

格納位置の求め方にはいくつかの手法がありますが、このプログラムでは行番号 03 で

```
hash ← (key mod hashArray の要素数) + 1
```

という計算を行い、格納位置を変数 `hash` に設定します。たとえば、値「23」を格納する場合、変数 `hash` の値は $(23 \bmod 11) + 1 = 2$ となるので、配列 `hashArray` の要素番号 2 に値「23」を格納します。



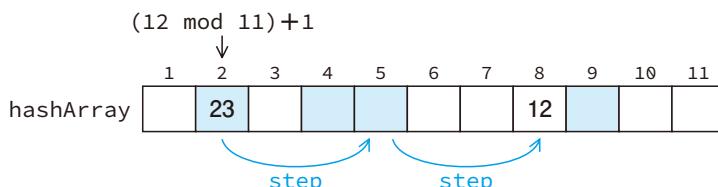
衝突が発生した場合：値によっては、算出される格納位置が同じになってしまう場合があります。このような場合を衝突といいます。

衝突が発生した場合は、一定の方法で空いている場所を探して、その位置に値を格納します。このプログラムでは、行番号 04 で

```
step ← 5 - (key mod 5)
```

という計算を行い、変数 `hash` の位置にすでに値が格納されている場合には、`step` の値だけ先の位置を新たに格納位置とします。もし、その位置も空いていなかった場合は、さらに `step` の値だけ先を探します。空いている場所が見つかるまでこの処理を繰り返します（行番号 05～08）。

たとえば、値「12」の格納位置は $(12 \bmod 11) + 1 = 2$ となり、値「23」の格納位置と同じになります。要素番号 2 に値が格納されている場合は、そこから $5 - (12 \bmod 5) = 3$ だけ先の要素番号 5 を格納位置とします。そもそもふさがっている場合は、さらに 3 つ先の要素番号 8、そもそもふさがっている場合は、さらに 3 つ先の要素番号 11 を探します。



なお、配列 hashArray の要素数は 11 なので、要素番号 11 より先には格納できません。その場合は先頭に戻って格納場所を探します（ラップアラウンド）。行番号 07 の

```
hash ← ((hash - 1) mod hashArray の要素数) + 1
```

はやや複雑な式ですが、要するに hash の値が hashArray の要素数を超えたたら、先頭に戻るという計算をしています（hash の値が 12 のとき、この式の計算結果が 1 になることを確認してください）。

問題解説

手続 insertHash は、引数に指定された値から、格納位置を

```
hash ← (key mod hashArray の要素数) + 1
```

で計算します。もし、その位置にすでに値が格納されていた場合は、

```
step ← 5 - (key mod 5)
```

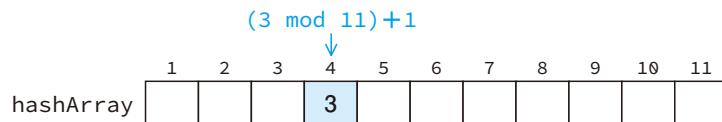
で計算した数ずつ格納位置をずらし、空いている場所を探します。

この手続を、

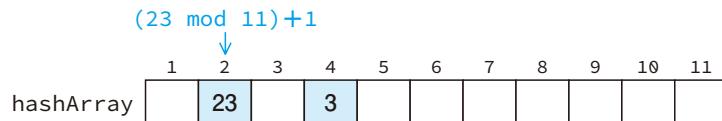
```
insertHash(3) → insertHash(23) →  
insertHash(14) → insertHash(12) →  
insertHash(15)
```

の順に実行すると、配列 hashArray の内容は次のようにになります。

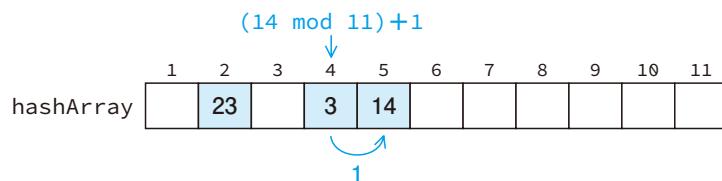
① insertHash(3) : $(3 \bmod 11) + 1 = 4$ が格納位置になります。



② insertHash(23) : $(23 \bmod 11) + 1 = 2$ が格納位置になります。

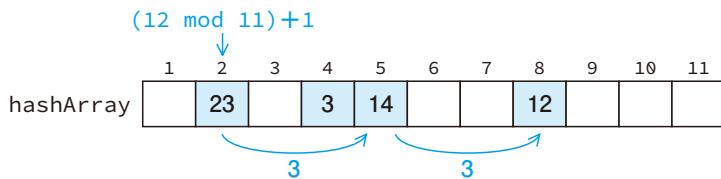


③ insertHash(14) : $(14 \bmod 11) + 1 = 4$ はふさがっているので、 $5 - (14 \bmod 5) = 1$ ずらし、5 が格納位置になります。

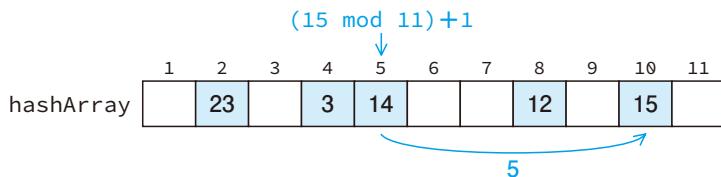


④ insertHash(12) : $(12 \bmod 11) + 1 = 2$ はふさがっているので、 $5 - (12 \bmod 5) = 3$ ずらし

ます。5もふさがっているので、さらに3ずらし、8が格納位置になります。



- ⑤ `insertHash(15)` : $(15 \bmod 11) + 1 = 5$ はふさがっているので、 $5 - (15 \bmod 5) = 5$ ずらし、10が格納位置になります。



以上から、`hashArray` の内容は { 未定義 , 23, 未定義 , 3, 14, 未定義 , 未定義 , 12, 未定義 , 15, 未定義 } となります。正解は **才** です。

○ 解答

問14 **才**



合格のカギ

ハッシュテーブル

ハッシュテーブルは、値から直接格納位置を計算できるので、探索がすばやくできるのが特徴です。また、キー値と値をペアで格納するマップにもハッシュテーブルが使われます。

なお、本問では値をハッシュテーブルの要素数で割った余りをハッシュ値としていますが、値の各桁を足し合わせ、その合計を要素数で割った余りとする方法もよく使われます。問題文を確認しましょう。



問

15

次のプログラム中の と に入る正しい答えの組合せを解答群の中から選べ。

手続 `outLines` は、引数に指定した文字列を、指定した桁数で改行して出力する手続である。

`outLines("abcdefghijklm", 5)` の実行結果を図に示す。

```
abcde  
fghij  
k
```

図 `outLines("abcdefghijklm", 5)` の実行結果

本問において、文字列に対する演算子 “+” は、文字列の連結を表す。

(プログラム)

```
○ outLines( 文字列型 : text, 整数型 : col )  
    文字列型 : buf  
    整数型 : cnt, i  
    buf ← ""  
    cnt ← 0  
    for (i を 1 から text の文字数 まで 1 ずつ増やす)  
        buf ← buf + text の i 文字目から成る文字  
        cnt ← cnt + 1  
        if ()  
            buf を出力して改行する  
            buf ← ""  
            cnt ← 0  
        endif  
    endfor  
    if ()  
        buf を出力して改行する  
    endif
```

解答群

	a	b
ア	cnt が 0 に等しい	cnt が cols に等しい
イ	cnt が 0 に等しい	cnt が cols より大きい
ウ	cnt が 0 より大きい	cnt が cols に等しい
エ	cnt が 0 より大きい	cnt が cols より大きい
オ	cnt が cols に等しい	cnt が 0 に等しい
カ	cnt が cols に等しい	cnt が 0 より大きい
キ	cnt が cols より大きい	cnt が 0 に等しい
ク	cnt が cols より大きい	cnt が 0 より大きい

※オリジナル問題

プログラムの解説

```

01 ○ outLines( 文字列型 : text, 整数型 : col)
02   文字列型 : buf
03   整数型 : cnt, i
04   buf ← ""
05   cnt ← 0
06   for (i を 1 から text の文字数 まで 1 ずつ増やす )
07     buf ← buf + text の i 文字目から成る文字列 ← 1 行分の文字列を変数 buf に
08     cnt ← cnt + 1                                     格納
09     if ( [ a ] )
10       buf を出力して改行する ← 1 行出力
11     buf ← ""
12     cnt ← 0
13   endif
14 endfor
15   if ( [ b ] )
16     buf を出力して改行する ← 最後の行を出力
17   endif

```

手続 `outLines` は、引数 `text` に指定した文字列を、指定した行数 `cols` ごとに改行して出力します。プログラムは、`text` から文字を 1 文字ずつ読み込み、変数 `buf` に格納していきます（行番号 07）。`buf` の長さが指定した行数になったら、`buf` の内容を出力して改行し、`buf` を空にします（行番号 09～13）。この処理を、`text` の末尾の文字まで繰り返します。

繰返し処理を抜けたら、`buf` に残っている文字列を出力して終了です。

問題解説

空欄 a：空欄の条件式が成立すると、`buf` の内容が出力されます。これは、`buf` の長さが指定した行数に

なった場合の処理です。

プログラムは、bufに1文字格納するたびに、変数cntの値を1増やしているので、bufの長さは変数cntに設定されます。cntの値が桁数colsになつたらbufの内容を出力します。したがって、空欄の条件式は次のようにになります。

09

if (cnt が cols に等しい)
↑
— 空欄 a

空欄 b：繰返し処理を抜けても、変数bufにはまだ出力されていない文字列が残っている可能性があります。行番号15～17では、これを確認して出力します。

bufの長さは変数cntに設定されるので、変数bufに出力されていない文字列が残っている場合は、変数cntの値が0より大きくなります。したがって、空欄の条件式は次のようにになります。

15

if (cnt が 0 より大きい)
↑
— 空欄 b

以上から、空欄aが「cnt が cols に等しい」、空欄bが「cnt が 0 より大きい」の**力**が正解です。

○ 解答 ○

問15 力



合格の力ギ

条件式の書き方は複数ある

空欄aに入る条件式「cnt が cols に等しい」は、「cnt が cols 以上」でも正しく動作します。同様に、空欄bの条件式「cnt が 0 より大きい」は、「cnt が 0 でない」としても正しく動作します。

このように、プログラムの書き方は1通りとは限りません。正解と思ったものが解答群にない場合は、同じ働きをするものがどれかを考えましょう。



16

次のプログラム中の と に入る正しい答えの組合せを解答群の中から選べ。ここで、配列の要素番号は 1 から始まる。

ある会社の今月の売上個数を商品ごとに登録した配列 master と、1日分の売上個数を商品ごとに登録した配列 trans がある。手続 updateMaster は、配列 trans に登録されている商品の売上個数を、配列 master の該当する商品の売上個数に加算する手続である。

商品ごとの売上個数は、クラス Item によって表現する。クラス Item の説明を図に示す。配列 master 及び配列 trans には、クラス Item のインスタンスが商品コードの昇順に登録されている。配列 master に登録されていない商品コードの商品は、配列 trans にも登録されないものとする。

メンバ変数	型	説明
code	整数型	商品の商品コード。同じ商品コードのインスタンスは重複して登録されないものとする。
sales	整数型	商品の売上個数。

図 クラス Item の説明

[プログラム]

```

○ updateMaster(Item の配列 : master, Item の配列 : trans)
整数型 : i, j
i ← 1
j ← 1
while ()
    if (master[i].code が trans[j].code と等しい)
        master[i].sales ← master[i].sales + trans[j].sales
        i ← i + 1
        j ← j + 1
    elseif ()
        i ← i + 1
    else
        j ← j + 1
    endif
endwhile

```

解答群

	a	b
ア	($i \leq \text{master}$ の要素数) and ($j \leq \text{trans}$ の要素数)	$\text{master}[i].code > \text{trans}[j].code$
イ	($i \leq \text{master}$ の要素数) and ($j \leq \text{trans}$ の要素数)	$\text{master}[i].code < \text{trans}[j].code$
ウ	($i \leq \text{master}$ の要素数) or ($j \leq \text{trans}$ の要素数)	$\text{master}[i].code > \text{trans}[j].code$
エ	($i \leq \text{master}$ の要素数) or ($j \leq \text{trans}$ の要素数)	$\text{master}[i].code < \text{trans}[j].code$

※オリジナル問題

プログラムの解説

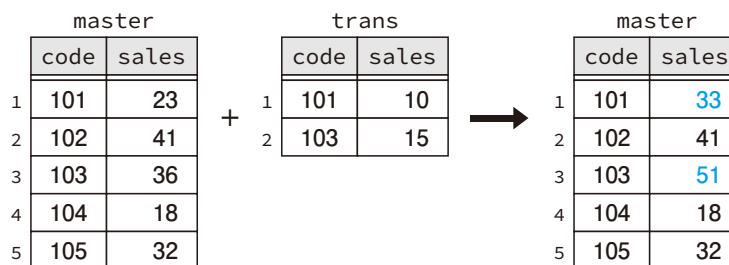
```

01 ○ updateMaster(Item の配列 : master, Item の配列 : trans)
02 整数型 : i, j
03   i ← 1
04   j ← 1
05   while ([ ] a)
06     if (master[i].code が trans[j].code と等しい)
07       master[i].sales ← master[i].sales + trans[j].sales
08     i ← i + 1
09     j ← j + 1
10   elseif ([ ] b)
11     i ← i + 1
12   else
13     j ← j + 1
14   endif
15 endwhile

```

【突合せ処理】

手続 `updateMaster` は、商品コードと売上個数の組を並べた 2 つの配列を読み込み、一方の配列に登録されているデータ (`trans`) を使って、もう一方の配列のデータ (`master`) を更新します。このような処理を **突合せ処理**といいます。



突合せ処理を行うには、2つの配列が同じキー（本問の場合は商品コード）で整列されていなければなりません。プログラムは、次のような手順でデータの更新を行います。

① master と trans の各要素の商品コードを、先頭から順に比較します（行番号 06）。以降は比較の結果によって、処理が②～④に分かれます。

② 両者の商品コードが一致する場合は、trans のデータを使って master のデータを更新します（行番号 07）。次に、比較する要素を、master と trans の両方とも 1つ先にすすめます（行番号 08, 09）。

master		trans	
code	sales	code	sales
101	33	101	10
102	41	103	15
103	36		
104	18		
105	32		

③ master の要素の商品コードのほうが、trans の商品コードより小さい場合、その商品は trans には登録されていません。この場合は、比較する要素を master のほうだけ 1つ先にすすめます。

master		trans	
code	sales	code	sales
101	33	101	10
102	41	103	15
103	36		
104	18		
105	32		

④ master の要素の商品コードのほうが、trans の商品コードより小さい場合、その商品は master に登録されていません（本問では、このような場合は生じないものとしています）。本問のプログラムではこの場合とくに何もせず、比較する要素を trans のほうだけ 1つ先にすすめます。

⑤ master か trans のどちらかの要素を末尾まで比較したら、処理を終了します。

問題解説

空欄 a：空欄には、繰返し処理を継続する条件式が入ります。行番号 06 では、

06	if (master[i].code が trans[j].code と等しい)
----	------------------------------------------

のように、2つの配列の要素を比較しています。したがって繰返しの間、配列 master の要素番号 i と配列 trans の要素番号 j が、配列の要素数を超えないようにしなければなりません。2つの条件は、それぞれ

i ≤ master の要素数

j ≤ trans の要素数

と書けます。2つの条件は、どちらか一方ではなく両方とも成立している必要があるので、2つの条件を and で結びます。

($i \leq \text{master}$ の要素数) and ($j \leq \text{trans}$ の要素数)

← 空欄 a

空欄 b：プログラムは、配列 `master` と配列 `trans` の各要素の商品コードを先頭から順に比較し、その結果によって次の3通りに処理を分岐します。

- 両者の商品コードが一致する場合：`trans` のデータを使って `master` のデータを更新します（行番号 07）。
- 空欄 b の条件が成立する場合： i を 1 増やし、配列 `master` の要素を 1 つ先にすすめます。
- その他の場合： j を 1 増やし、配列 `trans` の要素を 1 つ先にすすめます。

配列 `master` の要素を 1 つ先に進めるのは、`master` の要素の商品コードのほうが、`trans` の商品コードより小さい場合です。

master[i].code < trans[j].code	
code	sales
101	33
102	41
103	36
104	18
105	32

i → 1つ先にすすめる ↘

code	sales
101	10
103	15

j → ↗

したがって、**空欄 b** の条件式は次のようにになります。

`master[i].code < trans[j].code`

← 空欄 b

空欄 a が「($i \leq \text{master}$ の要素数) and ($j \leq \text{trans}$ の要素数)」、空欄 b が「`master[i].code < trans[j].code`」なので、正解は **イ** です。

解答

問 16 **イ**



合格の力ギ

突合せ処理のポイント

本問は2つの配列を比較していますが、突合せ処理は2つのファイルを比較する場合によく用いられます。2つのファイルが同じキー項目で整列済みであること、キー項目の比較結果に応じて、処理が3つに枝分かれすることを理解しておきましょう。



問

17 次の記述を読んで、設間に答えよ。

A 社は、自社の中継用メールサーバで、接続元 IP アドレス、電子メールの送信者のメールアドレスのドメイン名、及び電子メールの受信者のメールアドレスのドメイン名から成るログを取得している。ログの一部を図に示す。

項目番号	接続元 IP アドレス	送信者のメールアドレスのドメイン名	受信者のメールアドレスのドメイン名
1	AAA.168.1.5	a.b.c	a.b.d
2	AAA.168.1.10	a.b.c	a.b.c
3	BBB.45.67.90	a.b.c	a.b.d
4	AAA.168.1.10	a.b.d	a.b.c
5	BBB.45.67.89	a.b.d	a.b.e
6	BBB.45.67.90	a.b.d	a.b.c

注 1 : AAA で始まる IP アドレスは自社のグローバル IP アドレス、 BBB で始まる IP アドレスは社外のグローバル IP アドレスとする。

注 2 : a.b.c は自社のドメイン名とし、 a.b.d と a.b.e は他社のドメイン名とする。

図 中継用メールサーバのログ（抜粋）

A 社は、外部ネットワークからの第三者中継を遮断するため、以下のポリシーに基づいて中継を制限することにした。

- (一) 自社サイト内のからのメール転送（受信、中継）は、送信者のメールアドレスが自社の場合にのみ許可する。
- (二) 他社サイトからのメール転送（受信、中継）は、自社サイト宛しか受け取らない。
- (三) 他社サイトから他社サイトへのメール転送は拒否する。

設問 上記のポリシーに従う場合、遮断されるメール中継の項目番号はどれか。

解答群

- ア 項番 3, 5
- イ 項番 3, 6
- ウ 項番 5, 6
- エ 項番 3, 4, 5
- オ 項番 3, 5, 6
- カ 項番 4, 5, 6

※出典：平成 30 年度秋期基本情報技術者試験午前問題問 45 を改変

問題解説

第三者中継とは、不特定多数の第三者が、自社のメールサーバを勝手に利用して電子メールを送信することです。第三者中継はメールサーバの設定によって可能になりますが、迷惑メールの送信元などに悪用される可能性があるため、許可すべきではありません。

ポリシーの（一）～（三）の項目ごとに、どのようなメールが許可／制限されるかを確認しましょう。

（一）自社サイト内からのメール転送（受信、中継）は、送信者のメールアドレスが自社の場合のみ許可する。

接続元の IP アドレスが AAA で始まるアドレスで、送信者のメールアドレスのドメイン名が「a.b.c」のメールは、転送を許可します。

（二）他社サイトからのメール転送（受信、中継）は、自社サイト宛しか受け取らない。

接続元の IP アドレスが AAA で始まるアドレス以外の場合は、受信者のメールアドレスのドメイン名が「a.b.c」の場合にのみ、転送を許可します。

（三）他社サイトから他社サイトへのメール転送は拒否する。

接続元の IP アドレスが社外のアドレスで、送信者と受信者のメールアドレスのドメイン名がいずれも他社サイトの場合は、第三者中継とみなして転送を拒否します。

以上のポリシーにしたがって、メールログの項番 1～6 を検討しましょう。

項番 1：接続元 IP アドレスが自社、送信者のメールアドレスも自社なので、社内の送信者から社外に宛てたメールです。→**許可**

項番 2：接続元 IP アドレスが自社、送信者と受信者のメールアドレスも自社なので、社内の送信者から社内の受信者に宛てたメールです。→**許可**

項番 3：接続元 IP アドレスが社外、送信者のメールアドレスは自社で、受信者のメールアドレスは社外です。社員が社外からメールサーバにアクセスして、社外にメールを送った可能性がありますが、ポリシーはこのメール転送を許可していません。→**拒否**

項番 4：接続元 IP アドレスは自社ですが、送信者のメールアドレスが他社のドメインなので、メール転送は拒否されます。→**拒否**

項番 5：接続元 IP アドレスが社外、送信者と受信者のメールアドレスも社外なので、第三者中継のメールです。→**拒否**

項番 6：接続元 IP アドレスは社外ですが、受信者のメールアドレスが社内なので、社外の送信者から自社の受信者に宛てたメールです。→**許可**

以上から、転送を拒否するメールは項番 3、4、5 になります。正解は**工**です。

解答

問17 工



合格のカギ

情報セキュリティの問題は、問題文をよく読んで、設問の状況を把握することが重要です。



18 次の記述を読んで、設間に答えよ。

イベントの企画、運営業務を行うR社は、現行のオフィスが手狭になってきたため、オフィスを半年後に移転することが決定している。オフィス移転を機に、現行の情報システムを見直して新たな情報システムを構築することになった。各部門から出された要望を検討した結果、クラウドコンピューティングを利用した外部サービス（以下、クラウドサービス）の利用を検討することにした。

そこでまず、自社で運用している現行のメールサービスとグループウェアサービスについて、クラウドサービスに切り替えた場合の管理、運用面の課題について検討することにした。クラウドサービスには大きく分けてSaaS型、PaaS型、IaaS型の3種類があることから、それぞれの特徴を考慮して現行の自社運用との比較を行った。その結果を表に示す。

表 メールサービスとグループウェアサービスの管理と運用に関する比較

管理主体・管理内容	SaaS型	PaaS型	IaaS型	自社運用（現行）
ハードウェア・ネットワークの管理主体	事業者	事業者	事業者	自社
OS、ミドルウェアの管理主体	事業者	a	自社又は事業者	自社
アプリケーションの管理主体	事業者	b	自社	自社
迷惑メール対策、ウイルス対策の管理主体	自社又は事業者	自社又は事業者	自社	自社
自社の管理工数	小	中	大	大
エンドユーザーから見たサービスの稼働率	99.9%以上 (SLAに依存)	99.9%以上 (予想)	99.9%以上 (予想)	99.5%以上 (実績)

設問 表中のaとbに入れる正しい答えの組合せを、解答群の中から選べ。

解答群

	a	b
ア	事業者	事業者
イ	事業者	自社
ウ	自社	事業者
エ	自社	自社

※出典：平成24年度春期基本情報セキュリティスペシャリスト試験午後Ⅱ問題問2を改変

問題解説

クラウドサービスの種類として、SaaS、PaaS、IaaS の3種類の違いを理解しておきましょう。

SaaS (Software as a Service)	アプリケーションの機能を利用者に提供
PaaS (Platform as a Service)	アプリケーションの稼働環境を利用者に提供
IaaS (Infrastructure as a Service)	システムを稼働する回線やハードウェア環境のみを提供

SaaS では、事業者が開発したアプリケーションの機能をサービスとして利用者に提供するので、ハードウェアや OS、ミドルウェアはもちろん、アプリケーションの管理も事業者が行います。

PaaS では、アプリケーションの稼働環境だけを事業者が提供します。通常、ミドルウェアまでは事業者が管理しますが、アプリケーションの開発は利用者自身が行います。

IaaS では、事業者はハードウェアとネットワーク環境のみを提供し、それ以外はすべて利用者が行います。

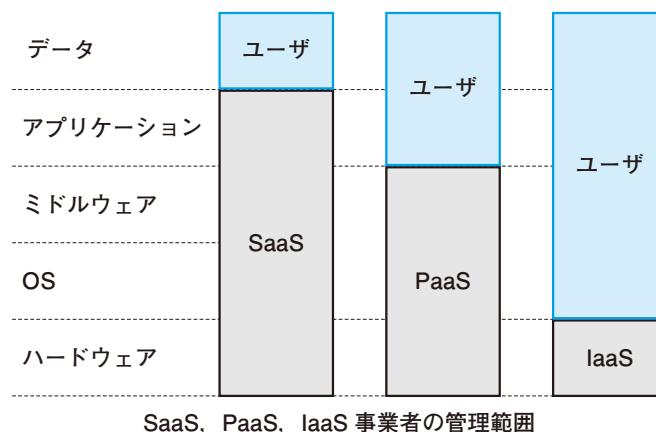
PaaS の場合、OS、ミドルウェアの管理主体は事業者ですが、アプリケーションの管理主体は自社になります。したがって、空欄 a が「事業者」、空欄 b が「自社」の **イ** が正解です。

解答
問18 **イ**



合格のカギ SaaS、PaaS、IaaS の違い

SaaS、PaaS、IaaS の違いを理解しておきましょう。





問 19

次の記述を読んで、設間に答えよ。

P社は、従業員数400名のIT関連製品の卸売会社であり、300社の販売代理店をもっている。P社では、販売代理店向けに、インターネット経由で商品情報の提供、見積書の作成を行う代理店サーバを運用している。また、従業員向けに、代理店ごとの卸価格や担当者の情報を管理する顧客サーバを運用している。代理店サーバ及び顧客サーバには、HTTP Over TLSでアクセスする。

P社のネットワーク構成を図1に示す。

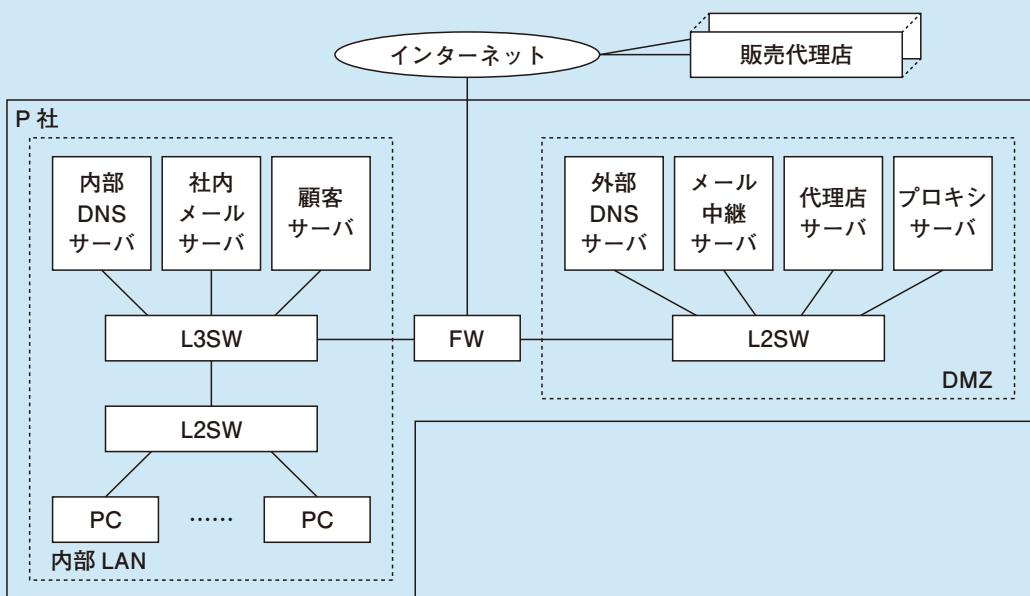


図1 P社のネットワーク構成

P社では、複数のサーバ、PC及びネットワーク機器を運用しており、それらには次のセキュリティ対策を実施している。

- a では、インターネットとDMZ間及び内部LANとDMZ間で業務に必要な通信だけを許可し、通信ログ及び遮断ログを取得する。
- b では、SPF (Sender Policy Framework) 機能によって送信元ドメイン認証を行い、送信元メールアドレスがなりすまされた電子メールを隔離する。
- c では、DMZのゾーン情報のほかに、キャッシュサーバの機能を稼働させており、■ c をDDoSの踏み台とする攻撃への対策を行う。
- d は、P社からインターネット上のWebサーバへのアクセスを中継し、通信ログを取得する。

設問 記述中の a ~ d に入れる正しい答えの組合せを、解答群から選べ。

解答群

	a	b	c	d
ア	FW	社内メールサーバ	内部 DNS サーバ	代理店サーバ
イ	FW	社内メールサーバ	プロキシサーバ	L3SW
ウ	FW	メール中継サーバ	外部 DNS サーバ	プロキシサーバ
エ	FW	メール中継サーバ	プロキシサーバ	L3SW
オ	L3SW	メール中継サーバ	外部 DNS サーバ	プロキシサーバ
カ	L3SW	社内メールサーバ	FW	外部 DNS サーバ
キ	L3SW	メール中継サーバ	プロキシサーバ	FW
ク	L3SW	社内メールサーバ	内部 DNS サーバ	外部 DNS サーバ

※出典：令和4年度秋期応用情報技術者試験午後問題問1を改変

問題解説

- 空欄 a では、インターネットと DMZ 間及び内部 LAN と DMZ 間で業務に必要な通信だけを許可し、通信ログ及び遮断ログを取得する。

空欄 a は、インターネットと DMZ 間、内部 LAN と DMZ 間の両方を中継する機器です。図のネットワーク構成から、そのような機器は「FW (ファイアウォール)」であることがわかります。ファイアウォールは、通過する通信の送信元や宛先、ポート番号などをチェックして、許可されてない通信を遮断します。

- 空欄 b では、SPF (Sender Policy Framework) 機能によって送信元ドメイン認証を行い、送信元メールアドレスがなりすまされた電子メールを隔離する。

SPF (Sender Policy Framework) は、送信元を偽った「なりすまし」のメールを防ぐための仕組みです。SPF に対応したメールサーバは、届いたメールが正当なメールサーバから送信されたものかどうかを、送信元の DNS サーバに問い合わせます。

P 社のネットワークにはメール中継サーバと社内メールサーバがありますが、社内メールサーバは社外のメールを中継しないので、SPF の機能は意味がありません。したがって、空欄 b には「メール中継サーバ」が入ります。

- 空欄 c では、DMZ のゾーン情報のほかに、キャッシュサーバの機能を稼働させており、DDoS の踏み台とする攻撃への対策を行う。

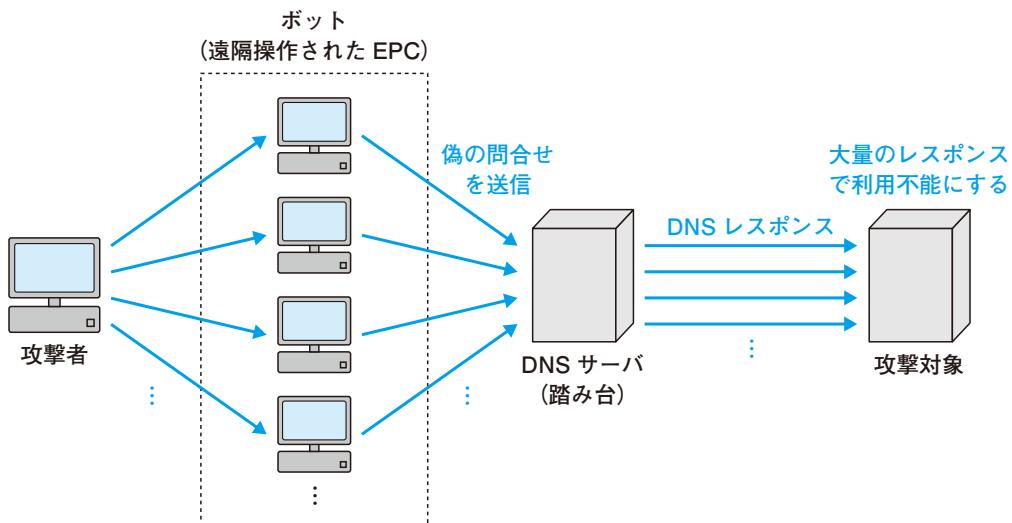
DMZ のゾーン情報とキャッシュサーバの機能を受け持つのは、DMZ にある「外部 DNS サーバ」です。DNS サーバは、「このドメイン名に対応する IP アドレスを教えて」という問合せに対して、必要な情報を返すサーバです。IP アドレスを知っていれば IP アドレスを返しますが、知らない場合は上位の DNS サーバに問合せを行います。一度問合せで得た情報はキャッシュサーバに保存されるので、次に同

じ問合せがきたときは、上位の DNS サーバに再度問合せをする手間を省いています。

ただし、このキャッシュサーバが外部からの問合せも受け付ける設定になっていると、これを悪用した攻撃に利用されてしまうことがあります。

この攻撃は、DNS サーバに大量の問合せを送り、その返信先として攻撃対象の IP アドレスを指定するもので、攻撃対象に大量の DNS レスポンスが送られてしまいます。この攻撃を **DNS リフレクション攻撃**といいます。

攻撃対象となるサーバに大量の通信を送り付け、サーバを利用不能状態にする攻撃を **DoS 攻撃** (Denial of Service attack) といいます。とくに DNS リフレクション攻撃では、遠隔操作された複数のコンピュータ（ボット）を使って、DNS サーバへの問合せを行います。このような DoS 攻撃を **DDoS 攻撃** (Distributed DoS attack) といいます。DNS リフレクション攻撃は、DDoS 攻撃の一種です。



- d は、P 社からインターネット上の Web サーバへのアクセスを中継し、通信ログを取得する。

社内の PC から外部の Web サーバにアクセスする際、アクセスを中継するサーバは「**プロキシサーバ**」です。プロキシサーバを利用すると、PC が外部とやり取りするログが記録できます。また、PC が外部と直接データのやり取りをしないので、マルウェアの侵入を防ぐことができます。

以上から、空欄 a 「FW」、空欄 b 「メール中継サーバ」、空欄 c 「外部 DNS サーバ」、空欄 d 「プロキシサーバ」の **ウ** が正解です。

解答

問19 **ウ**



合格のカギ

社内 LAN を構成するネットワーク機器やサーバの役割を理解しましょう。



問

20

次の記述を読んで、設問に答えよ。

通信販売会社のA社のオフィスは、入退室管理システムによって入室制限が行われている。社員は、非接触型ICカードである社員カードを所持しており、社員カードを部屋の入口に設置された読み取り装置にかざすと、社員カード内に記録されたIDによって入室の可否が判断される。入室が許可されるとドアが解錠される。オフィスの執務エリアは、間仕切りのない設計になっている。執務エリアにはプリンタ、コピー、文書保存などの機能をもつ複合機が3台設置され、複数の課で共有している。

A社では今回、物理的対策を中心にオフィスのセキュリティを見直すことになった。調査の結果、報告された問題点（一部）を図に示す。

- (一) 入退室の際、共連での入室が散見される。
- (二) 個人データが印刷された書類が複合機に放置されていることがある。
- (三) 書類や印刷物などを机の上に放置したままの離席が散見される。

図 報告された問題点（一部）

設問 報告された（一）～（三）の問題点への対策として、最も適切なものの組合せを解答群から選べ。

解答群

	(一)	(二)	(三)
ア	アンチパスパック	オンデマンド印刷	クリアスクリーン
イ	インターロック	オンデマンド印刷	クリアスクリーン
ウ	アンチパスパック	オンデマンド印刷	クリアデスク
エ	インターロック	オンデマンド印刷	クリアデスク
オ	アンチパスパック	オフセット印刷	クリアスクリーン
カ	インターロック	オフセット印刷	クリアスクリーン
キ	アンチパスパック	オフセット印刷	クリアデスク
ク	インターロック	オフセット印刷	クリアデスク

※出典：令和3年度秋期応用情報技術者試験午後問題問1を
改変

問題解説

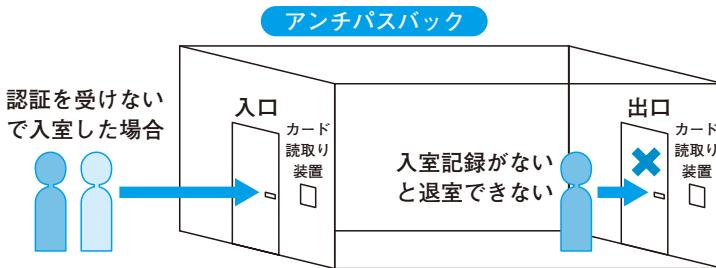
問題点（一）～（三）の対策を順に検討します。

（一）入退室の際、共連での入室が散見される。

社員の1人が自分の社員カードを読み取り装置にかざしてドアが開いたタイミングで、ほかの社員がカードをかざさずにいっしょに入室してしまうという問題です。社員ならいいのですが、関係者以外の人が入

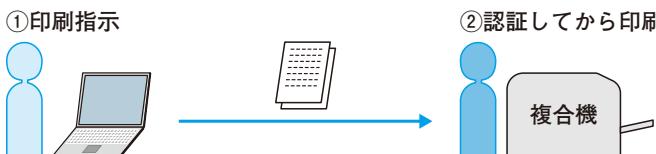
室する可能性があります。

この問題への対策として、カード読み取り装置を出口側にも設定して、入室状態になっていない人が退室しようとした場合は解錠しないようにする方法があります。このような仕組みを「アンチパスバック」といいます。



(二) 個人データが印刷された書類が複合機に放置されていることがある。

社員がPCから複合機に印刷指示を出し、個人データを含む書類を印刷して、そのまま放置してしまう状態です。この状態を防ぐための方法としては、複合機にカード読み取り装置などを装備しておき、社員が印刷指示した書類については、複合機に社員カードをかざして認証を受けた後で印刷されるようにする対策が有効です。印刷指示を出した社員が複合機の側に行かないと印刷が実行されないため、書類の放置を防ぐことができます。このような仕組みを「オンデマンド印刷」といいます。



(三) 書類や印刷物などを机の上に放置したままの離席が散見される。

重要な書類や印刷物を机の上に放置したり、鍵についていない引き出しに入れたままにして離席するのは、情報を盗まれるなどのセキュリティリスクがあります。離席する際に、書類やノートパソコンなどを机の上に出したままにせず、鍵のついたキャビネットなどに保管することを「クリアデスク」といいます。

なお、コンピュータをログインした状態のまま離席するのも、不正操作や情報の盗難などのリスクがあります。離席する際にコンピュータをログオフしたり、コンピュータの画面をロックすることを「クリアスクリーン」といいます。

以上から、問題(一)の対策が「アンチパスバック」、(二)の対策が「オンデマンド印刷」、(三)の対策が「クリアデスク」のウが正解です。

○ 解答 ○

問20 ウ

合格のカギ

最近では、入退室管理や机の整理といった、情報システム以外のセキュリティが重視されるようになっています。自分が勤めている会社のセキュリティ体制を見直すこともヒントにつながります。

■擬似言語の記述形式

擬似言語を使用した問題では、各問題文中に注記がない限り、次の記述形式が適用されているものとする。

〔擬似言語の記述形式〕

記述形式	説明
○ 手続名又は関数名	手続又は関数を宣言する。
<u>型名</u> : <u>変数名</u>	変数を宣言する。
/* <u>注釈</u> */	注釈を記述する。
// <u>注釈</u>	
<u>変数名</u> ← <u>式</u>	変数に <u>式</u> の値を代入する。
手続名又は関数名 (<u>引数</u> , ...)	手続又は関数を呼び出し、 <u>引数</u> を受け渡す。
if (<u>条件式</u> 1) <u>処理</u> 1 elseif (<u>条件式</u> 2) <u>処理</u> 2 elseif (<u>条件式</u> n) <u>処理</u> n else <u>処理</u> n + 1 endif	選択処理を示す。 <u>条件式</u> を上から評価し、最初に真になった <u>条件式</u> に対応する <u>処理</u> を実行する。以降の <u>条件式</u> は評価せず、対応する <u>処理</u> も実行しない。どの <u>条件式</u> も真にならないときは、 <u>処理</u> n + 1 を実行する。 各 <u>処理</u> は、0 以上の文の集まりである。 <u>elseif</u> と <u>処理</u> の組みは、複数記述することがあり、省略することもある。 <u>else</u> と <u>処理</u> n + 1 の組みは一つだけ記述し、省略することもある。
while (<u>条件式</u>) <u>処理</u> endwhile	前判定繰返し処理を示す。 <u>条件式</u> が真の間、 <u>処理</u> を繰返し実行する。 <u>処理</u> は、0 以上の文の集まりである。
do <u>処理</u> while (<u>条件式</u>)	後判定繰返し処理を示す。 <u>処理</u> を実行し、 <u>条件式</u> が真の間、 <u>処理</u> を繰返し実行する。 <u>処理</u> は、0 以上の文の集まりである。
for (<u>制御記述</u>) <u>処理</u> endfor	繰返し処理を示す。 <u>制御記述</u> の内容に基づいて、 <u>処理</u> を繰返し実行する。 <u>処理</u> は、0 以上の文の集まりである。

〔演算子と優先順位〕

演算子の種類		演算子	優先度
式		().	
単項演算子		not + -	
二項演算子	乗除	mod × ÷	高 ↑
	加減	+ -	
	関係	≠ ≤ ≥ < = >	
	論理積	and	
	論理和	or	↓ 低

注 演算子 . は、メンバ変数又はメソッドのアクセスを表す。

演算子 mod は、剰余算を表す。

〔論理型の定数〕

true, false

〔配列〕

配列の要素は、“[”と“]”の間にアクセス対象要素の要素番号を指定することでアクセスする。なお、二次元配列の要素番号は、行番号、列番号の順に“,”で区切って指定する。

“{”は配列の内容の始まりを、“}”は配列の内容の終わりを表す。ただし、二次元配列において、内側の“{”と“}”に囲まれた部分は、1行分の内容を表す。

〔未定義、未定義の値〕

変数に値が格納されていない状態を、“未定義”という。変数に“未定義の値”を代入すると、その変数は未定義になる。

索引

英数字

2相ロック	123	FRR	129	SaaS	49, 101, 300
3C分析	241	GPGPU	25	SEO	37, 231
ABC分析	51	GPL	119	SEO ポイズニング	37, 231
AES	37	GPU	25	SFA	139
AND	35, 111, 121, 211	HTML	35	SHA-256	199
APOP	41, 233	HTTPS	229	SLA	45, 49
ARP	125	IaaS	101, 300	SMTP	35
ASCIIコード	285	IDS	37	SMTP Submission	41
ASP	49	if ~ elseif ~ else 構文	56, 152, 247	SMTP-AUTH	233
BCP	47, 243	IMAPS	41	SOA	139
BPO	49, 139	IoT	143	Society 5.0	51
BPR	49, 139	IPsec	103, 229	SQL	223
C&Cサーバ	38	IPv4	125	SQL インジェクション	35, 127
CA	127	IRR	245	SRAM	31, 119
CAPTCHA	232	IT ポートフォリオ	241	SSD	31
CGM	143	JIS Q 27001	131	SSH	206
CHAP	39	LAN アナライザ	229	SSL/TLS	41
CIO	244	LRU	119	SWOT 分析	141
COCOMO	237	MHS	35	Telnet	206
CRYPTREC	233	MIME	35	TLS	206, 229
D/A変換器	31	MIPS	23	UML	41, 132, 139, 223
DDoS攻撃	303	MOT	243	WAF	39, 129
DFD	41, 133, 233	MTBF	117	WBS法	135, 237
DHCP	33, 225	MTTR	117	Web 2.0	51
Diffie-Hellman鍵交換法	103	NAND	120	while ~ endwhile 構文	67, 149, 254
DMZ	107, 231	NAPT	33, 225	WPA	229
DNS	33	NAT	33, 225	XOR	121, 209
DNSキャッシュポイズニング攻撃	227, 231	NOR	121	XP	235
DNSリフレクション攻撃	303	NOT	211		
do ~ while構文	67, 149, 254	NPV	245		
DoS攻撃	303	OR	121, 209, 211		
DRAM	31, 119, 219	OSI基本参照モデル	34, 125, 225		
EA	47, 139, 241	OSS	119		
EEPROM	31	PaaS	101, 300		
e-Gov	51	PCM方式	113		
ERPパッケージ	49	PDCA	237		
E-R図	41, 139	POP3	41		
FAR	129	POP3S	233		
FDQN	227	PPPoE	225		
FIFO	119	RADIUS	33		
for構文	58, 155	RAID	217		
		return文	156		
		ROI	245		
		RPA	241		
		RSA	37		
		S/MIME	233		

あ行

アクセシビリティ	243
アクティビティ図	132
アジャイル開発	43
後入れ先出し	72, 174, 266
アナグラム	194
アプリケーションアーキテクチャ	47, 139
アベイラビリティ	45
アローダイアグラム	45, 135, 237
アンチエイリアシング	121
アンチパスバック	305
移行	239
インスタンス	42, 77
インターネットオークション	143
インタプリタ	211
ウェルノウンポート番号	129

ウォードライビング	37
エクストリームプログラミング	235
エッジ	211
エンキュー	174
エンタープライズアーキテクチャ	47, 49, 139, 241
オーバフロー	19
オープンソースソフトウェア	119
オブジェクト指向	43, 133
重み	92
親	80, 269
オンデマンド印刷	305
オンラインモール	143

か行	
外観上の独立性	239
回収期間法	245
階乗	69
階層ディレクトリ構造	29
外部結合	43, 233
外部要因	141
外部割込み	25, 115
稼働率	27, 217
金のなる木	141
株式資本等変動計算書	143
カプセル化	133
可用性	39, 45
カレントディレクトリ	29
環境ラベリング制度	139
関係	223
関係データベース	32
間順走査	80, 168
関数	55, 156
関数従属	122
完全性	39
かんばん方式	243
関連	41
機械学習	21
木構造	80, 168, 269
機能要件	241
機密性	39
逆ポーランド記法	73, 168
キャッシュフロー計算書	53, 143
キャッシュメモリ	117
キュー	174
強化学習	211

教師あり学習	211
教師なし学習	211
競争戦略	141
共通鍵暗号方式	37, 103
共通結合	43
共有結合	233
行列	183
行列演算ユニット	25
金種計算	181
クイックソート	275
区分コード	211
組合せ	21
クラス	42, 76, 133, 266
グラフ	171, 211, 273
クリアスクリーン	305
クリアデスク	305
グリーン調達	139
グリーン電力証書	139
グリーン投資	139
クリティカルパス	45, 237
グローバル IP アドレス	33
クロスコンパイル	117
クロスサイトスクリプティング攻撃	35, 127, 231
クロスサイトリクエストフォージェリ攻撃	35
クロック周波数	215
継承	133
ゲートウェイ	34
桁落ち	19
桁別コード	211
結合	31, 121, 223
原始プログラム	211
子	80, 168, 269
コアコンピタンス	49
公開鍵	197
公開鍵暗号	127
公開鍵暗号方式	37, 103, 129
後順走査	80, 168
後置記法	73, 168
コールドスタンバイ	27
固定費	245
根	168, 269
コンパイラ	117
コンプライアンス	49
コンペア法	131

コンポーネント図

さ行

サービスデスク	137
再帰的	215
再帰的関数	23, 115, 213
再帰的プログラム	69, 160
最高情報責任者	244
再使用可能	215
最小公倍数	259
再早開始日	135
最大公約数	66, 259
最適化	117
再入可能	215
再配置可能	215
財務諸表	143
先入れ先出し	175
先入先出法	53
サブネットマスク	35, 125
差分バックアップ	219
サンドボックス	227
サンプリング周波数	19
シーケンス図	132
事業継続計画	47, 243
事業ドメイン	49
事業部制組織	51
辞書攻撃	231
システム監査人	47, 137, 239
システムテスト	239
実体	41
指摘事項	137
シフト演算	111
ジャーナルファイル	33
射影	31, 121, 223
シャドウページ	123
主キー	123
順列	21
条件網羅	133, 135, 235
状態遷移図	41, 133
衝突	287
情報落ち	19
情報システム部門	245
情報弱者	243
情報セキュリティ	39
情報バリアフリー環境	243
剩余演算子	147

ジョーアカウント攻撃	127
職能別組織	51
真正性	39
深層学習	21, 143
信頼性	39
衰退期	49
スケジューリング	27, 118
スタック	72, 113, 174, 213, 266
スタブ	234
スタンプ結合	43
ステートフルインスペクション	41
ステートマシン図	133
スマート工場	143
スマートメータ	143
制御結合	43, 233
成熟期	49
製造物責任法	245
生体認証システム	129
成長期	49
静的解析	227
正の相関	112
積集合	111
責任追跡性	39
節	80, 168
セッションハイジャック攻撃	127
セル生産方式	51
線形	113
前順走査	80, 168
全体最適化	139
選択	31, 121, 223
相関係数	112
走査	80
挿入ソート	83
増分バックアップ	219
ソーシャルエンジニアリング	231
ソートアルゴリズム	83
属性	223
損益計算書	143

た行

第三者中継	298
貸借対照表	143
ダイバーシティ社会	51
タブル	223
ダンプファイル	33
単方向リスト	76, 262

チェックサム法	131
チェックディジット	86
チェックポイント	123
チェックポイントファイル	33
チャレンジャ	141
チャレンジレスポンス	39
中央サービスデスク	137
中央値	96
抽象化	133
中置記法	73, 168
直列システム	27, 217
著作権法	53
突合せ処理	294
ディープラーニング	21, 143
デジタル証明書	127
デジタル署名	37, 196, 229
デジタルデバイド	243
ディレクトリ	28
ディレクトリトラバーサル攻撃	126
データアーキテクチャ	47, 139
データウェアハウス	139
データ結合	43, 233
データディクショナリ	139
データフローダイアグラム	139, 233
データマイニング	139
データリンク層	125, 225
デキュー	174
テクノロジアーキテクチャ	47, 139
テスト駆動開発	43
手続	55
デバッグ	117
デュアルシステム	219
デュプレックスシステム	27, 219
ド・モルガンの法則	159
投資利益率	245
動的解析	227
導入期	49
トポジカルソート	171
ドライバ	234
トランジクション	211
トレーサ	29
トレース	61, 260

な行

内部要因	141
内部割込み	25, 115

内容結合	43
二次元配列	98, 283
ニッチャヤ	141
二分木	80, 168, 269
二分探索木	269
二分探索法	163
ニューラルネットワーク	21
認証局	127
ネットワークアドレス	35
ネットワーク層	125, 225
ノード	211

は行

パーセプトロン	92
バーチャルサービスデスク	137
バイアス	92
バイオメトリクス認証	129
排他的論理和	121, 209
バイトコンパイル	117
ハイブリッド暗号方式	233
配列	58, 250
パケットキャプチャ	233
パケットフィルタリング	41, 131, 225
パスワード	105
パスワードリスト攻撃	127, 231
パターンマッチング法	227
バックアップファイル	33
バックドア	229
ハッシュ関数	199
ハッシュ値	105, 131
ハッシュテーブル	287
ハッシュ法	211
バッファオーバフロー攻撃	229
バッファリング	123
花形	141
バランススコアカード	241
判定条件／条件網羅	235
判定条件網羅	134, 235
ヒープ	190
ヒープソート	191
引数	61
非機能要件	241
ビジネスアーキテクチャ	47, 139
否定	158
否定論理積	120

否定論理和	121
否認防止	39
ビヘイビア法	131
秘密鍵	197
表意コード	211
標準タスク法	237
標準偏差	280
標本化	113
ファンクションポイント法	45, 135, 237
フォロワ	141
複数条件網羅	235
符号化	113
ブッシュ	72, 113, 174, 213, 266
物理層	125, 225
負の相関	112
プライベート IP アドレス	33
ブラックボックステスト	43, 133
フラッシュメモリ	31, 219
ブリッジ	34, 125
ブルートフォース攻撃	127, 231
フルバックアップ	219
プロキシサーバ	226, 303
プログラムステップ法	135
プロジェクト組織	51
プロジェクトマネージャ	245
プロセスイノベーション	243
プロダクトイノベーション	243
プロダクトポートフォリオマネジメント	141
プロダクトライフサイクル	49
プロトタイピング	43
分岐網羅	134, 235
分散	280
ペアプログラミング	43, 235
平均故障間隔	117
平均修理時間	117
平均値	178
併合	121
並行システム	219
ペイズの定理	19
並列システム	27, 217
ページ置換えアルゴリズム	119
ベンチマーкиング	241
変動費	245
変動費率	245
ポートスキャナ	229
ポート番号	129, 227
ポートフォリオ	241
ホーネー法	64
ホスティングサービス	49
ホットスタンバイ	27
ボットネット	38
ポップ	72, 113, 174, 213, 266
ホワイトボックステスト	43, 133
ま行	
負け犬	141
マップ	194
マトリックス組織	51
マルウェア	129, 227
マルチプロセッサシステム	27
マルチユーザシステム	27
丸め誤差	19
ミラーリング	217
無向グラフ	171, 211, 273
無相関	112
命令網羅	135, 235
メールマガジン	143
メディアコンバータ	225
メディアン	96
モジュール結合度	43, 233
モデリング	121
戻り値	55, 61, 156
問題児	141
や行	
ユークリッドの互除法	66, 215, 259
有向グラフ	273, 171
ら行	
ライトスルーアクション	117
ライトバック方式	117
ライブラリ	29
ライン生産方式	51
リーダ	141
リエンジニアリング	133
リエントラント	215
リカーシブ	215
リスク受容	131
リスク対応	131
リスク分析	131
リスト	22
リバースエンジニアリング	133
リバースブルートフォース攻撃	127, 231
リバースプロキシ	39
リピータ	35, 125
リピータハブ	225
リファクタリング	43, 133
リフレッシュ	31, 119, 219
リユーズブル	215
量子化	113
量子化ビット数	19
リロケータブル	215
リンク	29
隣接行列	211, 273
類推法	135, 237
ルータ	35, 125
ルートディレクトリ	29
レイヤ2スイッチ	225
レイヤ3スイッチ	225
レンダリング	121
連番コード	211
労働者派遣	145
ローカルサービスデスク	137
ローダ	29
ロールバック	123
ロールフォワード	123
ログファイル	33
論理積	35, 111, 121, 158, 211
論理否定	211
論理和	121, 158, 209, 211
わ行	
ワーム	129
和集合	111, 223
割込み	25, 117
ワンタイムパスワード	105

本書のご感想をぜひお寄せください	
https://book.impress.co.jp/books/1122101100	
読者登録サービス 	アンケート回答者の中から、抽選で図書カード(1,000円分)などを毎月プレゼント。 当選者の発表は賞品の発送をもって代えさせていただきます。 ※プレゼントの賞品は変更になる場合があります。
	

■商品に関する問い合わせ先

このたびは弊社商品をご購入いただきありがとうございます。本書の内容などに関するお問い合わせは、下記のURLまたは二次元バーコードにある問い合わせフォームからお送りください。

<https://book.impress.co.jp/info/>

上記フォームがご利用いただけない場合のメールでの問い合わせ先
info@impress.co.jp



※お問い合わせの際は、書名、ISBN、お名前、お電話番号、メールアドレスに加えて、「該当するページ」と「具体的なご質問内容」「お使いの動作環境」を必ずご明記ください。なお、本書の範囲を超えるご質問にはお答えできないのでご了承ください。

- 電話やFAXでのご質問には対応しておりません。また、封書でのお問い合わせは回答までに日数をいたたく場合があります。あらかじめご了承ください。
- インプレスブックスの本書情報ページ <https://book.impress.co.jp/books/1122101100> では、本書のサポート情報や正誤表・訂正情報などを提供しています。あわせてご確認ください。
- 本書の奥付に記載されている初版発行日から3年が経過した場合、もしくは本書で紹介している製品やサービスについて提供会社によるサポートが終了した場合はご質問にお答えできない場合があります。

■落丁・乱丁本などの問い合わせ先

FAX 03-6837-5023

service@impress.co.jp

※古書店で購入された商品はお取り替えできません。

かんたん合格 基本情報技術者予想問題集 令和5年度

2022年12月21日 初版第1刷発行

著者 株式会社ノマド・ワークス

発行人 小川 亨

編集人 高橋隆志

発行所 株式会社インプレス

〒101-0051 東京都千代田区神田神保町一丁目105番地

ホームページ <https://book.impress.co.jp/>

本書は著作権法上の保護を受けています。本書の一部あるいは全部について、株式会社インプレスから文書による許諾を得ずに、いかなる方法においても無断で複写、複製することは禁じられています。

Copyright © 2022 NOMAD WORKS. All rights reserved.

印刷所 日経印刷株式会社

ISBN978-4-295-01574-1 C3055

Printed in Japan