情報 2011年度共通試験解答例

文責：文科3類13組　佐野

## **はじめに**

**～注意書き～**

この解答が正確である保証はありません。というか、**作った人は文系なのであまり期待しないでください。**間違いを見つけた方は、鼻で笑って直して下さい。解答例は読む人が理解できることに重点を置いたため、長ったらしくなってるのは勘弁して下さい。

ちなみに解説や注釈は基本個人の趣味で構成されています。理解の助けになるかな？　なお、一部の語句説明は辞書に依るものがあり、教科書の記述と異なるかもしれません。注意して下さい。

口調が偉そうなのは受験期に読んだ参考書の影響です。気にしないで下さい。

文字数が多くてだるいのは仕様です。（´・ω・｀）

**色を付けて強調してあるとこ**を読むだけでも幸せになれるかもしれない。

質問があれば私を探し出して下さい。誰かに聞けばすぐ分かるはずです。見つからなかったら理系の人にでも聞いて下さい。

書体を統一できなかったorz

**～この解答を作るにあたっての参考資料～**

・情報の教科書

・授業で配られたプリント

・友人のレポート

・情報（原田）シケプリ（1、2、3、4全て）

・2006年度夏学期　シケプリ de 情報

・only my information（情報 共通問題 06～10夏 解答解説）

一部独自に入手したものがありますが、とりあえずこの解説を読んで分からないことがあれば、上の参考資料に目を通してみて下さい。

**～スペシャルサンクス～**

・同クラのO君（特に共通鍵・公開鍵暗号方式）

・同クラのF君

・その他校正を手伝ってくれた方々

・And You!!

**それでは、「情報」というこの素晴らしきクソゲーを乗り越えるため、皆さん頑張って下さい。**

## **解説編**

# **共通問題1**

## 1-1

**(1)解答**

・誤り：**コンピュータが、通信ケーブルで直接に結ばれている必要がある**

**・文章例**：メールデータはまず、送信側クライアントから送信側サーバに送られる。送信側サーバはDNSに宛先のIPアドレスを問い合わせて受信側サーバにデータを送り、そのデータを受信側サーバが受信側クライアントに送ることで、メールの送信が完了する。この中でデータはパケットに分割され、そのマシン間のやりとりはルータを介して行われる。（154字）

**・解説**

ネットワークを正しく理解しているかどうかが問われる問題である。

**～指定語句解説～**

**DNS**：Domain Name System

→**IPアドレス（注1、2）とドメイン名（注3）を対応づけるシステム**。ウェブページに接続したりする際は、このシステムが組み込まれたサーバに接続先のドメイン名を問い合わせてIPアドレスを取得し、そのIPアドレスをもつサーバに、ルータを介して接続している。

**パケット**

→送信の際に**細かく分けられたメッセージデータの総称**。このうち本文が含まれる部分をテキスト、送信先が含まれる部分をヘッダーと呼ぶ。

**ルータ**

→**異なるネットワーク同士を接続するもの**。ネットワークの接続点だと思えば良い

**～問題解説～**

例文では「コンピュータが、通信ケーブルで直接に結ばれている必要がある」という部分が間違っている。ネットワークにおいてはコンピュータ同士がケーブルで直接結ばれていることはまれで、大抵はルータと呼ばれる機器を結節点にしてつながっている。通信の際にはいくつものルータを経由することが多い。

文章解答のポイントは、

・**データ（パケット）が移動する順番（クライアント→サーバ→サーバ→クライアント）**

・**サーバがDNSに宛先のIPアドレスを問い合わせる**

・**データはパケットに分割される**

・**マシン間のパケットの移動はルータが仲介する**

である。以上のポイントを書き込めれば十分であろう。

→なお、類似の問題が**2006年度の共通問題1**で出題されている。そちらの方も参照のこと。

→短くまとめられなかったorz

**(2)解答**

**・誤り**：**お互いに鍵を公開し、公開された鍵を用いて暗号化と復号を行う**

**・文章例**：公開鍵暗号方式では、受信者は暗号化のための鍵と復号のための鍵を別々につくり、暗号化のための鍵の方を公開鍵として公開する。公開鍵を使って送信者は送信するデータを暗号化し、そのデータを受信者は秘密鍵で平文に復号する。送信の際の安全性は高いが、復号の際に時間がかかる。（131字）

**・解説**

鍵暗号についての理解が問われる問題である。

**～指定語句解説～**

**公開鍵・共通鍵・秘密鍵**

→文章を暗号化する方法には2種類ある。一つは共通鍵暗号方式で、もう一つは公開鍵暗号方式である。

・**共通鍵暗号方式**：**暗号化と復号に同じ情報（鍵）を使う**暗号化方式である。この鍵を**共通鍵**と呼ぶ。この鍵は送信者と受信者間で共有される。この鍵は「不特定多数に」公開されない（注4）。

・**公開鍵暗号方式**：**暗号化と復号にそれぞれ違う情報（鍵）**を使い、そのどちらかを「不特定多数」に公開する暗号化方式である。このとき、**公開された一方の情報**を**公開鍵**、**公開されないもう一方の情報**を**秘密鍵**と呼ぶ（注5）。

→**共通鍵暗号方式**は**一対一のやりとり**に、**公開鍵暗号方式**は**一対不特定多数のやりとり**に用いられると考えれば理解が早いかもしれない。

**平文**

→**暗号化の対象となるもとの文**である。

**～問題解説～**

例文は、「お互いに鍵を公開し、公開された鍵を用いて暗号化と復号を行う」というところが間違っている。**お互いに鍵を公開**するのは**共通鍵暗号方式**である。**公開鍵暗号方式**は、**受信する側が送信のための公開鍵を公開**するもので、秘密鍵を公開したりはしない（でなきゃ秘密鍵の意味がない）。

また、**公開された鍵で暗号化と複合化を行う**のも**共通鍵暗号方式**である。

文章解答のポイントは、

・**受信者は秘密鍵と公開鍵をつくり、公開鍵を公開**

・**送信者は公開鍵で平文を暗号化**

・**暗号化された文を、受信者が秘密鍵で復号**

・**安全性は高いが、復号の際に時間がかかる**

である。以上のポイントを書き込めれば十分であろう。

→なお、これと全く同じ問題が**2006年度の共通問題1**で出題されている。~~過去問リサイクル乙。~~

**(3)解答**

**・誤り**：**a×10秒より長くかかることはない**

**・文章例**：現在インターネットで主に使われる通信回線の交換方式はパケット交換方式である。この方式でサーバと通信する際は、データをパケットに分割し、ルータを介してやりとりする。通信速度は通信回線の混雑具合に左右され、通信回線が混雑していると通信速度は低下する。（123字）

**・解説**

インターネットにおけるデータのやりとりに対する理解が求められる問題である。

**～指定語句解説（上で重複しているものは省略）～**

**サーバ**

→**他のコンピュータにサービスを提供することを目的とするソフトウェア全体**（もしくはそのソフトウェアが稼働しているハードウェア）をいう。一般的にサーバと言えばファイルが存在する場所を指す（かも）。

**～問題解説～**

例文は、「a×10秒より長くかかることはない」というところが間違っている。現在のインターネットにおける通信回線交換方式の主流は**パケット交換方式（注6）**である。パケット交換方式では、**通信速度は交換機の回線の混雑具合に左右され、一定の通信速度を得られない**。当然のことながら、1メガのファイルをダウンロードするのにa秒かかったからといって次に10メガのファイルをダウンロードする時間がa×10秒になるとは限らないし、むしろそれ以上かかることの方が多い。

文章解答のポイントは

・**インターネット通信方式の主流はパケット交換方式**

・**データはパケットに分割される**

・**パケット交換方式での通信速度は、交換機のパケット混雑具合により左右**

である。以上のことを書き込めば十分であろう。字数に余裕があればDNSやルータについて言及してもいいかもしれない。ただし、主題から外れたことを書くのは望ましくない。

**(4)解答**

**・誤り**：**接続者の国、地域、所属組織などとの対応関係は一切ない**

**・文章例**：ドメイン名は、コンピュータの所在地を人間に分かるように示すものである。ドメイン名はコンピュータの所属を示し、それぞれのドメイン名はDNSによってIPアドレスと関連づけられる。また、あるドメイン名をもつ当該機関にその下にあるドメインの管理を一任することで、ドメインの分散管理が可能である。（140文字）

**・解説**

ドメイン名やIPアドレスに対する理解を求める問題である。

**～指定語句解説～**

→（1）の説明や注釈１，２，３を参照のこと。

**～問題解説～**

例文は、「接続者の国、地域、所属組織などとの対応関係は一切ない」ということが間違っている。ドメイン名はコンピュータがどこに所属しているかを示し、コンピュータを統括して管理するのに使うため、現実の機関と対応させる必要がある。

文章解答のポイントは、

・**ドメイン名は、人間向けに表されたコンピュータの住所**

・**DNSによりIPアドレスと関連づけ**

・**ドメイン名を用いての分散管理**

である。ホスト名とドメイン名を混同しないよう気を付けるように（詳しい説明は注釈参照）。

## **共通問題1-2**

# 1-2解答

ア：(b)HTTP　イ：(c)プログラム　ウ：(d)文書を収集　エ：(b)プログラム　オ：(c)データベース

## 1-2解説

ア

「規約」と言う言葉から「プロトコル（注7）」という言葉を連想しよう。各選択肢の内容を見ていくと、

(a)**HTML**：HyperText Markup Language（超文書特徴化言語）

→**ウェブページ（注8）を作るときなどに使われる言語（注9）**。「～.html」「～.htm」と拡張子に書かれているファイルは、全てHTMLで書かれている。

→なお、同じように使われる言語としてXML（eXtensible Markup Language）が存在する。私はこれに関して詳しくないので説明は割愛。

(b)**HTTP**：HyperText Transfer Protocol（超文書転送規約）

→ウェブページをウェブブラウザー（注10）に**転送する際のプロトコル**。

(c)**SMTP**：Simple Mail Transfer Protocol（簡易郵便転送規約）

→**メールを送受信する際のプロトコル**。

→なお、セキュリティ上様々な問題があることを覚えておこう。

(d)**URL**：Uniform Resource Locator（統一的資源所在地決定要素）

→**ファイルが存在する場所を示す文字列（注11）**のこと。住所。

→URIと混同されているとか言われるが、混乱するので考えたら負け。

以上の通りである。

したがって、以上の説明より**(b)が正しい**ことが分かる。

イ

まず最初に略語の解説を。

**CUI**：Character-based User Interface

→**文字を中心としたインターフェース（注12）**。Macの「ターミナル」やWindowsの「コマンドプロンプト」がそれに当たる。文字で指示を出すことを操作の中心とし、これを使いこなせれば上級者。

→対義語：GUI（Graphical User Interface）

→グイッ

**LAN**：Local Area Network

→**一つの施設内（例えば東京大学）のみをカバーするコンピュータネットワーク**。

→注意すべきこととして、東京大学のLANに接続しているだけでは、外部のネットワーク（例えば早稲田大学のLAN）に接続することができない。東京大学のLANと早稲田大学のLANが接続状態にあって初めて早稲田大学のLANに接続することができる。

→ちなみに、我々が普段インターネットと呼ぶものは、WAN（Wide Area Network）のことを指している。詳細はググれ。

以上より、回答の候補として残るのは(c)である。電子メールソフト以外のプログラムもウェブサーバと通信できるので、電子メールソフトに限定している(d)は不適。また、冒頭でHTTPの話をしているため、SMTPを扱う電子メールソフトはやはり不適である。

→なお、このようなソフト（プログラム）としては、2ちゃんビューワーやダウンロードNinjaなどが挙げられる。詳細は（以下略

ウ

HTTPは前述の通り**ウェブページをウェブブラウザーに転送する際のプロトコル**であるため、そこで行われるのは**文章の収集**である。(a)メール受信はそもそも文脈にそぐわないので不適、(b)決済を完了は間違ってはいないもののHTTPの話とずれるため不適。(c)モデル化？　何言ってんだこいつ(　・ω・)

エ

具体的な処理を行うのは(b)プログラムである。(d)**組み合わせ回路は、入力で出力が決まる回路**のことで、これ自体はプログラムの処理を支えるものである。

オ

　まず各選択肢の説明を行う。

**アルゴリズム**

→**問題を解決するための手順**。組み合わせ回路と同じくプログラムの処理を支えるものである。

→アルゴリズムが設計図で、組み合わせ回路がその設計図に従って作られたものである。設計図の書き方が残念だと作られるものも残念なことになる。

**データベース**

→**情報を系統立てて（例えば、五十音順）格納したもの**。検索をかけた際の速度が速い。

集合モデル

→データの分け方（データモデル（注13））の一つ。

　以上より、「公演情報などを記録した」ものとして適切なのは(c)データベースであることが分かる。

# 共通問題2

激しくめんどくさい

(1)解答例1 解答例2

i ← 1

j ← 1

while j ＜ 4 do

j ← j + 1

if votesｊ ＞ votesｉ

then

i ← j

endif

done

i ← 1

j ← 2

while j ＜ 5 do

if votesｉ ＞ votesｊ

then

j ← j + 1

else

i ← j

j ← j + 1

endif

done

(1)解説

**～記法の解説～**

**・一つの処理に一行**

**・“if”や“while”内の処理は、その構造毎に字下げを行う**

**・記号・処理解説**

**A（文字記号列）**：**「A」という変数**のこと。計算の必要上設定するもので、**自由に値を入れることができる**。

**A ←** **a**：**変数Aに値aを代入する**処理。Aにすでに値が入っていたとしても、**その値を上書き**する。

変数Aが以前の文で設定されていなければ、**同時に変数Aを設定**する。

**A ← A + a**：**変数Aに、Aの値に値aを加えたその値を代入**する処理。同じくAを上書きする。

**＞、＜**：言うまでも無く、大小記号のことである。**条件の指定に使う。**

**while “C” do**：**条件「C」を満たしている間**、**doneが現れるまでの処理をループして行わせる**処理。この処理の中に“while”や“if”を入れることができる。**処理を繰り返し行いたいときに使う。**

**done**：**“while”処理の終点**を示す。もし**条件「C」が満たされていなければ、“while C do”以下の処理に戻る**。これがないと、どこまでも“while”の処理が続いてエラーになる。

**if “D”**：**条件「D」を設定し、その中で指定された変数などが条件を満たしているかいないかで以降の処理を分ける**処理。**処理の分岐を行いたいときに使う。**

**then**：**指定された変数などが条件「D」を満たしているときに行う処理。**

**else**：**指定された変数などが条件「D」を満たしていないときに行う処理。**特に何も処理を行わないときは**省略可能。**

**endif**：**“if”処理の終点**を示す。これがないとthenやelseの処理が延々と続くことになる。

**～問題解説～**

　いきなり答えを書こうとすると勢いよくずっこける可能性があるので、まずは「どのようなアルゴリズムを作ればいいか」を考えてみよう。この際、自分の頭の中で言葉にしてみるのが一番である。

　まず、「最大値を求める」ための処理を考えてみよう。これができないと話が進まない。使えるのは上で示した処理だけであることにも注意しよう。

　一番楽な処理は、**「総当たり的に値を比較する」**ことだろう。本来なら「一括して全体の中から一番大きい値を求める」としたいところだが、そのような処理は使えないので諦める。

　さて、そのような処理はどのように行ったらいいだろうか。言葉にするならば、

・添え字値1と2が持つ要素値を比較する。

・要素値が大きい方の添え字値をiに代入する。

・添え字値iと添え字値3が持つ要素値を比較する。

・iの要素値が大きければ、添え字値iはそのまま。3の要素値が大きければ、添え字値iに3を代入。

・添え字値iと添え字値4が持つ要素値を比較する。

・iの要素値が大きければ、添え字値iはそのまま。4の要素値が大きければ、添え字値iに3を代入。

といった感じになるだろうか。

　これをより実際の処理に近づけてみると、

1. 添え字値“i”と添え字値“j”を設定する。

→添え字値“i”には常に**最大の要素値を持つ添え字値**が入り、添え字値jには**iと比較したい添え字値が入る**ものとする。

2. 添え字値“i”に値1を入れ、添え字値“j”には2を入れる。

→こうすることで、添え字値1と2の要素値が比較できるようになる。

3. 添え字値“i”と“j”の要素値を比較する。

4. 添え字値“i” が持つ要素値が大きければ、“j”に添え字値“j + 1”を代入。添え字値“j”が持つ要素値が大きければ、“i”に“j”の値を代入して、“j”に添え字値“j + 1”を代入。

→“j”に添え字値“j + 1”を代入することで、添え字値“i”と添え字値3の要素値が比較できる。

5. 以降処理3と処理4を繰り返す。

6. “j”が5になった時点で繰り返し処理を終了する。

→この時“i”に入っている値が、最大の添え字値である。

となる。問題文にあるように、要素値が互いに同値になることは考えなくて良い。

　以上の処理を、指定の記法に従って書いたのが先ほどの解答となる。

　ある意味論理力が試されるため、過去問で類似の問題を解いておくことをオススメする。

→説明が大変わかりにくくてごめんなさいm(\_ \_)m

　一応以下に、私がこの問題を解いた際の発想を書いておきます。多分こっちの方が分かりやすいかな？

・総当たり的に調べた方が楽じゃね？

↓

・まず1と2、次は大きい方と3、最後にまた大きい方と4を比較すればいいのか

↓

・いちいち書くのは面倒だし、“i”に比較して大きい方の値、“j”に比較したい方の値をいれて、“j”を1ずつ大きくしていけばいいや

↓

・だとしたら、最初に“i”に1、“j”に2を入れればいいじゃん

↓

・添え字値は4までだから、“j”が5になった時点で繰り返しを止めればいいよな

↓

・できた！

(2)解答

c = 7の時の配列“votes”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 添え字値 | 1 | 2 | 3 | 4 |
| 要素値 | 51000 | 81000 | 30000 | 20000 |

c = 4の時の配列“votes”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 添え字値 | 1 | 2 | 3 | 4 |
| 要素値 | 34000 | 27000 | 30000 | 20000 |

c = 0の時の配列“votes”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 添え字値 | 1 | 2 | 3 | 4 |
| 要素値 | 20400 | 20250 | 15000 | 20000 |

(2)解説

　ここまで来れば後は計算ゲームである。なお、**(1)ができていなくても解答できる**ことに注意したい。

　ちなみに配列votesの中身を求めるプロセスは二つあるが、そのうちの一つは気付かないと分からない。

**～プロセス1～**

ここでは、おとなしく指定のアルゴリズムに従って計算していく。

**・初期状態（c = 8の時）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 添え字値 | 1 | 2 | 3 | 4 |
| votes | 102000 | 81000 | 30000 | 20000 |
| seats | 0 | 0 | 0 | 0 |

i　← 1

seats1← 0 + 1

⇒**seats1 = 1**

votes1 ← 102000× 1 ÷ （1 + 1）

⇒**votes1 = 51000**

c ← 8 – 1

⇒**c = 7**

**（最初の処理に戻る）**

**・c = 7の時**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 添え字値 | 1 | 2 | 3 | 4 |
| votes | 51000 | 81000 | 30000 | 20000 |
| seats | 1 | 0 | 0 | 0 |

i　← 2

seats2← 0 + 1

⇒**seats2 = 1**

votes2 ← 81000× 1 ÷ （1 + 1）

⇒**votes2 = 40500**

c ← 7 – 1

⇒**c = 6**

**（最初の処理に戻る）**

**・c = 6の時**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 添え字値 | 1 | 2 | 3 | 4 |
| votes | 51000 | 40500 | 30000 | 20000 |
| seats | 1 | 1 | 0 | 0 |

i　← 1

seats1← 1 + 1

⇒**seats1 = 2**

votes1 ← 51000× 2 ÷ （2 + 1）

⇒**votes1 = 34000**

c ← 6 – 1

⇒**c = 5**

**（最初の処理に戻る）**

**・c = 5の時**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 添え字値 | 1 | 2 | 3 | 4 |
| votes | 34000 | 40500 | 30000 | 20000 |
| seats | 2 | 1 | 0 | 0 |

i　← 2

seats2← 1 + 1

⇒**seats2 = 2**

votes2 ← 40500× 2 ÷ （2 + 1）

⇒**votes2 = 27000**

c ← 5 – 1

⇒**c = 4**

**（最初の処理に戻る）**

**c = 4の時**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 添え字値 | 1 | 2 | 3 | 4 |
| votes | 34000 | 27000 | 30000 | 20000 |
| seats | 2 | 2 | 0 | 0 |

i　← 1

seats1← 2 + 1

⇒**seats1 = 3**

votes1 ← 34000× 3 ÷ （3 + 1）

⇒**votes1 = 25500**

c ← 4 – 1

⇒**c = 3**

**（最初の処理に戻る）**

**c = 3の時**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 添え字値 | 1 | 2 | 3 | 4 |
| Votes | 25500 | 27000 | 30000 | 20000 |
| Seats | 3 | 2 | 0 | 0 |

i　← 3

seats3← 0 + 1

⇒**seats3 = 2**

votes3 ← 30000× 1 ÷ （1 + 1）

⇒**votes3 = 15000**

c ← 3 – 1

⇒**c = 2**

**（最初の処理に戻る）**

**c = 2の時**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 添え字値 | 1 | 2 | 3 | 4 |
| Votes | 25500 | 27000 | 15000 | 20000 |
| Seats | 3 | 2 | 1 | 0 |

i　← 2

seats2← 2 + 1

⇒**seats2 = 3**

votes2 ← 27000× 3 ÷ （3 + 1）

⇒**votes2 = 20250**

c ← 2 – 1

⇒**c = 1**

**（最初の処理に戻る）**

**c = 1の時**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 添え字値 | 1 | 2 | 3 | 4 |
| Votes | 25500 | 20250 | 15000 | 20000 |
| Seats | 3 | 3 | 1 | 0 |

i　← 1

seats1← 3 + 1

⇒**seats1 = 4**

votes1 ← 25500× 4 ÷ （4 + 1）

⇒**votes1 = 20400**

c ← 1 – 1

⇒**c = 0**

**（cが0になったので処理を終了）**

**結果**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 添え字値 | 1 | 2 | 3 | 4 |
| Votes | 20400 | 20250 | 15000 | 20000 |
| Seats | 4 | 2 | 1 | 0 |

　　答えを求めるプロセスは以上となる。しかし、このプロセスを実際に行うのは明らかに効率が良くない。そこで、もう一つのプロセスを紹介する。

**～プロセス2～**

　さて、上のプロセス1を見ていると、一つの法則が見えてくる。

法則：「その時点で一番要素値votesが大きい添え字値の要素値seatsを+1し、その要素値votesに「 」を代入する。その後、cから-1する」

　この法則が見えてくれば、処理の計算が楽になるだろう。実際の過程は割愛。

(3)解答

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 添え字値 | 1 | 2 | 3 | 4 |
| 要素値 | 4 | 3 | 1 | 0 |

→答えの求め方は(2)参照

　数学的な論理力が試されるような問題である。このような問題で重要なのは、**まずどのような処理を行うべきか決めてから、その処理を実現するためのプロセスを考える**ことである。苦手なら本番ではスルーして、他の問題を解いた方がいいかもしれない。

# 共通問題3

わりかし何とかなる

## A-1

ア：シェルプロンプト

イ：コマンド

ウ：GUI

エ：アイコン

オ：ポインタ

カ：タッチパネル

キ：直接操作

ク：(c)

## 解説

ア：「g123456$」といった文字列を**シェルプロンプト**と呼ぶ。この文字列自体はシステムがコマンド待ちであることを示している。( ・ω・)ﾏﾀﾞｶﾅｰ　←こんな状態

イ：**コンピュータに指示を与えるための文字列**をコマンドと呼ぶ。キーボード操作に慣れない人にとってCUIが難しく感じる要因の一つかもしれない。

ウ：CUIが文字中心のコンピュータ操作であるのに対し、**GUIはグラフィック中心のコンピュータ操作**である。直感的に操作が可能であるが、そのために背後で膨大な処理が必要となる。~~Windows Vistaが重いのはこの処理が下手くそなせい。~~

→どうでもいいが、GUIが最初に標準搭載されたのはアップルのマッキントッシュである。

エ～オ：**WIMPシステムはGUIの別称**であり、**Window（ウィンドウ）、Icon（アイコン）、Menu（メニュー）、Pointer（ポインター）**の４つの頭文字から名付けられていることを覚えておくこと。

カ：**表示画面を直接指で触って入力するデバイス**は、**タッチパネル**である。タッチパッドとの混同に注意。タッチパッドはノートパソコンにあるマウスカーソルを動かすところである。

キ：GUIの背景にある考え方は「直接操作(direct manipulation)」と呼ばれるもので、「その装置あるいは表示をみれば、どのように実行可能かが即座にわかるようになる」というものである。詳しい説明は教科書p.221を参照のこと。

ク：(a)はむしろ逆のことを行っている。(b)はキーボード操作に慣れているユーザのことを無視している上に、本来のダブルクリックの操作と混同してしまう可能性があり却って良くない。(d)は別の話（強いて言えば入力フォームの話）なので関係ない。

# A-2

(1)解答

(A)の問題としては、情報の機密性が維持できていないこと、問題が発生したときの対応が遅れていることが挙げられる。(B)の問題としては、個人の権利利益に直接的な被害を与えるような情報が漏洩しているということが挙げられる。（105文字）

(1)解説

　セキュリティーやプライバシーについての理解を求める問題である。

**～セキュリティについて～**

セキュリティには3つの側面が存在する。

・**機密性**：**認可されたものにのみ情報にアクセスできる**ようにすること

→これが犯されれば**「情報漏洩」**の問題

・**完全性**：**情報が完全で正確である**こと

→これが犯されれば**「情報の改ざん」**の問題

・**使用可能性**：**必要なときに必要な情報資源にアクセスできる**こと

→これが犯されれば**「使用妨害」**の問題

また、情報セキュリティ対策には、

・**問題の発生自体を防ぐ対策**

・**問題の発生を想定した対策**

・**問題の発生に対する対策**

・**問題の再発を防ぐ対策**

の4つがある。

　まず最初にすべきことは、ソニーが起こした二件の情報漏洩に対する問題点の分析である。

　簡単に二つの事件をまとめると、

1. クレジットカードの情報が漏洩していた可能性が発覚
2. 個人に関わる情報が漏洩していた可能性が発覚

ということである。ここから二つの事件に共通する(A)セキュリティの問題と(B)プライバシーの問題を挙げていこう。

　 (A)の問題としては、

・機密性が維持できていない

→情報の漏洩

・問題の発生に対する対策が遅れている

→「可能性が発覚」

　(B)の問題としては、

・個人に直接的な被害を与えるような情報が流出している

→個人情報やクレジットカードの情報の漏洩

以上が挙げられる。解答としては以上のポイントを書き込めれば十分だと思われる。

(2)（解答例）

個人情報保護法と事業者。情報漏洩を防止するための安全管理や、それに伴う内部や委託先の監督の義務を怠っている。不正アクセス防止法とハッカー。ハッキングによる不正なアクセスを行っている。（91文字）

・解説

　プライバシーやセキュリティに関連する法律についての知識を求める問題である。

**～個人情報保護法と不正アクセス防止法～**

・**個人情報保護法**

→**個人情報の有用性に配慮しながら個人の権利・利益を保護**することを目的としたもの。主に**情報を保持する者の義務**を定めている。

・**不正アクセス防止法**

→**不正アクセスを禁止**することで情報社会の発展に寄与することを目的としたもの。主に**情報にアクセスする者の禁止事項**を定めている。

→より詳しい説明は教科書p.243

問題は「ハッキングによる情報流出」に関わってくる法律とその法律の対象者と行為を聞いている。問題からすぐに分かるものとして、以下の二つにまとめられるだろう。

|  |  |  |
| --- | --- | --- |
| **当該法** | **対象者** | **対象行為** |
| 個人情報保護法 | 事業者 | 安全管理、内部や委託先の監督 |
| 不正アクセス防止法 | ハッカー | ハッキング |

これらをまとめることができれば十分であろう。

→ちなみに、この問題の解答を「それぞれ」100文字で書くのか、「それぞれまとめて」100文字で書くのかが分からなかったので、まとめる方向で書きました。それぞれで100文字書こうと思ったら挫折しましたorz

## B

計算問題なんか大嫌いだ。

(1)解答・解説

(1a)

・解答：**3bit**

・解説

　「情報量（注14）」ならびに「平均情報量（注15）」という概念に対する理解を求める問題である（それぞれの詳しい説明、求め方は注釈参照）。

　問題は「面1から面8のどの面が上になるかについての平均情報量」を聞いている。つまり、「面1が上になる」、「面2が上になる」、（……）、「面8が上になる」という8個のメッセージが持つそれぞれの情報量を求め、そこから平均情報量を求めれば良いのである。

　このとき、**各面が等確率で上になる＝それぞれのメッセージが発せられる確率は等確率である**ため、それぞれのメッセージが発せられる確率は である。

**メッセージが等確率で発せられる場合**の平均情報量は、**個々のメッセージの情報量**と同じになることに注意すれば、答えは以下の通り。

**平均情報量 ＝ 個々のメッセージの情報量 ＝ ＝ 3(bit)**

　理解していればなんてことはない、簡単な問題である。

(1b)

・解答：**3.25**

・解説

　符号化に対する理解を求める問題である。ちなみに、同じような問題は06年度までの過去問には存在しなかった。要注意。

　問題文曰く、平均符号長＝符号数の期待値らしいのでおとなしくそれに従う（注16）。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 面 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 符号 | Ⅰ. | ⅠⅠ. | ⅠⅠⅠ. | ⅠⅤ. | Ⅴ. | ⅤⅠ. | ⅤⅠⅠ. | ⅤⅠⅠⅠ. |
| 符号数 | 2 | 3 | 4 | 3 | 2 | 3 | 4 | 5 |
| 発生確立 |  |  |  |  |  |  |  |  |

さて、求める平均符号長は上の表より、

＋＋＋＋＋＋＋

＝ ＝ **3.25**

となる。

なお、２＝Ⅱとかでで示せばいいじゃん、とか言うのは無粋である。

(1c)

・解答

表２の穴埋めは以下

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 面 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 符号 | BAB | BBAB | BBBAB | BAABAB | AABAB | AABBAB | AABBBAB | AABBBBAB |

平均符号長：**5.5**

・解説

~~作業ゲー乙。~~符号間の対応関係をちゃんと把握していれば解ける問題である。間違いを防ぐために、表１の下に符号を書いていった方がいいかもしれない。

平均符号長の求め方は~~面倒くさい~~上と同じなので割愛。

(1d)

・解答例：1をAAA、2をAAB、3をABA、4をABB、5をBAA（安全マークはついてるかい）、6をBAB、7をBBA、8をBBBとすればよい。そうすると平均符号長は3となり、平均情報量と合致する。

・解説

　数学的発想力が問われる問題である……といいたいところだが、試験中に浮かぶのはAを0、Bを1に見立てた二進法ぐらいだろう。~~発想力もクソも無い。~~計算方法はやっぱり(1b)と同じなので割愛。平均情報量は平均符号長の下限であるため、これ以外の解答は存在しないと思われる。比較しろ、というのはこれでいいのかな……？

→他に解法が思いついた方は、こっそりほくそ笑んで下さい。

(2)

(2a)

・解答：**１.984375(bit)**

・解説

~~処理ゲー乙。~~表を作って求めるのが手っ取り早い。確率の逆数の対数が情報量であることをお忘れ無く。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 面 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 確率 |  |  |  |  |  |  |  |  |
| 情報量 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 7 |

平均情報量は、それぞれの「確率×情報量」の総和であるから、  
＋＋＋＋＋＋＋

＝＝**1.984375(bit)**（なお、結果はエクセル計算）

　まさか小数点6位まで求めさせるとは、恐るべし。

(2b)

・解答：(2a)で求めた平均情報量が1.984375bitであるのに対し、(1a)で求めた平均情報量は3bitである。(2a)より(1a)が大きくなるのは、それぞれのメッセージが発せられる確率がばらばらであるからだと考えられる。

・解説

　比較→議論(´・ω・`)ﾅﾝﾉｺｯﾁｬ?　よく分からなかったので適当なことを書いておきました。正しい解答が分かる人は、「m9(^Д^)ﾌﾟｷﾞｬｰ」と笑ってクラスメイトに教えてあげて下さい。

(2c)

解答：1をB、2をAB、3をAAB、4をAAAB、5をAAAAB、6をAAAAAB、7をAAAAAAB、8をAAAAAAAとすればよい。

解説

　それぞれのメッセージが発せられる確率がばらばらの場合、**発せられる確率が高いメッセージの符号を少なくし、発せられる確率が低いメッセージの符号を多くすれば良い**。そのような符号化を、**ハフマン符号化（注17）**という。

　さて、この符号化によって本当に平均符号長は最低になっているのか？　~~検証はめんどくさいので各自に任せまｓ~~３ビットの符号化と対比してみよう。

～3ビットの符号化の場合～

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 面 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 符号 | AAA | AAB | ABA | ABB | BAA | BAB | BBA | BBB |
| 確率 |  |  |  |  |  |  |  |  |

＋＋＋＋＋＋＋

＝＝**2.8125(bit**)

～ハフマン符号化の場合～

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 面 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 符号 | B | AB | AAB | AAAB | AAAAB | AAAAAB | AAAAAAB | AAAAAAA |
| 確率 |  |  |  |  |  |  |  |  |

＋＋＋＋＋＋＋

＝＝**1.984375(bit)**

以上の結果より、ハフマン符号化の方が、平均符号長が短いことが分かる。また、平均符号長＝平気情報量であるため、これ以上短い平均符号長は無いことが分かる。

**終わりに**

　特に問題全体の総括とかする時間も気力も無いので、最後に一言。

**6割狙いでおｋ。**

最後の最後で思いっきり手を抜きました、すみません。

ここまで読んでいただいた皆さん、ありがとうございました。拙い上に長ったらしい文章でごめんなさい。反省はしてません。

では、頑張って下さい。

**注釈**

注1：まず、IPについて説明する。IP（Internet Protocol）とは、**異なるネットワーク間で情報をやりとりする際のプロトコル**である。このプロトコルのもとで、情報はパケット（細切れの状態）にされて送り先へ送られる。この際必要となるのがIPアドレスである。

注2： IPアドレスは、**IPに則って、コンピュータの所在を数字で表したもの**で、情報の送信者を特定するのに使われたりする。2chとかで本人特定の際に使われるのはこの情報である（多分）。

→下手に公開するとハッキングされちゃうかもしれない。

注3：まず、**ホスト名**は**人間に分かるように、コンピュータの所在を文字列で表したもの**であり、**ドメイン**はその**ホスト名を構成する文字列のこと**である。ドメインは“.（ピリオド）”によって区切られる。

例）www.u-tokyo.ac.jp

上の例でいえば、「www.u-tokyo.ac.jp」がホスト名であり、「.jp」が日本に所在することを示すドメイン名、「.ac」は公的機関であることを示すドメイン名、「u-tokyo」はどこの機関かを示すドメイン名である。

→**ホスト名が人間向けに表されたコンピュータの所在情報**で、**IPアドレスはコンピュータ向けに表されたコンピュータの所在情報**である。

→ホスト名とドメイン名の区別は間違えがちです、というか私は間違えました。皆さん気をつけましょう。

注4：共通鍵暗号方式の利点は、**復号する際に時間がかからない**ところにある。欠点としては、まず**共有する相手毎に共通鍵を用意する必要がある**こと。相手が増えるたびに管理すべき共通鍵の数が増えてしまう。そして、**共通鍵を厳重に管理しなければならないこと**。共通鍵が第三者に漏洩すると、その鍵を共有している相手とのデータのやりとり全てが筒抜けになってしまう。

注5：公開鍵暗号方式の利点は、**セキュリティの強固さ**にある。公開鍵から秘密鍵を作り出すことは、可能ではあるものの、実際にかかる時間が年単位であるため実用的ではない。欠点は、**復号の際に時間がかかる**ことである。

注6：現在通信回線を確立する方式には二つ種類がある。「回線交換方式」と「パケット交換方式」である。詳しい説明は教科書p.52を参照のこと。

注7：プロトコル（Protocol）は、**情報をやりとりする際の約束事**。通信規約。

→「名前は？」と聞かれたら自分の名前を答えるのは一種の約束事である。ここで回答側が持つプロトコルが異なっていたりすると、「いい天気ですね」と回答側が答えたりするため、会話が成り立たなくなる。

→ついでに言えば、オタクと一般人のそりが合わないのはそれぞれ持つプロトコルが違うから、とか思ってみる。

→慣習と近いものがあるかもしれない。

注8：我々が普段見ているウェブページは、**HTMLやXMLで作られた（記述された）ファイル**である。このファイルのことを、**ハイパーテキスト**（hypertext）という。

→一つ余計なことを言えば、WordやExelのファイルもHTMLやXMLで作られている。保存の際に拡張子を「～.docx」から「～.zip」にして解凍してみるとそのことが確かめられる。

注9：ちなみに、HTMLは**マークアップ言語**である。マークアップとは、**普通の文に（タグで囲む等して）目印を付け、その文が文章中でどのような働きを示すこと**。

例)）<a href="http://todai.info/">UTaisaku-Web<a>

→「<a href=～> <a>」がタグで、「UTaisaku-Web」という文がマークアップされている文。

「UTaisaku-Web」という文が、http://todai.info/へのリンクであるということを示している。

注10：ウェブブラウザーには様々な種類があり、Windowsの“Internet Explorer”、Macの“Safari”がそれに当たる。これらプリインストールされているブラウザは概して性能が悪いので、慣れた人はFirefox（Mozilla Foundation）、Chrome（Google）などのサードパーティーが開発したブラウザを使う。ただし、プリインストールされているブラウザに関しては、サードパーティー開発のブラウザの影響を受けて改善が進み、だんだん実用に耐えられるレベルになってきている。**ただしIEはクソ**。異論は認めない。

注11：なお、ドメイン名とURLの区別は以下の通り。

例）http://todai.info/sikepuri/search/（UTaisakuのシケプリ検索ページ）

**“todai.info”がドメイン名**で、**URLが“http://todai.info/sikepuri/search/”全体**である。

注12：インターフェースの説明はあまりにも長くなるためここでは割愛する。教科書p211を参照のこと。

注13：データモデルにはいくつか種類があり、集合モデル、ネットワークモデル、階層モデルなどがそれにあたる。それぞれの詳細は教科書p.79を参照のこと。

注14：まず、ある現象に対する曖昧さを減少させるものとして、**メッセージ**がある。さて、**情報量**とは、「**メッセージが受け手に与えるなんらかの量**」であり、定義の仕方によって求め方が変わる。

**～場合の数の変化から求める～**

情報量を、「メッセージが、**行動の曖昧さを減少させる度合い**」を示すものとする（例：試験範囲）。そうすると情報量は、「メッセージを**受け取る前の場合の数**」を「情報を**受け取った後の場合の数**」で割ったものの**対数**として表される。

つまり、

**情報を受け取る前の場合の数**

**log2** **情報を受け取った後の場合の数**

である。

また、求められた情報量はそれぞれ**足し合わせることができる**。

**～メッセージ内容の発生確率から求める～**

情報量を、「メッセージが持つ**内容の珍しさ**」を示すものとする（例：ヒッグス粒子の発見）。そうすると情報量は、**「そのメッセージが発せられる確率」の逆数の対数**で表される。

つまり、

**log2 （そのメッセージが発せられる確率の逆数）**

である。

この情報量も上と同じく足し合わせることができる。

情報量の単位は“**bit**”である。これはコンピュータのメモリなどで使われる方の“bit”とは切り離して考えた方が、頭がこんがらがらなくて済む。

なお、一般的には**対数の底には2**が使われている。試験の際には必要な数値が示されるので、対数表を覚えたりする必要は無い。

より詳しい説明は、教科書p.40を参照のこと。

なお、**人生に於ける「情報量」の情報量は、限りなく0に近い**。←ココ重要。テストに出るぞー（殴

注15：メッセージ全体を、メッセージ1、2、3、（……）、nの集まりとする。そうすると、メッセージ全体の**平均情報量**の求め方は、

**「1が発せられる確率」×「1が持つ情報量」＋「2が発せられる」×「2が持つ情報量」＋「3が発せられる確率」×「3が持つ情報量」＋（……）＋「nが発せられる確率」×「nの情報量」**

となる。つまり、「個々のメッセージが持つ**情報量**」を、「個々のメッセージが発せられる**確率**」でかけ算し、**その総和**を求めるのである。期待値の求め方と同じである。

　また、**メッセージが等確率で発せられる場合**の平均情報量は、**個々のメッセージの情報量**と同じになる。

より詳しい説明は、教科書p.42を参照のこと。

注16：教科書の説明よりこちらの方がわかりやすいので、平均符号長については問題の方の記述で覚えた方がいいかもしれない。なお、平均符号長のさらなる説明については教科書p.44以降を参照のこと。

注17：ハフマン符号化とは、**二種類の符号（Ａ・Ｂ）でメッセージを符号化**することを考えた場合、**一番発せられる確率が高いメッセージ**を**B一文字**で表し、**一番発せられる確率が低いメッセージ**を**Aのみの文字列**で表し、**それ以外のメッセージ**を、**Aのみの文字列とB一文字**で表すものである。

つまり、確率がばらばらなn種類のメッセージを確率の高い順に並べ替え、それぞれをMk(k=1、2、3、4、(…)、(n-1)、n)として符号化すると、

(k=1)　B

(k=2)　AB

(k=3)　AAB

(k=4)　AAAB

…

(k=n-1)　A×(n-1)B

(k=n)　A×n

となるような符号化のことである。