

大学入学共通テスト「情報」 対策セミナー

2023年12月23日

アシアル株式会社
アシアル情報教育研究所



本セミナーの目的と概要

2025年1月の共通テストから『情報』が追加される

本研修の目的

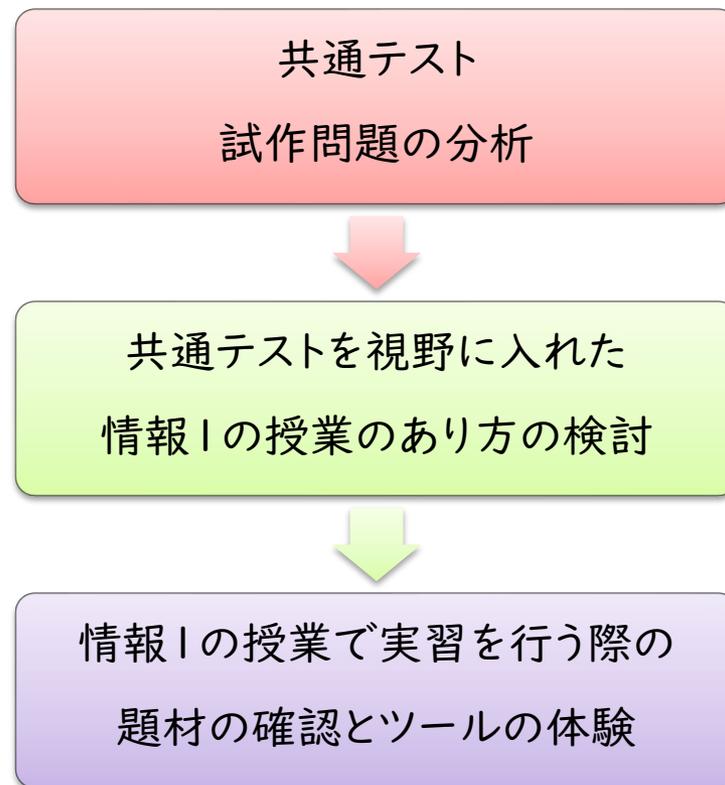
- 共通テストを視野に入れた、情報科の授業のあり方を考える

本研修・資料の概要

- 共通テスト「情報」で出題される問題について、公表されている資料を元に分析する
- 共通テストを視野に入れる場合、どのような授業があり得るか検討し、実習を行う意義を確認する
- 授業で実習を行う場合に利用できる素材、ツールを実際に使ってみる
 - Monaca Education
 - 論理回路シミュレータ
 - 表計算ソフト など

※本研修で扱わないこと

- そのまま利用できる授業案は提供しない



スケジュール

	時間	内容
共通テスト 試作問題の分析と 実習中心の授業のアイデア	13:00 - 14:00	共通テストの試作問題の分析 <ul style="list-style-type: none">試作問題のダウンロード方法の確認各問題の概要と回答のポイント・学習のポイント・実習中心の授業のアイデア<ul style="list-style-type: none">第1問 情報社会の問題解決・コンピュータの仕組み・情報デザインなど第2問 A コンピュータの仕組み、B モデル化とシミュレーション第3問 プログラミング第4問 データの活用
休憩	14:00 - 14:10	
単元「プログラミング」の実習 の進め方	14:10 - 15:00	共通テストの試作問題 第3問 プログラミングの分析 <ul style="list-style-type: none">概要と回答のポイント・学習のポイント Monaca Educationを用いた、実習によるプログラミングの学習方法の紹介
振り返りとまとめ	15:00 - 15:10	<ul style="list-style-type: none">挨拶アンケートの記入事務連絡 など

第1部

	時間	内容
共通テスト 試作問題の分析と 実習中心の授業のアイデア	13:00 - 14:00	共通テストの試作問題の分析 <ul style="list-style-type: none">試作問題のダウンロード方法の確認各問題の概要と回答のポイント・学習のポイント・実習中心の授業のアイデア <ul style="list-style-type: none">第1問 情報社会の問題解決・コンピュータの仕組み・情報デザインなど第2問 A コンピュータの仕組み、B モデル化とシミュレーション第3問 プログラミング第4問 データの活用

資料は全4問を解説していますが、今日は特に次の2問に時間を取ります

- 第1問 問2
パリティビット
- 第4問
データの活用

共通テスト 試作問題の分析



共通テスト 試作問題とは①

大学入試センターが2022年11月に公表。

ダウンロード可能

試作問題及び試作問題の概要等

https://www.dnc.ac.jp/kyotsu/shiken_jouhou/r7/r7_kentou_joukyou/r7mondai.html

The screenshot shows the website for the University Entrance Center (DNC). The page title is '令和7年度試験の問題作成の方向性、試作問題等'. It features a navigation menu at the top with categories like '受験者・保護者の方', '高校関係者の方', '大学関係者の方', '研究者の方', and '関連・入札関係の方'. The main content area includes a breadcrumb trail, a search bar, and a list of links for downloading documents. Below this, there is a section titled '試作問題及び試作問題の概要等' which contains a table summarizing the test questions and their solutions.

	試作問題の概要	試作問題	正解表
国語	概要『国語』(164.5 KB)	試作問題『国語』(2.0 MB)	正解表『国語』(59.5 KB)
地理歴史		試作問題『地理総合、地理探究』(7.8 MB)	正解表『地理総合、地理探究』(61.3 KB)
		試作問題『歴史総合、日本史探究』(7.5 MB)	正解表『歴史総合、日本史探究』(62.8 KB)
	概要『地理歴史』(476.5 KB)	試作問題『歴史総合、世界史探究』(5.4 MB)	正解表『歴史総合、世界史探究』(64.2 KB)

共通テスト 試作問題とは②

資料作成時点(2023年10月)で、最も共通テストの問題に近い
⇒分析して、授業の準備に使う意義が高い

【補足】

上記Webページには、他に以下の2つの資料がリンクされており、作問の方針が分かるようになっている

- 『令和7年度大学入学者選抜に係る大学入学共通テストの 問題作成方針に関する検討の方向性について』
- 『令和7年度大学入学者選抜に係る大学入学共通テストの 出題教科・科目の問題作成方針に関する検討の方向性について』

共通テスト サンプル問題とは

大学入試センターが2021年に公表したもの

(参考) 試作問題等令和4年度までの検討状況

https://www.dnc.ac.jp/kyotsu/shiken_jouhou/r7/r7_kentoujoukyou/

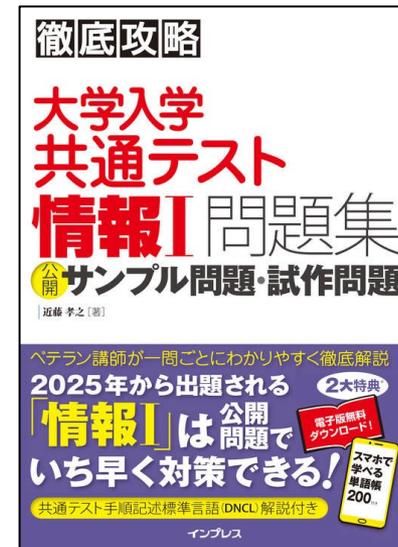
- 「試作問題」の方が、「サンプル問題」より新しい
→「試作問題」の方が、実際の共通テストに近くなっていると考えられる
- 上記の通り、Webページのタイトルにも「(参考)」とあり、過去の情報として位置付けられている

参考) 市販書籍

『徹底攻略 大学入学共通テスト 情報Ⅰ 問題集 公開サンプル問題・試作問題』

(近藤孝之著、株式会社インプレス)

- サンプル問題・試作問題の解答と解説をまとめたもの
※解答は公開されています
- 巻末付録に用語集がつく



今後(2024年2月頃~)、教科書会社・参考書出版社による、同じ水準の問題を扱う問題集の出版が見込まれる

共通テスト 試作問題 第1問



試作問題

第1問 問1 インターネットの利用

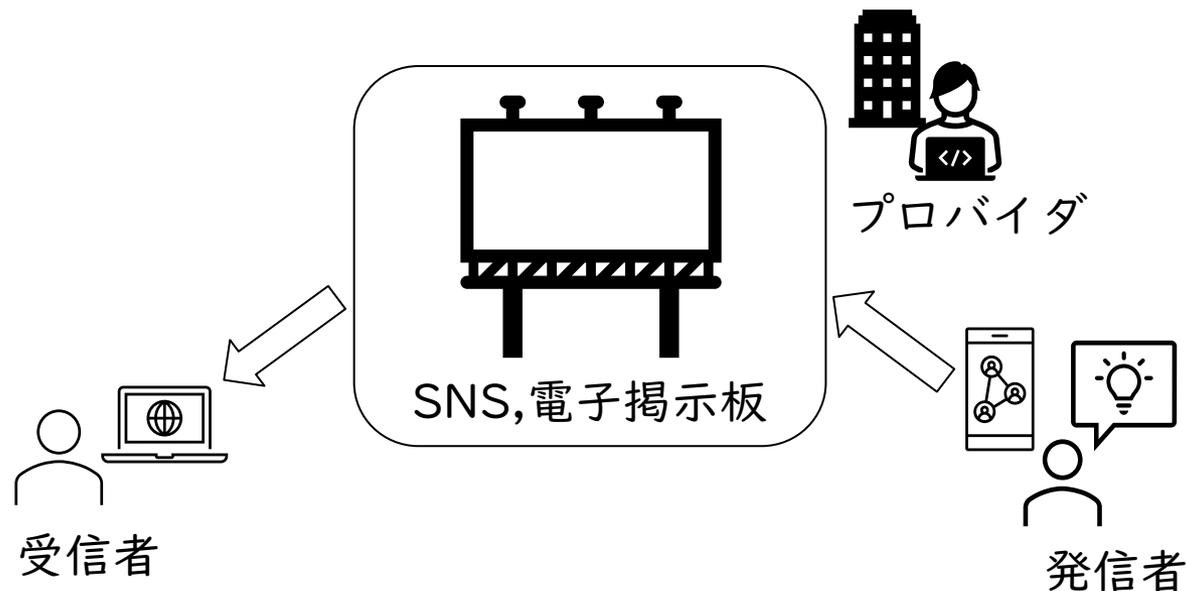


試作問題 第1問

問1 インターネットサービスの安全な利用

問題

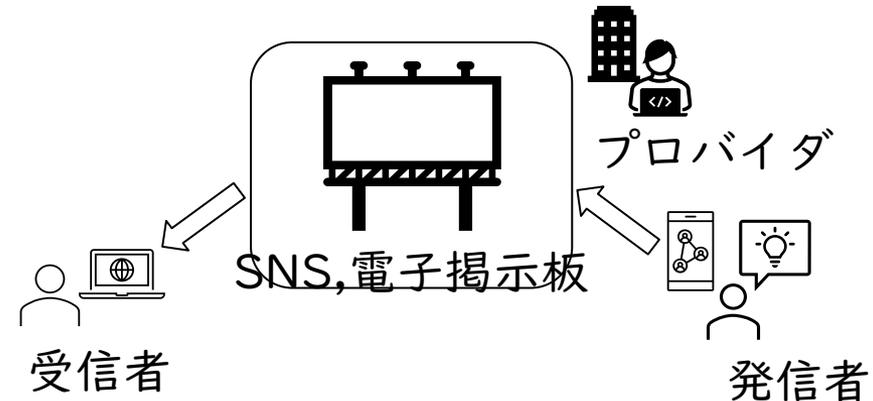
- SNSやメール、Webサイトを利用する際の注意や判断を問う。**自身の安全**や**他者の権利**の侵害への配慮ができるかどうか
- **情報の信憑性**を確かめる方法を問う



試作問題 第1問

問1 インターネットサービスの安全な利用

解答するためのポイント



- 高校生も身近に利用しているインターネットサービスについて判断する
 - 仕組みと、**関係者**に目を向けることができるか
 - 安全に利用するための**基本的な注意点**が想起できるかどうか
 - 自身の安全や利便性だけでなく、**他者の権利**に配慮できるかどうか

学習のポイント

- 普段利用している身近なサービスについて、**用語**を使って解像度を高めて理解する
- 具体的な**事例**と、教科書で取り上げられている用語とを関連付ける

試作問題

第1問 問2 パリティチェック



試作問題 第1問

問2 パリティチェック①

問題文に含まれる資料

- パリティビットを追加する方法(データの中の"1"の個数を数えて、偶数個なら0、奇数個なら1のパリティビットを付ける)

問題

1. このパリティビットによって、判定できるのはどのような誤りか
2. ある16進法で表現されたデータを、2進法表記にし、パリティビットを付けると、どのようなデータになるか

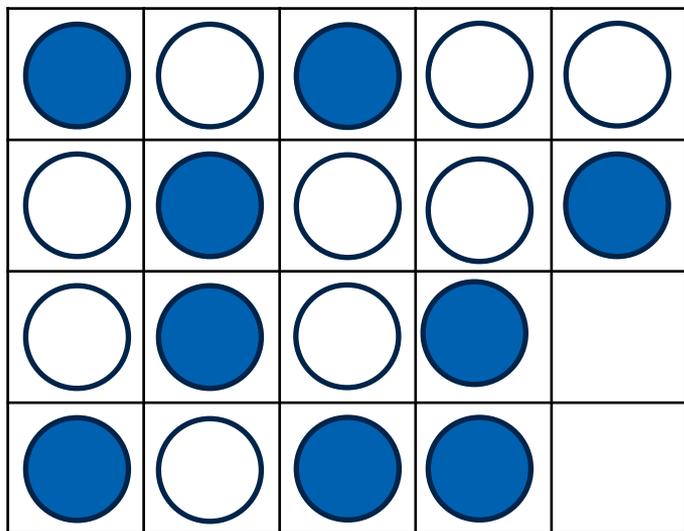
試作問題 第1問

問2 パリティチェック②

パリティビットの考え方を、2色のマグネットで確認する



濃い色のマグネット: 1
薄い色のマグネット: 0
と見なす



濃色(●)が2個(偶数個)なので、パリティビットは淡色(○)



濃色(●)が1個(奇数個)なので、パリティビットは濃色(●)



濃色(●)が□個(□数個)なので、パリティビットは□(□)



濃色(●)が□個(□数個)なので、パリティビットは□(□)

考えてみよう

試作問題 第1問

問2 パリティチェック③

パリティビットを使って、間違い探しをする

データ				パリティビット
●	○	●	○	○
○	●	○	○	●
○	●	○	●	●
●	○	●	●	●

前の問題で、
計算してパリティビットを付けた。
この問題では、パリティビットで
データを検証する

Q1) データに誤りが混入したのは、どの行ですか？

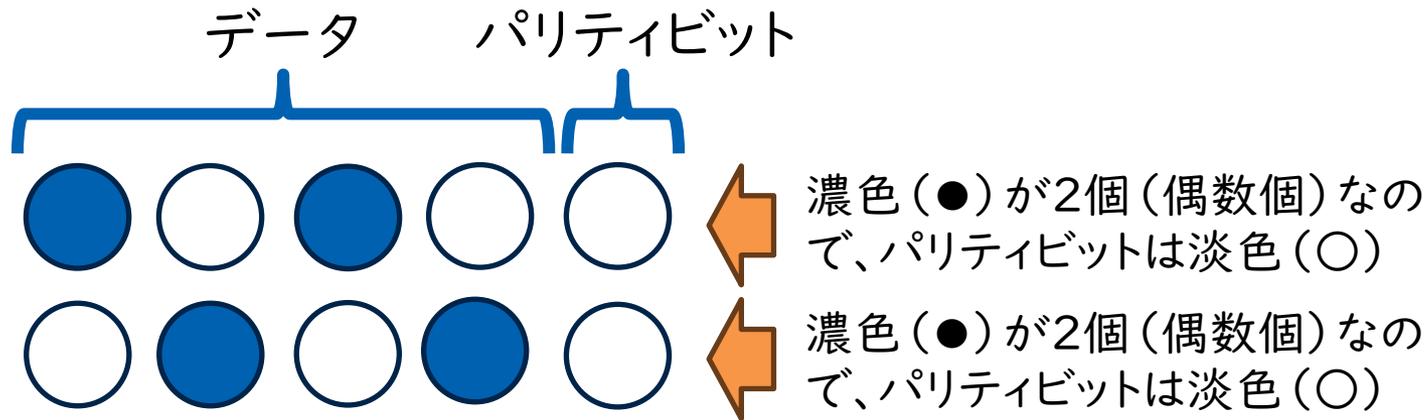
Q2) 誤りのある行について、訂正はできるでしょうか？

データの中の"1"の個数を数えて、
偶数個なら0、奇数個なら1のパリティビットを付ける

試作問題 第1問

問2 パリティチェック④

パリティビットの限界を確認する



データの並び(4つのマグネットの並び)は全然違うのに、
パリティビットは同じ色

- ⇒ データが間違っって並べ替えられても、誤りは発見できない
- ⇒ 偶数個の誤りは、検出できない

試作問題 第1問

問2 パリティチェック⑤

解答するためのポイント

- 送信したデータに混入する「誤り」には、複数のパターン(場合)があることを思い起こした上で、**場合分け**して、一つ一つの場合について「問題文のパリティビットの仕組みで、誤りが検出できるか・位置を判定できるか」を考える
- 16進法の値を**2進法に変換する**
- 2進法の列に(問題に挙げられている方法で)パリティビットを付ける

学習のポイント

- ビット、パリティチェックなどの基本概念を知っている
- 2進数、16進数、十進数の間の**基数変換**ができる

状況に合わせて、段階を踏んで学習を進める

1. データを送受信する

- スイッチのオン・オフ、「0」「1」の2種類の値で、信号が送れる

コンピュータの
仕組み
(2進数)

情報通信
ネットワーク

2. データにパリティビットを付けて、誤り検出できるようにデータを送受信する

- どのような誤りなら見つけられるか
- どのような誤りは見つけられないか

情報通信
ネットワーク

3. 応用

- (誤り検出とは別の) **誤り訂正**の仕組みを理解する
- 「データにパリティビットを付ける」というアイデアを応用した例を見つける

共通テスト
対策

試作問題

第1問 問3 論理回路と真理値表

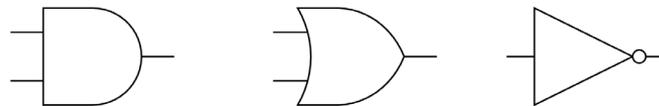


試作問題 第1問

問3 論理回路と真理値表①

問題文に含まれる資料

- 3種類の論理回路（論理積AND、論理和OR、否定NOT）の図記号と、真理値表



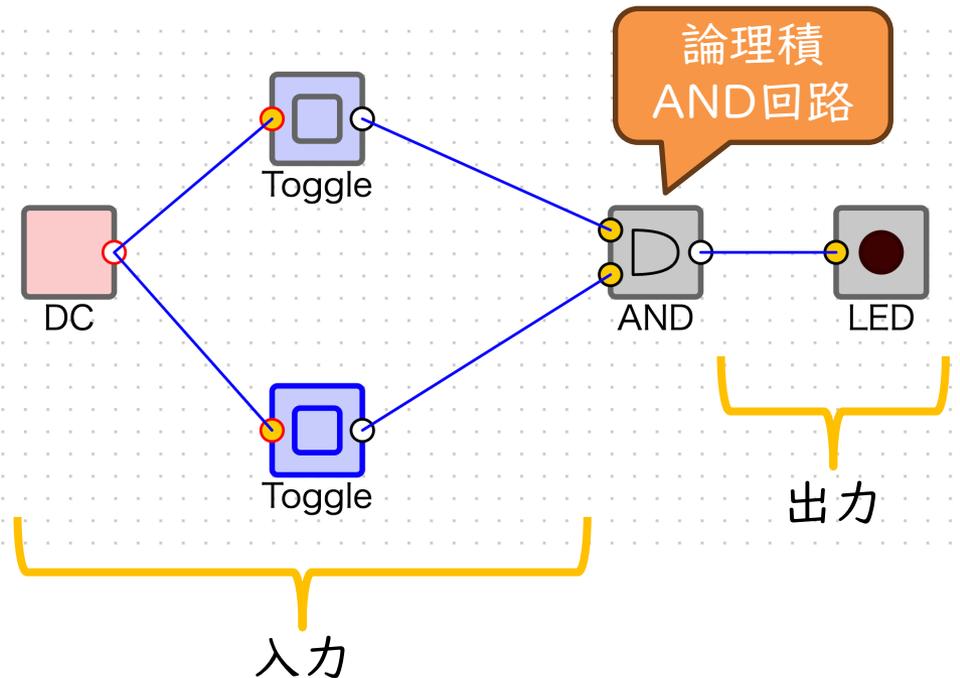
問題

- 「二つのトイレの両方が使われているとき、ランプを点灯する」という振る舞いをする論理回路を選択肢から選びなさい
- 「三つのトイレのうち二つが使われているとき、ランプを点灯する」という振る舞いをする論理回路を完成させなさい
- 「三つのトイレのうち二つが使われているとき、ランプを点灯する」という振る舞いをする論理回路の真理値表を選びなさい

試作問題 第1問

問3 論理回路と真理値表②

論理回路



- 論理回路部品に、電源とスイッチ (=入力) と、LEDライト (=出力) を組み合わせると、真理値表の意味が分かりやすくなる
- 論理回路の動作は、シミュレータで**体験**できる

体験しないと
抽象度が高く見えるので、
体験することを推奨します

試作問題 第1問

問3 論理回路と真理値表③

入力		論理回路の動作			出力
A	B				X
偽 0 オフ	偽 0 オフ	スイッチ 上:オフ 下:オフ		LED 消灯	偽 0 オフ
真 1 オン	偽 0 オフ	スイッチ 上: オン 下:オフ		LED 消灯	偽 0 オフ
偽 0 オフ	真 1 オン	スイッチ 上:オフ 下: オン		LED 消灯	偽 0 オフ
真 1 オン	真 1 オン	スイッチ 上: オン 下: オン		LED 点灯	真 1 オン

試作問題 第1問

問3 論理回路と真理値表④

解答するためのポイント

- 文章で表現された機構の振る舞いを、論理回路として抽象化して考える
- 文章で表現された機構の振る舞いを、真理値表として抽象化して考える

学習のポイント

- 論理積、論理和、否定について、**入力と出力を真理値表にまとめることができる**
- 基本の論理回路を組み合わせて、複雑な回路を作ったときに、**入力の組み合わせによって出力が変わることが分かる**
- 論理回路について、入力と出力の組み合わせを**真理値表に整理**できる

論理回路と真理値表

意図通りの動作をする
機械を作るために考える

- 完成している**真理値表を読み**、入力からどんな出力が得られるか(=どんな振る舞いをするか)、**言葉で表現できる**
 - 例: 「**三つのトイレのうち二つが使われているとき、ランプを点灯する**」
→ 別の表現をすると?

3人で行う多数決

(参考)

Monaca Educationスタンダードプラン

サポートページ

<https://edu.monaca.io/standard>

※有償プランご契約の学校様に、パスワードをお知らせしています

3章 コンピュータとプログラミング

- 教材・動画を公開しています



3章：コンピュータとプログラミング

本プランは標準時数100時間分の教材を用意しております。一部の単元はAPSを別途入手して下さい。

[Youtubeで関連動画をみる](#)

単元名称	概要	標準時数	教材
コンピュータの仕組み (2進数)	論理回路シミュレータなどを利用して2進数を学びます。	1	ダウンロード
コンピュータの仕組み (16進数)	16進数で色の配色を行ったり2進数との相互変換を行います。	1	ダウンロード
コンピュータの仕組み (論理回路)	AND・OR・NOT回路などを論理回路シミュレータで学びます。	1	ダウンロード
アルゴリズム 探索	アルゴリズムの例として探索を学習します。	1	ダウンロード
アルゴリズム ソート	バブルソートを題材にソートを学習します。	2	ダウンロード
プログラミング入門	印刷教材のPython or JavaScriptから選択してご利用下さい。	6	-
モデル化とシミュレーション	APSの権利計算アプリをご利用下さい。	2	-
プログラミングの実践 (オリジナル作品作り)	APSのおみくじアプリ等をご利用下さい。	4	-

(参考)

Monaca Educationスタンダードプラン 動画『3章：コンピュータとプログラミング』

非公開のYouTube再生リスト

1本あたり5分から10分程度の動画

1. アルゴリズム バブルソート ①～④
2. アルゴリズム 探索 ①～④
 - 二分探索
 - 計算量
3. コンピュータの仕組み(論理回路) ①～⑥
 - 論理演算
 - 3つの論理回路
 - 論理回路シミュレータ
 - 真理値表
 - 半加算回路



試作問題

第1問 問4 情報デザイン



試作問題 第1問

問4 情報デザイン①

問題文に含まれる資料

- 情報を整理し、表現するための整理法「究極の5つの帽子掛け」の5つの基準の説明と、その例
- 情報デザインの例: 鉄道の路線図、宿の満足度評価ランキング

問題

- 「鉄道の路線図」および「宿の満足度評価ランキング」について、整理法がどのように適用されているか説明する文章について、選択肢から適切な言葉を選ぶ

試作問題 第1問

問4 情報デザイン②

究極の5つの帽子掛け

「どんな情報でも、1つの側面に注目して**抽象化**したら、5つの規準で分類・整理できる」というアイデア

場所	アルファベット	時間	カテゴリー	階層
• 都道府県の人口	• 辞書	• 歴史年表	• 生物の分類	• 重要度順のToDoリスト

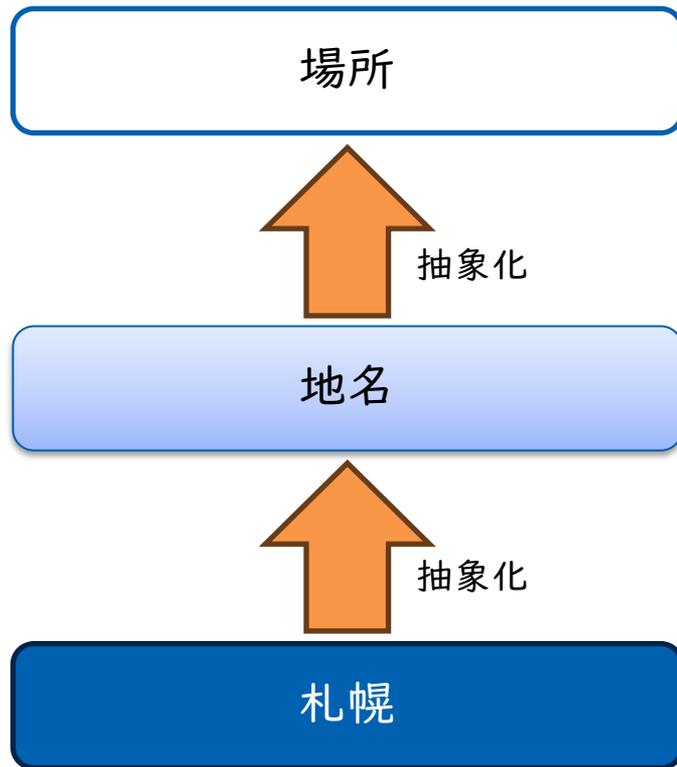
⇒「**鉄道の路線図**」「**温泉がある宿の満足度評価ランキング**」はどれに当てはまるか？

⇒具体的な値（駅名、ホテル名など）から、「**一般化されたルール**（5つの帽子掛け）のどれに当てはまるか」を考えるには、具体的な値を**抽象化・一般化**して、ルールと同じ程度の抽象度の言葉に言い直す必要がある

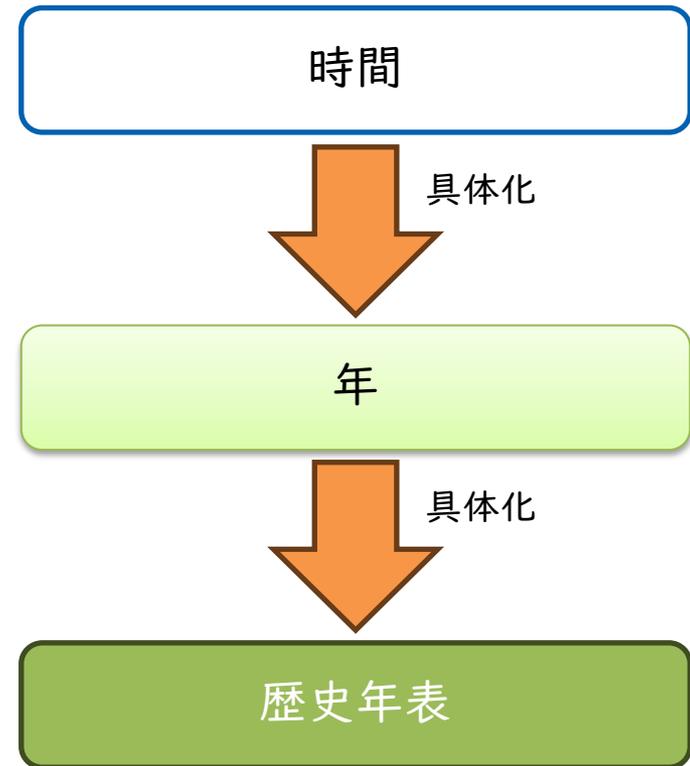
試作問題 第1問

問4 情報デザイン③

【具体的⇨抽象的】



【抽象的⇨具体的】



試作問題 第1問

問4 情報デザイン④

解答するためのポイント

- 「鉄道の路線図」には、どのような**具体的な値**が含まれているか読み取る
 - 例：さっぽろ、大通…
- 読み取った値は、どのような概念なのか、**抽象化**する
 - 例：駅名、地名…
- 抽象化した概念は、5つの基準のどれに当てはまるか対応を調べる

学習のポイント

- 抽象化された図の中から、表現されている**具体的な値**を取り出す
 - その値は、現実の事物の中のどの側面について取り出されたか判定する
- 
- 現実の事物について、1つの側面に注目して値を取り出す
 - その値を使って、**抽象化**した図を作成する

試作問題 第1問

問4 情報デザイン⑤

「～は、～の一種」

- (具体的な値)は、(抽象的な概念)の一種
- 「札幌」は、「地名」の一種
- 「地名」は、「場所」の一種



「～の例を挙げると、～」

- (抽象的な概念)の例を挙げると、(具体的な値)
- 「時間」による情報の整理の例を挙げると、「歴史年表」

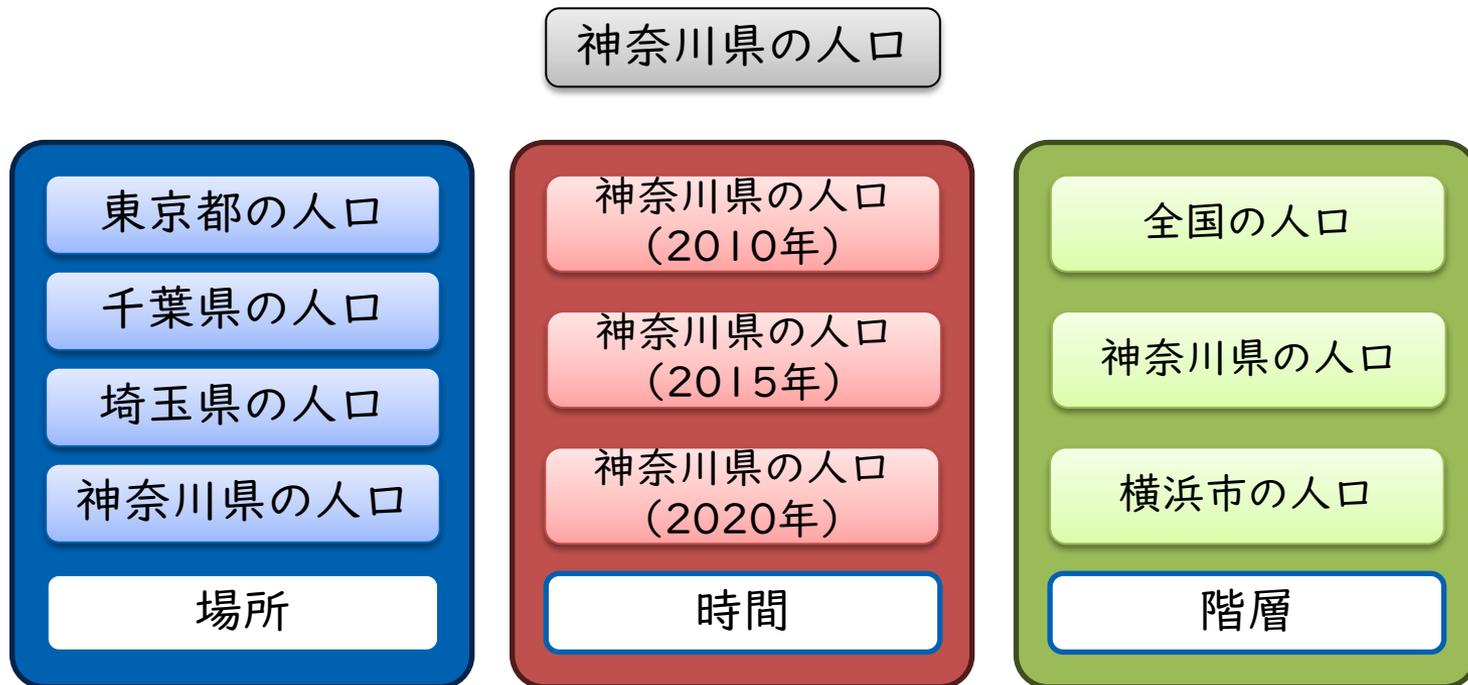


試作問題 第1問

問4 情報デザイン⑥

一般化や抽象化は、一通りではない

- 例:「神奈川県の人ロ」という情報は、集められている他の情報との組み合わせ次第で、色々な方法で分類・整理され得る



共通テスト 試作問題 第1問 まとめ

- インターネットの安全な利用、他者の権利の尊重など基本的な振る舞いについての問題
- 基礎的な知識や計算手順を問う問題
 - パリティチェック
 - ビット列
 - 基数変換
- 論理回路、真理値表を使って簡単な機構を考える問題
- デザインされた成果物から、どんな技法が使われているか問う問題

どの問題も、**詳細な知識は問題文に記載**があり、完璧に暗記しておく必要は無いらしいことが推測できる。

正しい解を求めるには、**知識をもとに、適切な考察や推論、計算が行える**必要がある。

共通テスト 試作問題 第2問



試作問題 第2問 A



試作問題 第2問

A 二次元コード①

問題文に含まれる資料

- 二次元コードの説明

- 特許
- 入れられる情報の量
- 形状の特徴（黒白の小さな正方形と、3箇所隅に二重の少し大きい正方形）
- 「符号化」概念と関連付けて、用語（セル）を紹介
- セルの個数（縦×横）の種類
- 3箇所隅の正方形の意味（=位置検出の目印）
- 4段階の誤り訂正機能（≠誤り検出機能）
- 二次元コードは自分でも生成できる

- ツールによって生成された二次元コードの例

- 文字数と、誤り訂正のレベルを組み合わせ、様々なパターンで生成したもの（※縦・横のセル数が変わっている）



二次元コードの例

※試作問題から引用

パリティビットの問題で扱ったのは、「誤り検出」

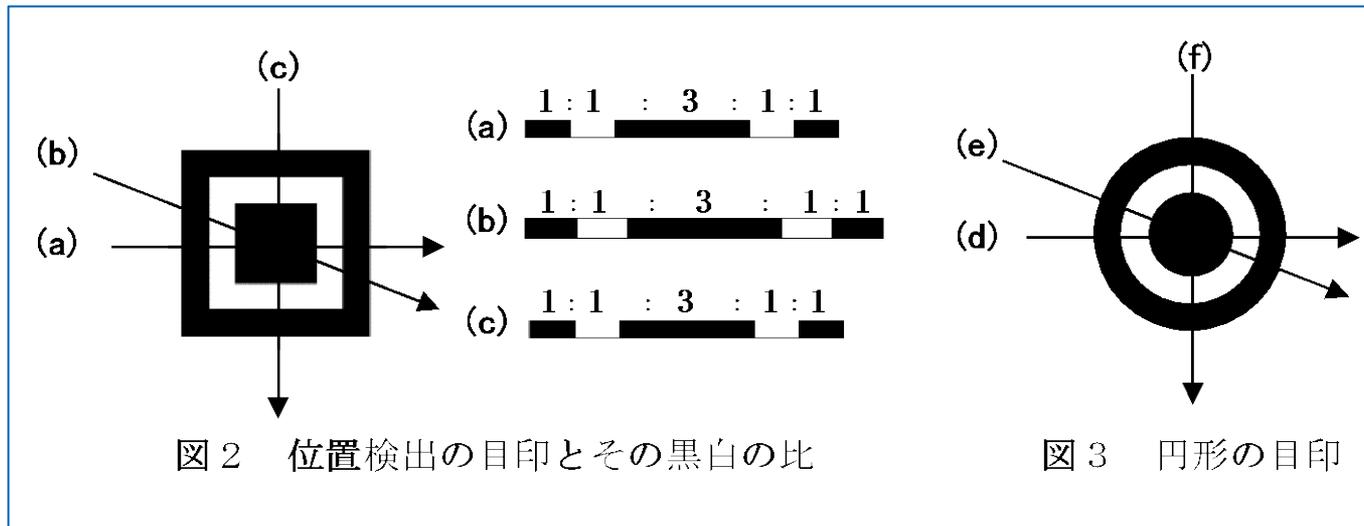
試作問題 第2問

A 二次元コード②

問題

1. 二次元コードが(※**特許**を取得していることを踏まえた上で)普及した理由を選ぶ
2. 3箇所での位置検出の目印が二重の正方形になっている理由を選ぶ
 - 図形を貫く直線を引いた時、線上の黒・白・黒・白・黒の色の比が決まっている

※試作問題から引用



試作問題 第2問

A 二次元コード③

問題

3. 二次元コードを生成するツールで、**入力するデータの量**(文字数)と、**誤り訂正のレベル**を指定する。このとき、生成される二次元コードのセルの縦・横の数はどうなるか。二次元コードの例から、正しく推定している文章を選ぶ

- 誤り訂正レベルが同じとき、入力するデータの量(文字数)はセルの数はどのように影響するか
- 入力するデータの量(文字数)が同じとき、誤り訂正レベルはセルの数はどのように影響するか

4. 入力するデータの量と誤り訂正レベルを**組み合わせ**たいくつかのパターンで二次元コードを生成する。結果、出来上がる二次元コードの縦・横のサイズを推定する

	15文字	20文字	30文字	40文字
復元能力 7%	 21×21	 25×25	 25×25	 29×29

※試作問題から引用

復元能力 7%	 21×21
復元能力 30%	 29×29

試作問題 第2問

A 二次元コード④

解答するためのポイント

- 知的財産権における**特許の位置付け**を踏まえた上で、普及することに意義がある技術を普及させる方法を挙げられる
- 2つのパラメータ(文字数と誤り訂正レベル)が、1つの出力(二次元コードのサイズ)にどのように影響するか、**図示された例を適切に解釈し、推定に使えるルールを文章化**できる

学習のポイント

- 「**複数の値を入力し、出力を得る**」というツールを利用してみる
 - 一方を固定し、他を変更して、出力を比較する
 - 変更していた値を固定し、固定していた方の値を変更して、出力を比較する
- 例:RGBによる色の生成ツール
https://anko.education/tool/app_rgb

試作問題 第2問 B



試作問題 第2問

B モデル化とシミュレーション①

問題文に含まれる資料①

- 状況説明
 - 文化祭でクレープを販売する
 - 文化祭は2日間の日程。1日目が終わりに、データが集まった
 - 1日目のデータを元にシミュレーションをして、2日目に臨む

試作問題 第2問

B モデル化とシミュレーション②

問題文に含まれる資料②

- 度数分布表

- 到着間隔 (秒)
- 人数
- 階級値
- 相対度数
- 累積相対度数

表1 到着間隔と人数

到着間隔 (秒)	人数	階級値	相対度数	累積相対度数
0 以上～ 30 未満	6	0 分	0.12	0.12
30 以上～ 90 未満	7	1 分	0.14	0.26
90 以上～150 未満	8	2 分	0.16	0.42
150 以上～210 未満	11	3 分	0.22	0.64
210 以上～270 未満	9	4 分	0.18	0.82

- 表計算ソフトの乱数生成を用いてシミュレーションした到着間隔の表

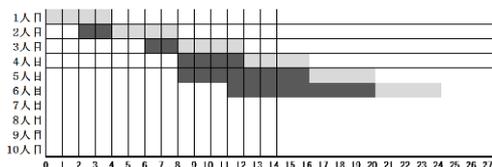
- 1人目から10人目までの到着間隔

表2 乱数から導き出した到着間隔

	生成させた乱数	到着間隔
1人目	—	0分
2人目	0.31	2分
3人目	0.66	4分
4人目	0.41	2分

- 到着時間、待ち時間、対応時間を表示した図

- 途中まで作成されている



※各表・図は試作問題から引用

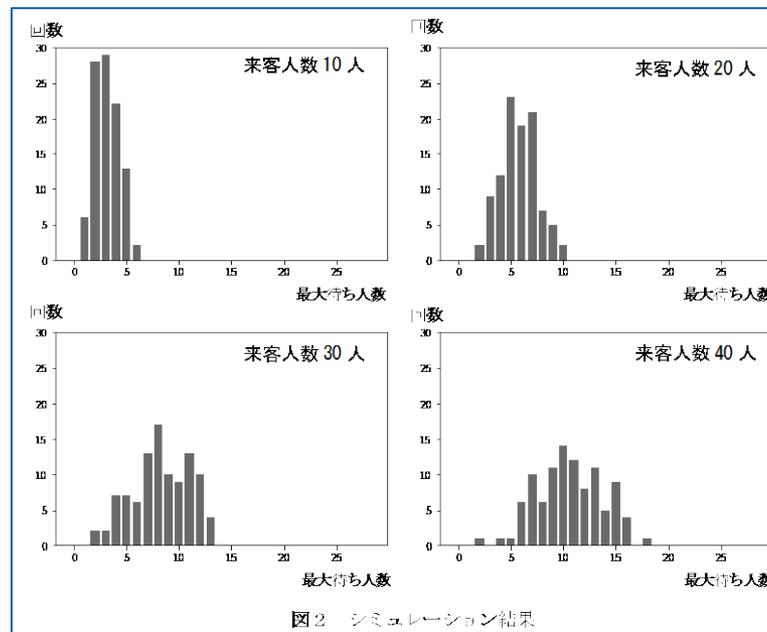
- 「到着間隔の表」を用いて、続きを作図できる

試作問題 第2問

B モデル化とシミュレーション③

問題文に含まれる資料③

- 来客人数ごとにシミュレーションを100回ずつ行い、ある最大待ち人数が何回現れたかを表示する**ヒストグラム**



※試作問題から引用

試作問題 第2問

B モデル化とシミュレーション④

問題①

1. 複数の表の情報を総合して、値を求める

- ① **度数分布表**の**累積相対度数**の値と、「乱数から導き出した到着間隔」の表の値から、到着間隔の数字を計算する
- ② 「乱数から導き出した到着間隔」の表の内容を、「シミュレーション結果（作成途中）」に反映させて、「最大待ち人数」と「最長の待ち時間」を求める

2. 来客人数ごと（10人、20人、30人、40人）に、100回ずつシミュレーションをして、100回それぞれの「最大待ち人数」を調べ、100回のうち何回その待ち人数が発生したかを**ヒストグラム**にしたものを見て、そこから「読み取れないこと」を選択肢から選ぶ

試作問題 第2問

B モデル化とシミュレーション⑤

問題②

3. 条件が変わった時に、シミュレーションはどのような変化をするか、予測する

- 対応時間を4分から3分に短縮できた場合、「来客人数40人」の条件のシミュレーションがどのように変化するかを考え、グラフの形状を予測する

試作問題 第2問

B モデル化とシミュレーション⑥

解答するためのポイント

- 1つ1つの資料(表、グラフ)について、表の列の意味、グラフの縦軸・横軸の意味を、問題文と資料から読み取る
- 読み取った情報を元に表や図を完成させる
- シミュレーション結果から、仮説を読み取る

学習のポイント

- 一般的な表(度数分布表など)について、階級値や度数、相対度数、累積相対度数などの意味を把握しておく
- 与えられた値を、表やグラフに表現する
- 複数(2つ)の資料から情報を読み取り、数値や文章で表現する

「待ち行列」の問題だが、
待ち行列理論が分からないと解けない、という問題ではない

共通テスト 試作問題 第2問 まとめ

- 問題文に多くの資料や情報が埋め込まれている
- **表の列、グラフの縦軸・横軸**から、その資料が何を表しているか把握して、当てはまる値を考える
- **度数分布表**や**ヒストグラム**のような、基本的なデータの表現方法を踏まえて、その問題を解くために考えられた表やグラフの意図をつかみ、実際にデータを読み取る
- 試験のその場で、与えられた資料から情報を読み取り、回答を導き出すことは不可能ではないが、「**累積相対度数**」のような基本的な概念は知っておく必要がある

共通テスト 試作問題 第3問



試作問題 第3問

プログラムによる問題解決①

問題文に含まれる資料

- (その場に限っての)「**上手な払い方**」を定義する会話

個別・具体的な例

- 最初の例(日常でも目にする例):

460円の商品を買うのに、510円を払って、釣り銭を50円受け取る(釣り銭が、50円玉1枚で済む)

- 定義を整理して、「**一般化された定義**」にする会話

一般化

「客が支払う枚数と、釣り銭を受け取る枚数の合計を最小にする」

- 条件の追加:

硬貨の種類(1円玉、5円玉、10円玉、50円玉、100円玉。※500円玉は無し)

- 例示:46円の商品を買う時

- ちょうど支払う場合: $10円 \times 4 + 5円 \times 1 + 1円 \times 1$ 枚数は6枚

- 51円支払う場合:

- 支払い: $50円 \times 1 + 1円 \times 1$ 枚数は2枚

- お釣り: $5円 \times 1$ 枚数は1枚 合計3枚

具体例で説明

- 名前を付ける:**最小交換硬貨枚数**

短い名前を付ける

試作問題 第3問

プログラムによる問題解決②

問題

1. 一般化した解決法を文章で説明しながら、擬似プログラム言語による表現で穴埋めをし、プログラムの仕様(※)を確定する
 - 引数として渡した金額(例:46)について、最小になる枚数(例:6枚)を返す関数を利用する
2. 問1で仕様を確認した関数について、そのアルゴリズムを考える。文章およびDNCL言語によるプログラムの穴埋めをする 貪欲法
3. 2.でアルゴリズムと実装を考えた関数を利用して、「最小硬貨交換枚数」の計算を実現するアルゴリズムを考える。文章およびDNCL言語によるプログラムの穴埋めをする

※仕様…何を入力として受け取り、何を出力するか決めたもの。

アルゴリズム(プログラム)は、仕様を満たすように考え、作らなければならない

試作問題 第3問

プログラムによる問題解決③

問題を分割する

「最小交換硬貨枚数」
問題

問題を
分割しながら
解法を考える

「最小交換硬貨枚数」問題

関数「枚数」

金額を渡すと、その金額を
表せる最小枚数を返す

釣り銭が0円から99円の範囲で
繰り返し調べる

- 支払額(=価格+釣り銭)の最小枚数を求める(※関数「枚数」を使う)
- 釣り銭の最小枚数を求める(※関数「枚数」を使う)
- 合計の枚数を求める
- 「合計の枚数」が、今まで調べた範囲で最小なら、合計の枚数を記録する

試作問題 第3問

プログラムによる問題解決④

解答するためのポイント

- 「関数の入力と出力の仕様(何を入力すると、何が返されるか)を把握して、利用する」必要がある
- 問題の複雑さに圧倒されないようにする
- 問題を分割し、「問題を解決する上で、必要な機能は何か」に目を向けて、段階的にプログラムを作っていく
- 問1で求めるのは、
支払う額の最小枚数 +
釣り銭の最小枚数
$$\text{支払う額} = \text{価格}(x) + \text{釣り銭}(y)$$
- 「46円の商品を買う」とき、「51円払って5円お釣りをもらう」という答えをいきなり得ようとしているのではない
- 「釣り銭の額が0円の場合、1円の場合、・・・、99円の場合」と調べる処理を繰り返して、最小枚数になる支払い額を見つける

試作問題 第3問

プログラムによる問題解決⑤

試作問題を解く上で必要なプログラミングの技術的な知識は、教科書で取り上げられている範囲内

- **制御構造**

- 順次構造
- 条件分岐構造
- 反復（繰り返し）構造

- **データの取り扱い**

- 変数
- 配列

- **演算**

- 算術演算
- 比較演算

- **その他**

- 関数

+アルゴリズム

- 繰り返しと条件分岐を組み合わせる
- 繰り返しを使って、配列の全ての要素を扱う
- 繰り返し実行される処理は関数にする



整列（ソート）や、
探索（サーチ）と
同じレベル

試作問題 第3問

プログラムによる問題解決⑤

学習のポイント

- 何らかの**問題を解決するために**プログラミングの個々の知識・技能を習得して、利用する
- プログラミングの詳細な知識や、高度な技能を習得することは重要ではない
- 易しい問題からやや難しい問題まで、
 - ①問題を把握し、
 - ②問題を小さい問題に分割して
 - ③小さい問題を解決するプログラムを設計・作成・実行・検証することを積み重ねて、解決することに取り組む

共通テスト 試作問題 第3問 まとめ

- 「プログラミングで解決しようとしている課題」を、問題文で丁寧に説明している
 - よく知られた問題（整列、探索など）に対するアルゴリズム（バブルソート、線形探索など）をそのまま問う問題ではなく、その場で問題を与え、順を追ってアルゴリズムを考えていく
 - 問題とアルゴリズムを機械的に暗記する必要は無い
- プログラミング言語（DNCL言語）の基本的な文法を利用して、問題を解決しようとしている
 - 加算（+）や減算（-）、除算（÷）、剰余算（%）
 - 配列、添え字
 - 繰り返し、条件分岐
 - 関数（関数の引数、戻り値）
 - プログラミング言語（DNCL言語）の詳細な文法事項を問う問題ではない
- **様々な難易度の問題をプログラムで解決する実践が有効と考えられる**

共通テスト 試作問題 第4問



試作問題 第4問

データの分析①

問題文に含まれる資料①

生活時間の実態に関する統計調査

- 15歳以上19歳以下の若年層
- 都道府県別
- 平日1日の中で各生活行動に費やした時間(分)の**平均値**
- スマートフォン・パソコンなどの使用時間をもとに**グループ化**
 - 表1-A:スマートフォン・パソコンなどの使用時間が1時間未満のグループ⇨「使用時間が短いグループ」
 - 表1-B:スマートフォン・パソコンなどの使用時間が3時間以上6時間未満のグループ⇨「使用時間が長いグループ」
- それぞれの表の行:都道府県
- それぞれの表の列:睡眠(分)、身の回りの用事(分)、食事(分)、通学(分)、学業(分)、趣味・娯楽(分)

都道府県	睡眠(分)	身の回りの用事(分)	食事(分)	通学(分)	学業(分)	趣味・娯楽(分)
北海道	439	74	79	60	465	8
青森県	411	74	73	98	480	13
茨城県	407	61	80	79	552	11
栃木県	433	76	113	50	445	57

※試作問題から引用

試作問題 第4問

データの分析②

問題文に含まれる資料②

- データを扱う際の条件
 - **欠損値**の取扱い ⇨ 欠損値のある都道府県は除外する
 - **外れ値**の取扱い ⇨ 外れ値も含めて考える

試作問題 第4問

データの分析③ 問1①

問題

1. 与えられたデータを使って、どのような**仮説**を分析できるか/できないかを選ぶ

解答するためのポイント

- 与えられた**データ**が何を表しているか/**表していないか**を判断する
 - 勝手な推測をしない
 - 問題文・資料の中に無い情報や知識を挿入しない
- 「生活行動時間(分)」の情報はあるが、その時間が朝なのか・夜なのかの情報は含まれていない

試作問題 第4問

データの分析④ 問1②

学習のポイント

- 表の行・列、グラフの種類・軸などから、「何を表そうとしているか」を正しく見出す
- 与えられた表やグラフから、何が言えるか/何が言えないかについて、文章にして表現する
- 「ある情報・知識を得たい場合、どのようなデータを集める必要があるか」を検討する

試作問題 第4問

データの分析⑤ 問2①

問題文に含まれる資料

- 箱ひげ図

- 睡眠の時間
 - グループ別
- 学業の時間
 - グループ別
- 外れ値は「○」で表記

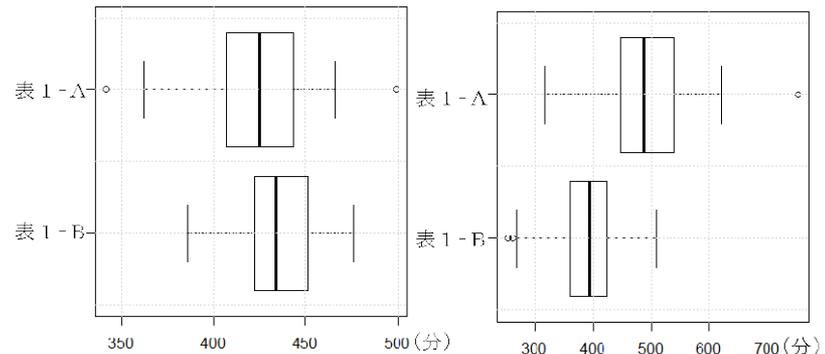


図1 睡眠の時間の分布

図2 学業の時間の分布

※試作問題から引用

試作問題 第4問

データの分析⑥ 問2②

問題

- 箱ひげ図から読み取れる情報を選ぶ

解答するためのポイント

- **データ別・グループ別**に作成された4つの箱ひげ図に示されている情報を正しく読み取る

学習のポイント

- 表の行・列、グラフの軸・種類などから、「何を表そうとしているか」を正しく見出す
- 与えられた表中の値やグラフの形状から導き出せる情報を、文章にして表現する

試作問題 第4問

データの分析⑦ 問3

問題文に含まれる資料

- 箱ひげ図
 - 表I-Aの値から表I-Bの値を引いた差を使って図示

問題

- 箱ひげ図から読み取れる情報を選ぶ

解答するためのポイント

- データを複合して作成された箱ひげ図に示されている情報を正しく読み取る

学習のポイント

- 表の行・列、グラフの軸・種類などから、「何を表そうとしているか」を正しく見出す
- 与えられた表中の値やグラフの形状から導き出せる情報を、文章にして表現する

試作問題 第4問

データの分析⑧ 問4

問題文に含まれる資料

- 表1-Aのデータについて、**散布図**に箱ひげ図を組み合わせた図

問題

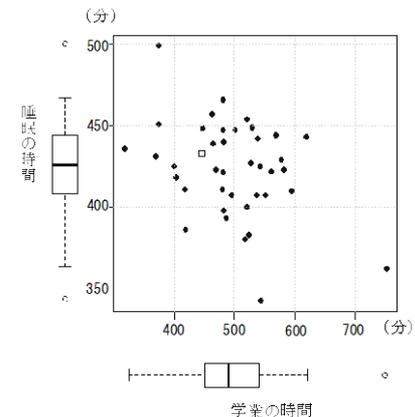
- 2つのデータの間にある「弱い負の相関」という評価について、**散布図・箱ひげ図**から読み取れる情報を選ぶ

解答するためのポイント

- 「**負の相関**」の意味を理解している
- 散布図に示されている情報を正しく読み取る

学習のポイント

- 2つの値の間の「**相関関係**」について意味を説明できるようにする
- 相関係数を求める式を暗記しておく必要はない



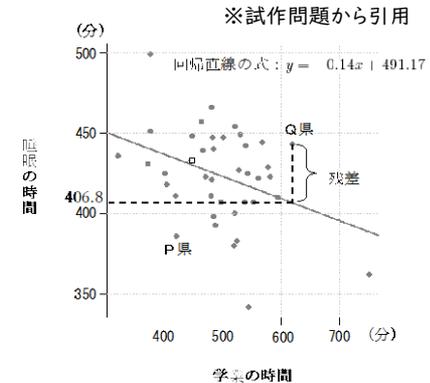
※試作問題から引用

試作問題 第4問

データの分析⑨ 問5①

問題文に含まれる資料

- 散布図に**回帰直線**を書き加えた図
- その図から読み取った情報についての文章
- 「**残差**」の概念の説明
- 回帰直線の式から求めた**推定値**と、残差を**標準化**して作った値で作った散布図



問題

- 回帰直線から求めた推定値と、残差を標準化して作った値を使い、外れ値となるデータを見出す

試作問題 第4問

データの分析⑩ 問5

解答するためのポイント

- **回帰直線**の意味が分かる
- データの**標準化**の意味が分かる
- 問題文の説明から「**残差**」の意味が分かる

学習のポイント

- **散布図**に示したデータから、**回帰直線**を引いてみる
- データを**標準化**してみる
- 推定値と実際の値を比較して、推定の精度を調べる
- 表計算ソフトを使うと、簡単に試すことができる

共通テスト 試作問題 第4問 まとめ

- 表の行・列の意味を読み取る
- 欠損値、外れ値などの基本的な概念の理解は必須
- グラフの種類ごとの表現の意味を読み取り、どのような仮説や推測が可能か（可能でないか）、文章で表現する
- 示された表やグラフから、データの特徴を読み取り、文章で表現する
- データを、複数のグループに分けて、表形式で表示する。そのグループ別のデータを、同じグラフの形式で表現して、グラフの形状の違いからデータの特徴を読み取る
- 二つの値を持つデータについて、散布図や相関（正の相関・負の相関）・相関係数・回帰直線などの概念を使って分析する

試作問題 第2問B、第4問

2系列のデータ(身長・体重)

1	身長	体重	2	身長	体重
A	167.5	67.4	a	170.3	58.2
B	170.4	68.0	b	166.9	72.0
C	170.0	58.8	c	177.8	64.4
D	171.5	54.6	d	186.0	69.3
E	172.0	76.9	e	176.9	61.5
F	175.7	93.4	f	191.2	69.3
G	172.6	70.7	g	180.5	97.0
H	168.2	54.4	h	174.5	54.7
I	173.4	64.5	i	170.6	59.2
J	175.0	84.9	j	199.0	108.2
K	168.0	48.5	k	194.0	87.4
L	164.9	65.2	l	164.2	46.8
M	161.2	54.7	m	166.6	82.2
N	180.2	50.9	n	140.7	57.2
O	174.0	59.4	o	175.9	63.5
P	169.0	61.4	p	186.1	75.8
Q	170.6	53.1	q	173.9	79.4
R	175.8	71.7	r	170.3	69.1
S	171.9	68.5	s	158.1	53.1
T	172.0	65.1	t	199.2	89.1

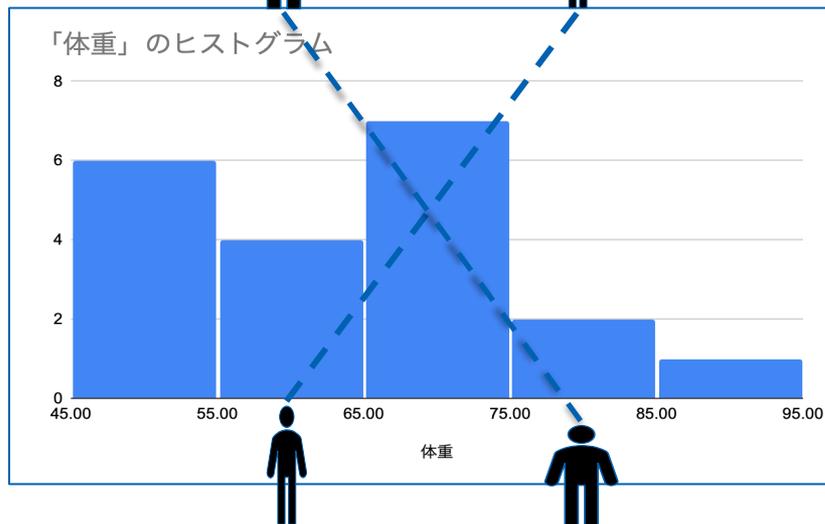
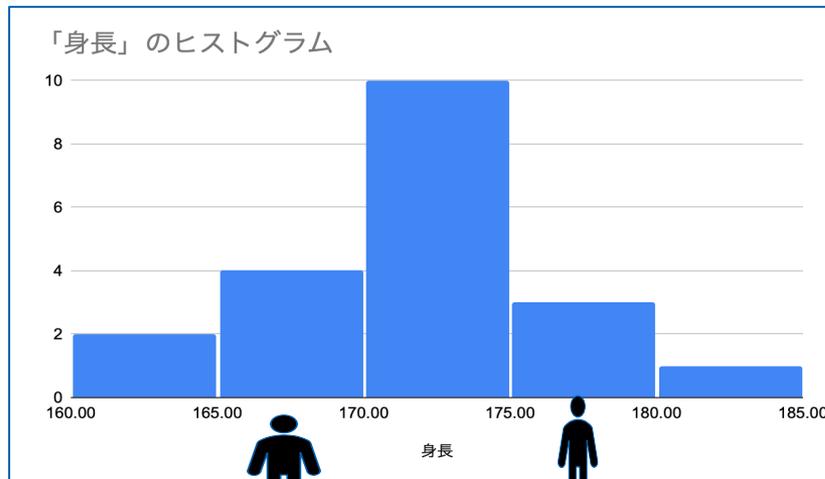
20人のグループが二つある

このグループのデータから、身長と体重には、どんな関係があると言えるだろうか？

「背が高いと、体重が重くなるだろう」という常識的な見通しは、データの分析によってどの程度ははっきりと言えるだろうか？

試作問題 第2問B、第4問

どう分析するか:ヒストグラム



身長と体重のヒストグラムを並べると

- 身長がどのように分布しているか
- 体重がどのように分布しているか

それぞれの傾向は分かる

しかし、

- 背の高い人の体重はどうか
 - 背の低い人の体重はどうか
- は分からない

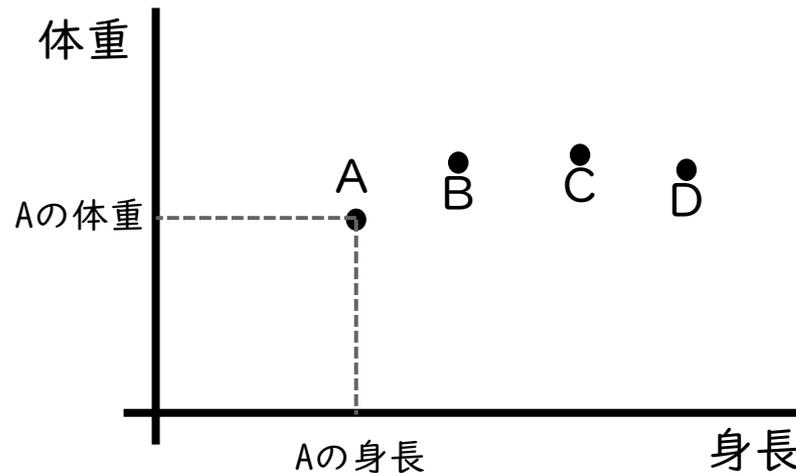
二つのデータの系列をまとめて見る方法が必要

試作問題 第2問B、第4問

散布図

散布図

一つのデータ(人)の、二つの値(身長・体重)を、二次元の座標の点として表現する



20個データがあるなら、20個の点を描く

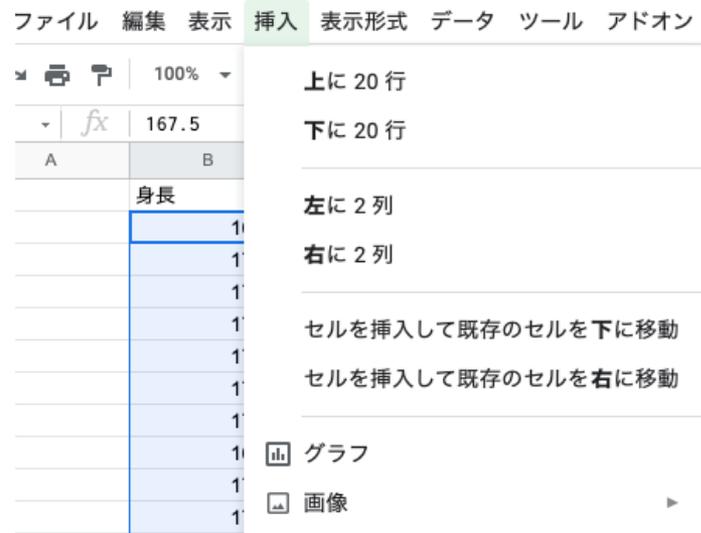
表計算ソフトを用いて散布図を描くことができる

試作問題 第2問B、第4問 表計算ソフトで散布図を描く①

①ドラッグして、データの範囲を選ぶ。

	A	B	C
1		身長	体重
2	A	167.5	67.4
3	B	170.4	68.0
4	C	170.0	58.8
5	D	171.5	54.6
6	E	172.0	76.9
7	F	175.7	93.4
8	G	172.6	70.7
9	H	168.2	54.4
10	I	173.4	64.5
11	J	175.0	84.9
12	K	168.0	48.5
13	L	164.9	65.2
14	M	161.2	54.7
15	N	180.2	50.9
16	O	174.0	59.4
17	P	169.0	61.4
18	Q	170.6	53.1
19	R	175.8	71.7
20	S	171.9	68.5
21	T	172.0	65.1

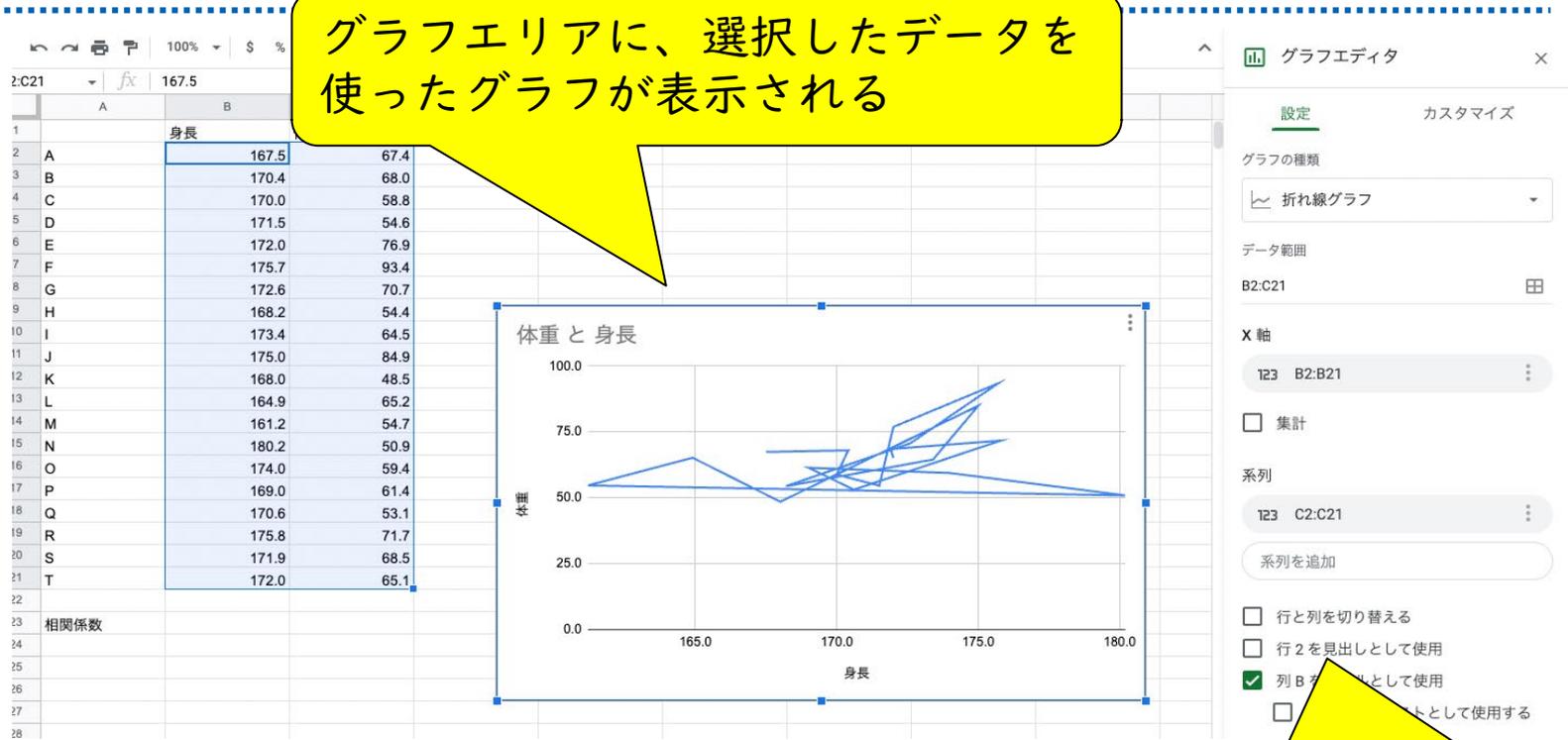
②表計算ソフトの挿入メニューを選び、表示される中からグラフを選ぶ。グラフエリアとグラフエディタが表示される



グラフエリアとグラフエディタの
説明は次のページで

試作問題 第2問B、第4問 表計算ソフトで散布図を描く②

グラフエリアに、選択したデータを使ったグラフが表示される

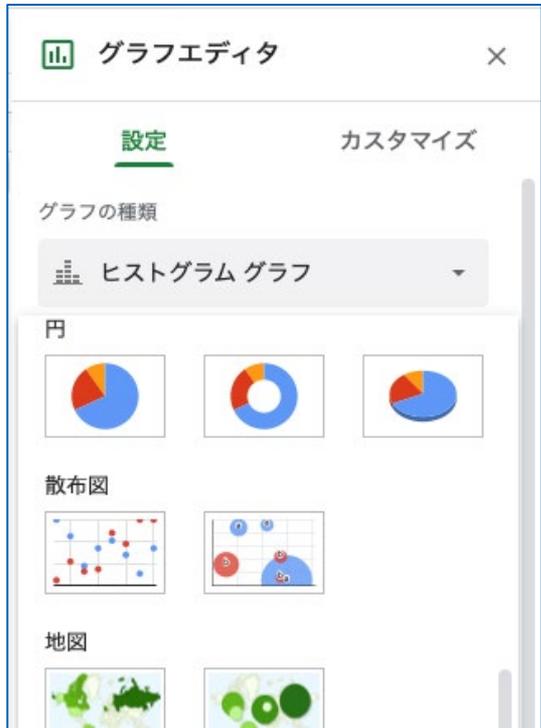


✓ グラフエディタを再表示するには



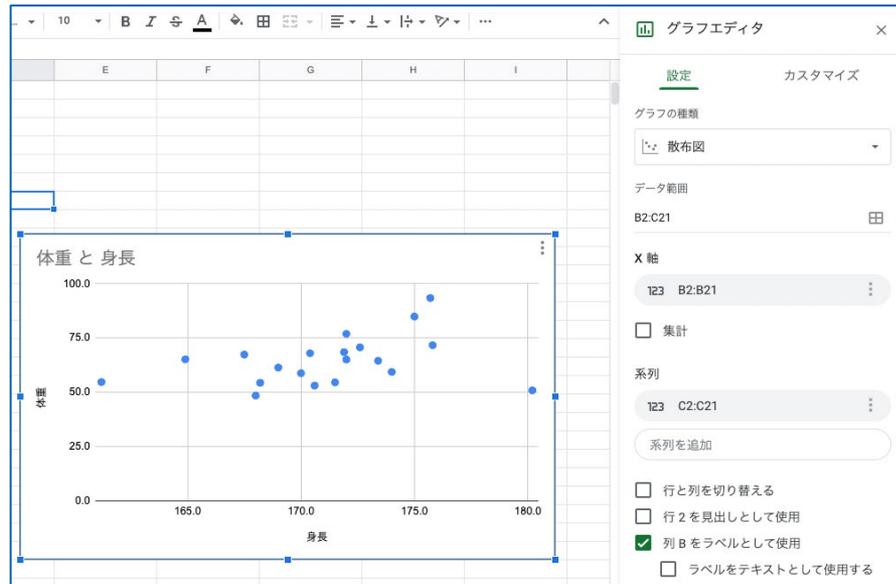
グラフエディタで、グラフの設定を
変更することが出来る

試作問題 第2問B、第4問 表計算ソフトで散布図を描く③



③ グラフエディタの「設定」->
「グラフの種類」とたどり、
「散布図」を選ぶ。

④ グラフエリアに散布図が表示される。



※ グラフエディタの「カスタマイズ」->「グラフと軸のタイトル」を
書き換えると、 グラフの上のタイトルを変更できる

試作問題 第2問B、第4問

データを理解するために、表やグラフを作ってみる

- **ひと目では、その特徴をつかむことができない量のデータを与える**
- 1つのアイテムに、複数の値(系列)がある
 - 例:生徒の身長と体重
- 1群のデータには、10件以上のアイテムが含まれる
 - 例:1クラスには10人以上の生徒がいる
- 2群以上のデータがあり、比較できるようになっている
 - 例:2クラス分の身長・体重データがある
- **表やグラフ、統計量を求める**
- **表計算ソフトを利用する**
- 手作業や、手計算にこだわる必要はない
- 統計量や、統計的な概念を理解するために、計算式を学び、手計算することには固有の意義がある

(参考)

Monaca Educationスタンダードプラン

サポートページ

<https://edu.monaca.io/standard>

※有償プランご契約の学校様に、パスワードをお知らせしています

4章 情報通信ネットワークとデータの活用

- 教材・動画を公開しています

4章：情報通信ネットワークとデータの活用

本章では標準時数11時間分の教材を用意しております。一部の単元はAPSを別途入手して下さい。

Youtubeで関連動画を見る

単元名称	概要	標準時数	教材
情報通信ネットワークの仕組み	自身でLANを設計できることを目標に学習。	2	ダウンロード
情報セキュリティ	外出先などでも安全にネットを利用するために必要な知識の習得を目標に学習。	2	ダウンロード
情報システムとサービス	オープンデータや二次利用可能なライセンス、また、GISに言及しています。	1	ダウンロード
データベース	APSのRDB蔵書管理をご利用下さい。	3	ダウンロード
基本統計量（表計算ソフトの関数）	表計算ソフトの関数を用いて基本統計量を学習します。	1	ダウンロード
散布図と回帰直線・相関係数	データの傾向から予測します。	2	ダウンロード

公開している動画

(参考)

Monaca Educationスタンダードプラン 動画『4章： 情報通信ネットワークとデータの活用』

非公開のYouTube再生リスト

1本あたり5分から30分程度の動画

1. 情報通信ネットワークの仕組み
2. 情報セキュリティ
3. 情報システムとサービス
4. 基本統計量①～②
5. 散布図・相関係数・回帰直線①～④
6. 情報システムとデータベース①～③
7. 情報システムとデータベース応用①～⑤



第2部

	時間	内容
単元「プログラミング」の実習の進め方	14:10 - 15:00	試作問題 分析のまとめ Monaca Educationを用いた、実習によるプログラミングの学習方法の紹介 <ul style="list-style-type: none">• Monaca Educationの概要• ログイン• 基本の使い方と、授業への導入（ブロック崩しプロジェクトによる例） 試作問題 第3問 プログラミングの分析 <ul style="list-style-type: none">• サンプルプロジェクトのインポート
振り返りとまとめ	15:00 - 15:10	<ul style="list-style-type: none">• 挨拶• アンケートの記入• 事務連絡 など

大学入学共通テスト「情報」 対策セミナー 第2部

Monaca Educationを用いた
実習によるプログラミングの
学習方法の紹介

アシアル株式会社
アシアル情報教育研究所



共通テスト 試作問題の分析



共通テスト 試作問題の分析

知識を問う問題は、基礎的なもの

- 細かく・最新の・マニアックな知識ではなく、基礎的な内容

基礎的な知識に、簡単なルールを追加した上で、それを適用して、結果や効果を推測させる問題がある

- 必要な追加の情報は問題文によって示されるので、(基礎的な知識に追加情報を加えて)論理的に考えを進めれば、解答できる
 - ※ITの知識がある人が、「あまり勉強しないでも、その場で解ける」と言うことがあるのは、これが理由
- 広い範囲の知識、応用的な知識を知っていれば、より簡単に短時間で解けるかもしれない
- ただし、抽象度の高い知識・情報が多く、座学だけで応用的な知識を積み上げるのは難しい
- その場で資料を適切に読む訓練を積む方が適切である

共通テストを意識した 情報科の授業の方向性の提案



共通テストを意識した 情報科の授業の方向性の提案①

- 教科書を「**満遍なく**」学習する必要はない
 - (おそらく、普通の授業時間の範囲で満遍なく学習することは不可能)
 - 細かな、応用的な範囲を無理に追求して、基礎的な知識の習得と活用の技能がおろそかになると、かえって共通テストが解けない可能性がある
- **体験的に学ぶ実習授業が必要である**
 - 基礎的な**知識**に、
 - (追加的に) **手順やルールを適用**して、**結果や効果を推測**し、
 - 実際に**結果や効果を確認**して、自身の推測が正しかったか**検証**する
- **土台になる知識・技能が体験的に身に付いた後であれば、詳細・応用はドリル教材などで補完できる**

共通テストを意識した 情報科の授業の方向性の提案②

実習しながら、基礎的な概念を体験的に学習する

細かな知識を、基礎的な問題集やドリル型の教材で習得・確認する

共通テストの特徴的な問題形式に慣れる。長文を読み解いて、解答に辿り着く練習をする

休憩

14:00 - 14:10

共通テスト 試作問題 第3問



試作問題 第3問

プログラムによる問題解決①

問題文に含まれる資料

- その場に限っての「上手な払い方」を定義する会話
 - 最初の例(日常でも目にする例):
460円の商品を買うのに、510円を払って、釣り銭を50円受け取る(釣り銭が、50円玉1枚で済む)
- 拡張して、再定義:
「客が支払う枚数と、釣り銭を受け取る枚数の合計を最小にする」
 - 条件の追加:
硬貨の種類(1円玉、5円玉、10円玉、50円玉、100円玉。※500円玉は無し)
 - 例示:46円の商品を買う時
 - ちょうど支払う場合: $10円 \times 4 + 5円 \times 1 + 1円 \times 1$ 枚数は6枚
 - 51円支払う場合:
 - 支払い: $50円 \times 1 + 1円 \times 1$ 枚数は2枚
 - お釣り: $5円 \times 1$ 枚数は1枚 合計3枚
 - 名前を付ける:**最小交換硬貨枚数**

試作問題 第3問

プログラムによる問題解決②

問題

1. 一般化した解決法を文章で説明しながら、擬似プログラム言語による表現の穴埋めをする
 - 引数として渡した金額(例:46)について、最小になる枚数(例:6枚)を返す関数を利用する
2. 問1で仕様(※入力に対して何を出力するか)を確認した関数について、そのアルゴリズムを考える。文章およびDNCL言語によるプログラムの穴埋めをする
3. 2.でアルゴリズムと実装を考えた関数を利用して、「最小硬貨交換枚数」の計算を実現するアルゴリズムを考える。文章およびDNCL言語によるプログラムの穴埋めをする

試作問題 第3問

プログラムによる問題解決③

問題を分割する

「最小交換硬貨枚数」
問題

問題を
分割しながら
解法を考える

「最小交換硬貨枚数」問題

関数「枚数」

金額を渡すと、その金額を
表せる最小枚数を返す

釣り銭が0円から99円の範囲で
繰り返し調べる

- 支払額 (= 価格 + 釣り銭) の最小枚数を求める (※関数「枚数」を使う)
- 釣り銭の最小枚数を求める (※関数「枚数」を使う)
- 合計の枚数を求める
- 「合計の枚数」が、今まで調べた範囲で最小なら、合計の枚数を記録する

試作問題 第3問

プログラムによる問題解決④

解答するためのポイント

- 「関数の入力と出力の仕様(何を入力すると、何が返されるか)を把握して、利用する」必要がある
- 問題の複雑さに圧倒されないようにする
- 問題を分割し、「問題を解決する上で、必要な機能は何か」に目を向けて、段階的にプログラムを作っていく
- 問1で求めるのは、
支払う額の最小枚数 +
釣り銭の最小枚数
$$\text{支払う額} = \text{価格}(x) + \text{釣り銭}(y)$$
- 「46円の商品を買う」とき、「51円払って5円お釣りをもらう」という答えをいきなり得ようとしているのではない
- 「釣り銭の額が0円の場合、1円の場合、・・・、99円の場合」と調べる処理を繰り返して、最小枚数になる支払い額を見つける

試作問題 第3問

プログラムによる問題解決⑤

試作問題を解く上で必要なプログラミングの技術的な知識は、教科書で取り上げられている範囲内

- **制御構造**

- 順次構造
- 条件分岐構造
- 反復（繰り返し）構造

- **データの取り扱い**

- 変数
- 配列

- **演算**

- 算術演算
- 比較演算

- **その他**

- 関数

+アルゴリズム

- 繰り返しと条件分岐を組み合わせる
- 繰り返しを使って、配列の全ての要素を扱う
- 繰り返し実行される処理は関数にする



整列（ソート）や、
探索（サーチ）と
同じレベル

試作問題 第3問

プログラムによる問題解決⑤

学習のポイント

- 何らかの**問題を解決するために**プログラミングの個々の知識・技能を習得して、利用する
- プログラミングの詳細な知識や、高度な技能を習得することは重要ではない
- 易しい問題からやや難しい問題まで、
 - ①問題を把握し、
 - ②問題を小さい問題に分割して
 - ③小さい問題を解決するプログラムを設計・作成・実行・検証することを積み重ねて、解決することに取り組む

共通テスト 試作問題 第3問 まとめ

- 「プログラミングで解決しようとしている課題」を、問題文で丁寧に説明している
 - よく知られた問題（整列、探索など）に対するアルゴリズム（バブルソート、線形探索など）をそのまま問う問題ではなく、その場で問題を与え、順を追ってアルゴリズムを考えていく
- プログラミング言語（DNCL言語）の基本的な文法を利用して、問題を解決しようとしている
 - 加算（+）や減算（-）、除算（÷）、剰余算（%）
 - 配列、添え字
 - 繰り返し
 - 関数（関数の引数、戻り値）
 - プログラミング言語（DNCL言語）の詳細な文法事項を問う問題ではない
- 様々な難易度の問題をプログラムで解決する実践が有効と考えられる

情報Ⅰ プログラミング領域 実習授業



プログラミングを学習する

「プログラミングに慣れる」

- プログラミングに慣れる
- プログラミングをするための**開発環境・ツール**に慣れる

プログラミングを学習する

- プログラム言語の文法を学習する
- プログラム言語を用いてプログラムを作成して、動作させる
- 動作する**プログラムの内容を観察**して、一部**修正**して、**結果を確認**する

この時間の概要

Monaca Educationを試用する

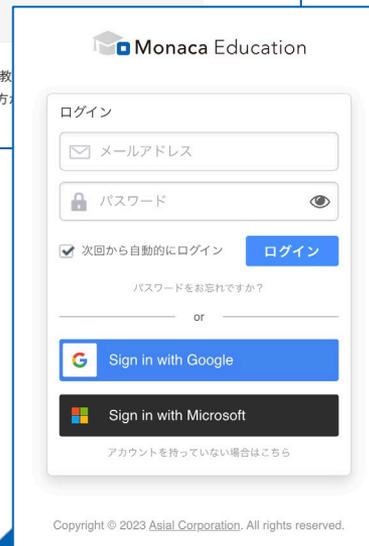
- プログラミングの授業の導入部
- プログラムの開発環境・ツールに慣れる

試作問題 第3問のプログラム作例を動かしてみる

- 動作するプロジェクトをWebサイトからインポートする
 - Monaca Educationの機能を利用する
- 作成済みのプログラムを一部編集して、機能を拡張する

Monaca Education

<https://edu.monaca.io>



- クラウド上で動作するプログラムの開発環境
 - 教材も多数用意
- Webブラウザがあれば利用可能
 - ソフトウェアの追加インストールの必要無し
- 生徒1人に1つのアカウントを用意して使用
 - 先生には管理アカウントを提供。パスワードの再発行も簡単
 - Googleアカウント、MSアカウントを使ったシングルサインオン (SSO) も可能

I. Monaca Education Webサイトにアクセスする <https://edu.monaca.io>



アプリ開発によるプログラミング教育

ログイン アカウント作成

Monaca Education

Monacaで学ぶ理由 教材 教員研修 料金表 FAQ 学習者向け お知らせ

先生の想いと生徒の未来に寄り添う
プログラミング教育

お問い合わせ

Monaca Educationとは

Monaca Educationは、教育機関における情報教育を通じて中高生の未来を拓くプログラミング教育サービスです。生徒たちはスマホアプリを作りながらプログラミングの基礎学習から作品作りまで取り組みます。現場の先生方からは、教材の内容に加えて、研修や技術サポート、補助教材が豊富で安心して授業に取り組みるとご評価いただいています。

2. ログインする

Monaca Educationサイトの右上「ログイン」ボタンからログインページへ

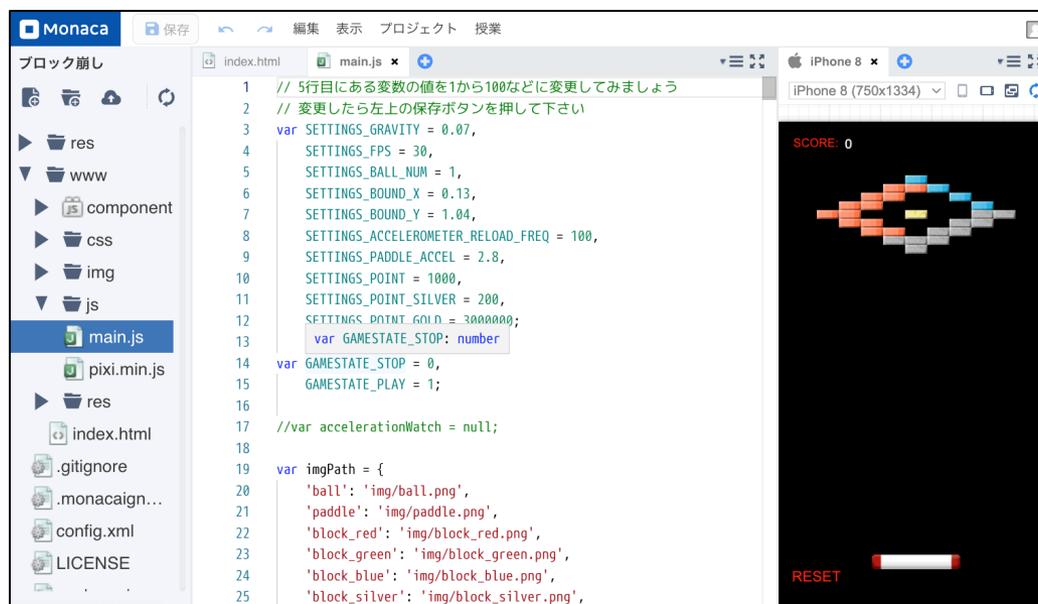
A screenshot of the Monaca Education login page. The page features the Monaca Education logo at the top. Below the logo, there is a 'ログイン' (Login) section with a 'メールアドレス' (Email address) input field, a 'パスワード' (Password) input field with an eye icon for visibility, and a '次回から自動的にログイン' (Remember me) checkbox. A blue 'ログイン' button is positioned to the right of the password field. Below the password field, there is a link 'パスワードをお忘れですか?' (Forgot your password?). A horizontal line with 'or' in the center separates the password field from the social login options. There are two social login buttons: 'Sign in with Google' (blue) and 'Sign in with Microsoft' (black). At the bottom, there is a link 'アカウントを持っていない場合はこちら' (Click here if you don't have an account).

- 生徒用のアカウント、パスワードは事前に配布しておくスムーズ
- 手入力する場合(※コピー&ペーストできない場合)は、紛らわしい字に注意する(数字の1と小文字l、大文字Iなど)

Monaca Educationの プロジェクト「ブロック崩し」

Monaca Educationにログイン⇨

「新しいプロジェクトを作る」⇨「ブロック崩し」で作成



- すぐに動作する
- ソースコードが付いているので、変更して試しながら学べる

3. ブロック崩しプロジェクトを作成する

①

「新しいプロジェクトを作る」



3. ブロック崩しプロジェクトを作成する

②

テンプレートを選び、

プロジェクトの詳細を入力する



- プロジェクト名 (氏名)
- 詳細 (今日の日付)

3. ブロック崩しプロジェクトを作成する

③

プロジェクトをクリックで選択した後、
「クラウドIDEで開く」をクリックする



4.プログラムの動作を試す

右端のプレビューパネルでブロック崩しができる

- 画面の要素や機能を紹介する

The screenshot shows the Monaca IDE interface. On the left is the 'プロジェクトパネル' (Project Panel) with a file tree. The center is the 'エディタ' (Editor) showing JavaScript code for a game. On the right is the 'プレビューパネル' (Preview Panel) showing a mobile browser with a game running. Callouts point to specific features: the project panel, the editor, the refresh icon in the preview panel, and the preview panel itself.

更新アイコン
プログラムを
実行しなおす

プロジェクトパネル
ダブルクリックで、
ファイルを選ぶ

エディタ
プログラムを
編集する

プレビュー
パネル
プログラムを
実行する

(生徒向けの導入、進捗の調整) ブロックにボールを当てて、全部壊してみよう

- 制限時間3分

クリアできたら
挙手!

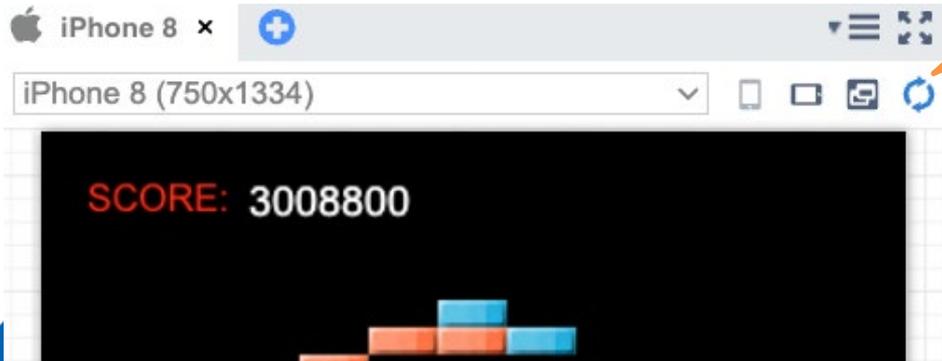
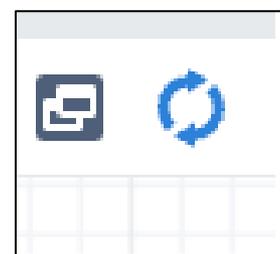


- 何回やり直してもよい

- プレビューパネル右上の矢印のアイコンをクリックすると、再チャレンジ

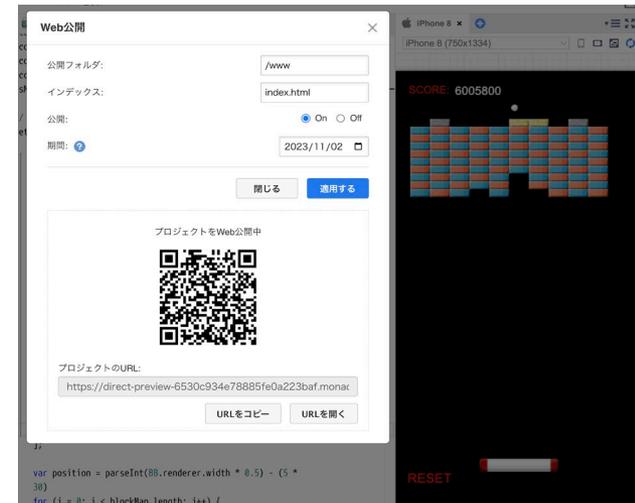
- 自分のベストの得点をメモしておこう

- 画面上の「SCORE」の横に得点が表示されている



6. Web公開機能を使って、先生やクラスメートに公開する

Web公開機能でURLを発行する



発行したURLにWebブラウザでアクセスすると、他の人からもカスタマイズしたアプリを利用できる



あんこエデュケーション ブログ

Monaca Educationで大学入学共通テストの試作問題に取り組む～釣り銭が少ない上手な払い方

<https://anko.education/archives/2979>

Monaca Educationにログイン済みのWebブラウザで別にタブを開き、上記URLにアクセスする。

または、あんこエデュケーション(<https://anko.education/>) トップページから「ブログ」をたどる



第3問 図1

(1) `Kouka = [1, 5, 10, 50, 100]`

(2) `kingaku = 46`

(3) `maisu = 0, nokori = kingaku`

(4) `i` を `キ` ながら繰り返す：

4から0まで1ずつ減らし

(5) `maisu = ケ + ケ`

`maisu = maisu + nokori ÷ Kouka[i]`

(6) `nokori = コ`

`nokori = nokori % Kouka[i]`

(7) 表示する(`maisu`)

図1 目標の金額ちょうどになる最小の硬貨枚数を計算するプログラム

※試作問題から引用



第3問 図1のプログラム

Python版

```
import math
```

```
Kouka = [1,5,10,50,100]
```

```
kingaku = 46
```

```
maisu = 0
```

```
nokori = kingaku
```

```
for i in range(4,-1,-1):
```

```
    maisu = maisu + math.floor( nokori/Kouka[i] )
```

```
    nokori = nokori % Kouka[i]
```

```
print(maisu)
```

(1) `Kouka = [1,5,10,50,100]`

(2) `kingaku = 46`

(3) `maisu = 0, nokori = kingaku`

(4) `i` を `キ` ながら繰り返す:

4から0まで1ずつ減らし

(5) `maisu = ク + ケ`

`maisu = maisu + nokori ÷ Kouka[i]`

(6) `nokori = コ`

`nokori = nokori % Kouka[i]`

(7) 表示する(`maisu`)

図1 目標の金額ちょうどになる最小の硬貨枚数を計算するプログラム

第3問 図1のプログラム

JavaScript版

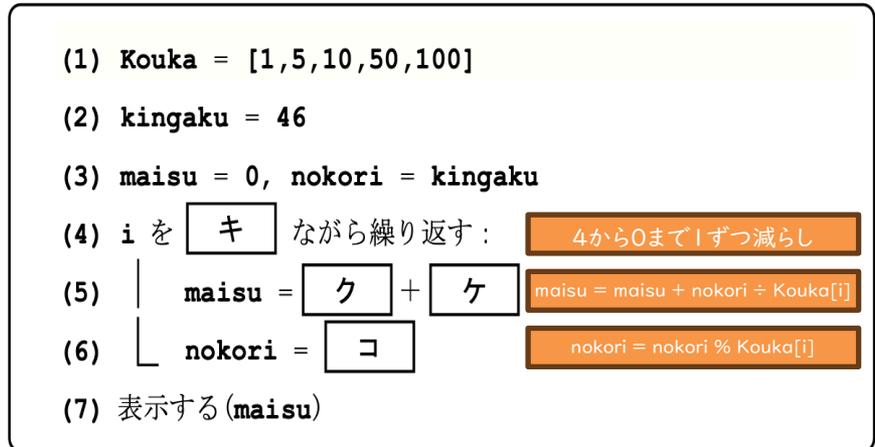


図1 目標の金額ちょうどになる最小の硬貨枚数を計算するプログラム

```
let Kouka = [1,5,10,50,100]
let kingaku = 46
let maisu = 0, nokori = kingaku
for(let i=4; i>=0; i-- ){
    maisu = maisu + Math.floor( nokori / Kouka[i] )
    nokori = nokori % Kouka[i]
}
document.write( maisu );
```

第3問 図2

(1) `kakaku = 46`

(2) `min_maisu = 100`

tsuri を 0 から 99 まで
1 ずつ増やしながら繰り返す

(3) `サ` を `シ` から 99 まで 1 ずつ増やしながら繰り返す：

(4) `shiharai = kakaku + tsuri`

(5) `maisu = ス + セ` `maisu = 枚数(shiharai) + 枚数(tsuri)`

(6) もし `ソ` < `min_maisu` ならば：

(7) `タ = ソ` `もし maisu < min_maisu ならば：
min_maisu = maisu`

(8) 表示する (`min_maisu`)

図2 最小交換硬貨枚数を求めるプログラム

※試作問題から引用

第3問 図2のプログラム

Python版

```
import math

def Maisu(kingaku):
    Kouka = [1,5,10,50,100]
    maisu = 0
    nokori = kingaku
    for i in range(4,-1,-1):
        maisu = maisu + math.floor( nokori/Kouka[i] )
        nokori = nokori % Kouka[i]
    return maisu

kakaku = 46
min_maisu = 100
for tsuri in range(0,100):
    shiharai = kakaku + tsuri
    maisu = Maisu(shiharai) + Maisu(tsuri)
    if maisu < min_maisu:
        min_maisu = maisu
print(min_maisu)
```

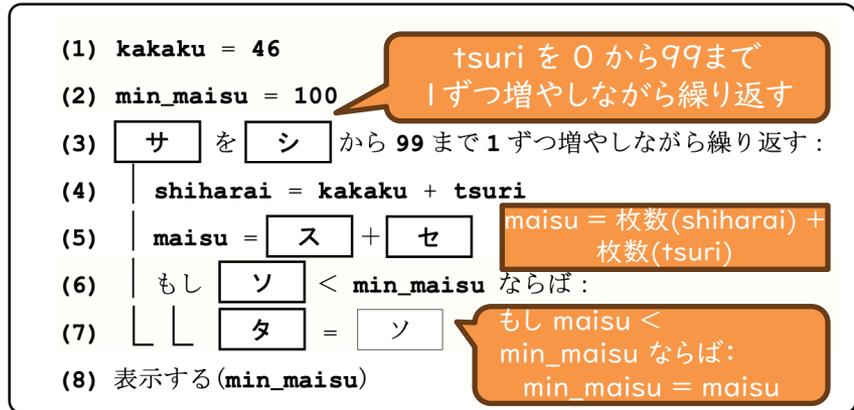


図2 最小交換硬貨枚数を求めるプログラム

第3問 図2のプログラム

JavaScript版

```
function Maisu(kingaku){
    let Kouka = [1,5,10,50,100]
    let maisu = 0, nokori = kingaku
    for(let i=4; i>=0; i-- ){
        maisu = maisu + Math.floor( nokori / Kouka[i] )
        nokori = nokori % Kouka[i]
    }
    return maisu;
}

let kakaku = 55
let min_maisu = 100
for(let tsuri=0; tsuri <= 99; tsuri++){
    let shiharai = kakaku + tsuri;
    let maisu = Maisu(shiharai) + Maisu(tsuri);
    if ( maisu < min_maisu ){
        min_maisu = maisu;
    }
}
document.write(min_maisu);
```



図2 最小交換硬貨枚数を求めるプログラム

振り返りとまとめ



プログラミングを学習する

「プログラミングに慣れる」

- **プログラミング**に慣れる
- プログラミングをするための**開発環境・ツール**に慣れる

プログラミングを学習する

- プログラム言語の**文法**を学習する
- プログラム言語を用いてプログラムを**作成して、動作させる**
- 動作する**プログラムの内容を観察**して、一部**修正**して、**結果を確認**する

本研修の目的と概要

2025年1月の共通テストから、『情報』が追加される

本研修の目的

- 共通テストを視野に入れた、情報科の授業のあり方を考える

本研修・資料の概要

- 共通テスト「情報」で出題される問題について、公表されている資料を元に分析する
- 共通テストを視野に入れる場合、どのような授業があり得るか検討し、実習を行う意義を確認する
- 授業で実習を行う場合に利用できる素材、ツールを実際に試してみる
 - Monaca Education
 - 論理回路シミュレータ
 - 表計算ソフト など

※本研修で扱わないこと

- そのまま利用できる授業案は提供しない

共通テストを視野に入れた 授業の方向性の提案

実習しながら、基礎的な概念を体験的に学習
する

細かな知識を、基礎的な問題集やドリル型の
教材で習得・確認する

共通テストの特徴的な問題形式に慣れる。長
文を読み解いて、解答に辿り着く練習をする

	時間	内容
共通テスト 試作問題の分析と 実習中心の授業のアイデア	13:00 - 14:00	共通テストの試作問題の分析 <ul style="list-style-type: none"> 試作問題のダウンロード方法の確認 各問題の概要と回答のポイント・学習のポイント・実習中心の授業のアイデア <ul style="list-style-type: none"> 第1問 情報社会の問題解決・コンピュータの仕組み・情報デザインなど 第2問 A コンピュータの仕組み、B モデル化とシミュレーション 第3問 プログラミング 第4問 データの活用
休憩	14:00 - 14:10	
単元「プログラミング」の実習 の進め方	14:10 - 15:00	共通テストの試作問題 第3問 プログラミングの分析 <ul style="list-style-type: none"> 概要と回答のポイント・学習のポイント <p>Monaca Educationを用いた、実習によるプログラミングの学習方法の紹介</p>
振り返りとまとめ	15:00 - 15:10	<ul style="list-style-type: none"> 挨拶 アンケートの記入 事務連絡 など

【参考】事前にいただいた質問とその答え

- 申込時にいただいた質問とその答えを記載しています。
- 質問については、原文通りではなく、一部編集しています。

質問とその答え①プログラミングの目標レベル

質問

高校の卒業時点で、プログラミングをどの程度まで身につけた状態が目標として想定されているのでしょうか。

回答

平成30年告示の学習指導要領

(https://www.mext.go.jp/content/1407073_11_1_2.pdf)のp.33に、例を挙げて説明されています。

「プログラミング言語の基本的な概念・機能を習得して、問題を解くための基本的なアルゴリズムを実装できる」と言い換えられると思います。

「プログラミング言語の基本的な概念・機能」の範囲は、教科書の記載から、次のように想定できます。

- 基本制御構造（順次、条件分岐、繰り返し）
- データ構造（変数、配列/リスト）
- その他（関数の定義と利用）

複雑・高度なプログラムを作成できるようになることは求められていません。

質問とその答え②プログラミングの実習授業の扱い

質問

プログラミングの実習の匙加減が
難題です。

広い範囲・高度なレベルを目指せば時間がかかるし、

狭い範囲・低いレベルでは十分な
学習になりません。

回答

プログラミングを使いこなすためには実習は必須と考えます。

ただし、実習をまじえながら、知識・技能をゼロから積み上げる方法（変数から初めて、順次実行、条件分岐へ…配列と繰り返しを組み合わせで…）では時間が足りなくなるのは確実です。

小さな完成版のプログラムの動作を見せた後で、少しずつ変更して動作が変わる様子を確認しながら、プログラミングに関する各概念を学ぶ方法が考えられます。2コマ程度から始めることができます。

【参考】

用語集：APS（アプリプログラミングシート）

<https://edu.monaca.io/glossary-top/aps>

APS（アプリプログラミングシート）一覧

<https://edu.monaca.io/aps-top>

質問とその答え③DNCLについて

質問

DNCL (共通テスト手順記述標準言語) の動向と教授法について教えてください。

回答

DNCLの今後について、大学入試センターから正式に発表された情報はありません。

非常に限定された文法だけ、共通テストの問題文の中で提示・説明されると考えられます。

DNCLについて、特別な教授法を考える必要はないと考えています。

JavaScriptやPythonなどの手続き型言語で、変数・配列・条件分岐・繰り返し・代入・演算(数値演算、比較演算、論理演算)・関数などの機能を標準的な方法で使える状態になっていれば、DNCLによる出題を、それらの言語に置き換えて考えるのは簡単だからです。

質問とその答え④プログラミング言語の選定

質問

授業で使うプログラミング言語はやはりPythonがいいでしょうか？

「DNCLはPythonに似ている」と聞いたことがあります。

回答

Python、JavaScript、Scratchなど、手続き型の言語であればどれでも構わないと思います。

学校や、生徒や学生の個人の環境(※)で、プログラミングおよびデバッグの環境を準備しやすい言語を選ぶと良いのではないのでしょうか。

※授業の後の、共通テスト向けの学習を想定しています。

質問とその答え⑤問題集

質問

参考問題集などありますか？

回答

「大学入試センターが公開したサンプル問題、試作問題と同等の水準のもの」ということで言いますと、2023年12月現在、十分な内容を備えているものは無いように思います。

共通テスト水準の問題集は、各出版社が制作中で、来年(2024年)1月末以降に発表されるものと見ています。

なお、既刊の各社の問題集は、共通テストの直接的な準備向けではないものの、基礎を身につけるのに役立つものが多いと思います。

質問とその答え⑥自学自習の方法

質問

生徒が授業外で自学自習を進めるのに適した方法が知りたいです。

回答

基本的な概念を把握し、知識を習得するための自学自習には、スタサプなどの学習サービス、アプリが役に立ちます。

プログラミングや、データ活用などの単元については、実習がよいと考えます。弊社サービスや、Webサイトのコンテンツをご利用ください。

実習を要する内容については、ゼロから生徒に自学自習させるのは難しいため、生徒に対して、動作するプログラムや、表計算ソフトを使った分析例を提示した上で、少しの変更・操作で結果が得られる課題を提示するのがよいと考えます。

質問とその答え⑦高2までに学習しておくべきこと

質問

共通テストを受ける場合、高2の3学期までに「情報」の受験に対してやっておくべき事項は何でしょうか。

回答

『情報I』の、4つの内容のまとめり((1)情報社会の問題解決、(2)コミュニケーションと情報デザイン、(3)コンピュータとプログラミング、(4)情報通信ネットワークとデータの活用)の全てについて、それぞれ**何らかの実践的・実習的な学習**をしておくことが望ましいと考えます。それぞれ2～4コマ程度でも意義があります。

抽象度が高く、知識・経験の少ない高校生が、座学だけで理解する・機械的に暗記するのは困難だと考えるからです。

概念を実習を通じて応用することは、共通テスト対策上も大切だと思います。

一方、高2の3学期までに教科書の内容を全て網羅するのは諦めざるを得ません。

質問とその答え⑧共通テストに向けた対策

質問

共通テストに向けた対策は生徒たちに何をやらせれば良いでしょうか。

回答

2024年1月末以降、出版社や教育サービス会社から、情報の共通テスト対策を謳う問題集が発刊されると思われます。それらに取り組みさせる方法が考えられます。

共通テスト対策をうたわない教材であっても、情報Ⅰの学習範囲に含まれる基礎知識の網羅的な習得に役立てられます。

その他、他科との連携を通じて、情報Ⅰの学習内容を応用する方向もあり得るかもしれません。

- 問題文に示された実社会での状況や課題を読み取る
 - 課題を整理して、解法を探る
 - (習得している)手法・技法を使って解答を得る
- これらの問題傾向は、他科目と共通しています。

【付録】本資料で紹介した アシアルのサービス・教材・ツールのまとめ



Monaca Education

<https://edu.monaca.io>

アプリ開発によるプログラミング教育

ログイン アカウント作成

Monaca Education

Monacaで学ぶ理由 教材 教員研修 料金表 FAQ 学習者向け お知らせ



先生の想いと生徒の未来に寄り添う
プログラミング教育

お問合わせ

Monaca Educationとは

Monaca Educationは、教育機関における情報教育を通じて中高生の未来を拓くプログラミング教育サービスです。生徒たちはスマホアプリを作りながらプログラミングの基礎学習から作品作りまで取り組めます。現場の先生方からは、教材の内容に加えて、研修や技術サポート、補助教材が豊富で安心して授業に取り組みめるとご評価いただいています。

Monaca Educationスタンダードプラン

サポートページ

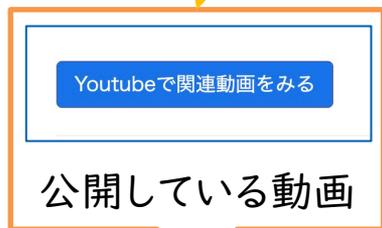
<https://edu.monaca.io/standard>

※有償プランご契約の学校様に、パスワードをお知らせしています

3章 コンピュータとプログラミング

4章 情報通信ネットワークとデータの活用

- 教材・動画を公開しています



3章：コンピュータとプログラミング

本プランは標準時数100時間分の教材を用意しております。一部の単元はAPSを別途入手して下さい。

[Youtubeで関連動画をみる](#)

単元名称	概要	標準時数	教材
コンピュータの仕組み (2進数)	論理回路シミュレータなどを利用して2進数を学びます。	1	ダウンロード
コンピュータの仕組み (16進数)	16進数で色の配色を行ったり2進数との相互変換を行います。	1	ダウンロード
コンピュータの仕組み (論理回路)	AND・OR・NOT回路などを論理回路シミュレータで学びます。	1	ダウンロード
アルゴリズム 探索	アルゴリズムの例として探索を学習します。	1	ダウンロード
アルゴリズム ソート	バブルソートを題材にソートを学習します。	2	ダウンロード
プログラミング入門	印刷教材のPython or JavaScriptから選択してご利用下さい。	6	-
モデル化とシミュレーション	APSの権利計算アプリをご利用下さい。	2	-
プログラミングの実践 (オリジナル作品作り)	APSのおみくじアプリ等をご利用下さい。	4	-

Monaca Educationスタンダードプラン 動画『3章：コンピュータとプログラミング』

非公開のYouTube再生リスト

1本あたり5分から10分程度の動画

1. アルゴリズム バブルソート ①～④
2. アルゴリズム 探索 ①～④
 - 二分探索
 - 計算量
3. コンピュータの仕組み（論理回路） ①～⑥
 - 論理演算
 - 3つの論理回路
 - 論理回路シミュレータ
 - 真理値表
 - 半加算回路



Monaca Educationスタンダードプラン 動画『4章： 情報通信ネットワークとデータの活用』

非公開のYouTube再生リスト

1本あたり5分から30分程度の動画

1. 情報通信ネットワークの仕組み
2. 情報セキュリティ
3. 情報システムとサービス
4. 基本統計量①～②
5. 散布図・相関係数・回帰直線①～④
6. 情報システムとデータベース①～③
7. 情報システムとデータベース応用①～⑤



あんこエデュケーション

<https://anko.education>

プログラミング教育のためのサンプルアプリ教材サイト

本サイトは『Monaca Education』の姉妹サイトです

 あんこ
エデュケーション

初級 中級 上級 素材集 ツール集 ブログ

プログラミング教育のための
サンプルアプリ教材サイト

 Monaca(もなか)から飛び出したあんこ
サンプルアプリを提供中(指導案はありません)。

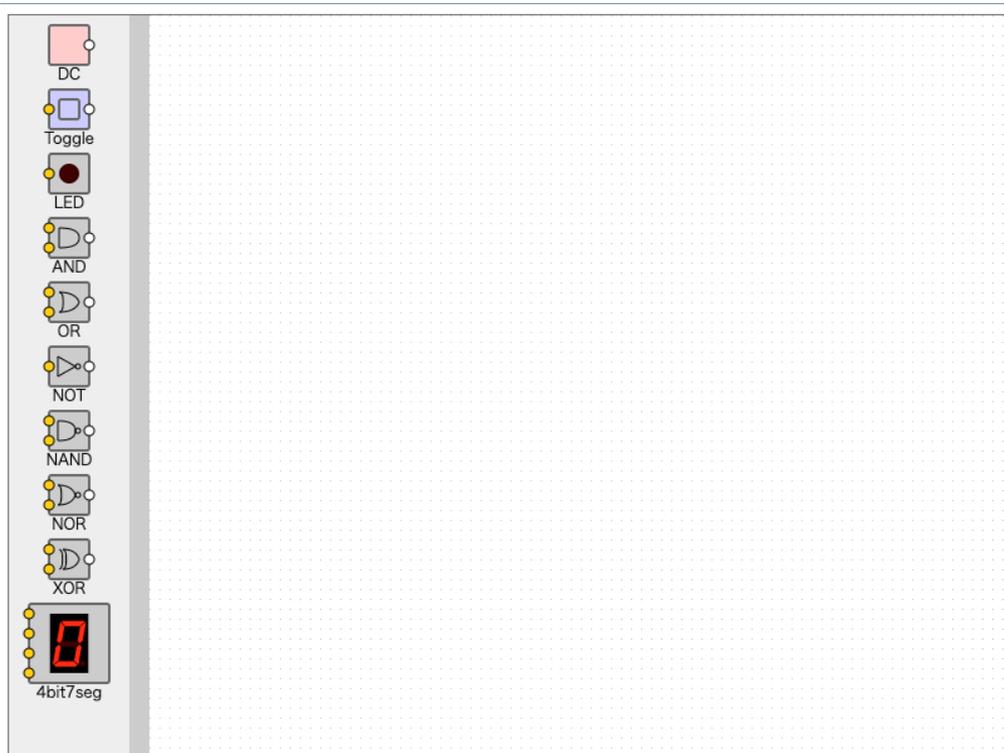
 甘さ控えめ
難易度別に初級・中級・上級を用意。

 有料広告はありません
授業での利用も安心。

ツール集：論理回路シミュレータ

ブラウザで、次のURLにアクセスする

<https://anko.education/tools/simcirjs/>



Copyright (c) 2014 Kazuhiko Arase
Licensed under the MIT license



(参考) あんこエデュケーション
<https://anko.education>

アシアルが運営するWebサイトです。
ツールや資料、情報科関連のブログ記事等を掲載しています。

授業中も利用しやすくするため、
広告無しにしてあります。

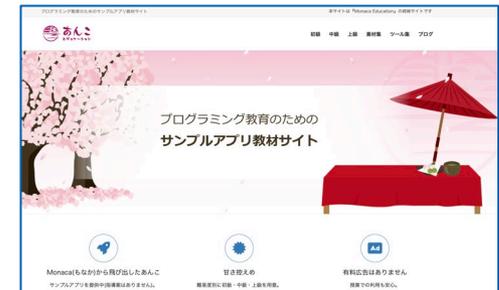
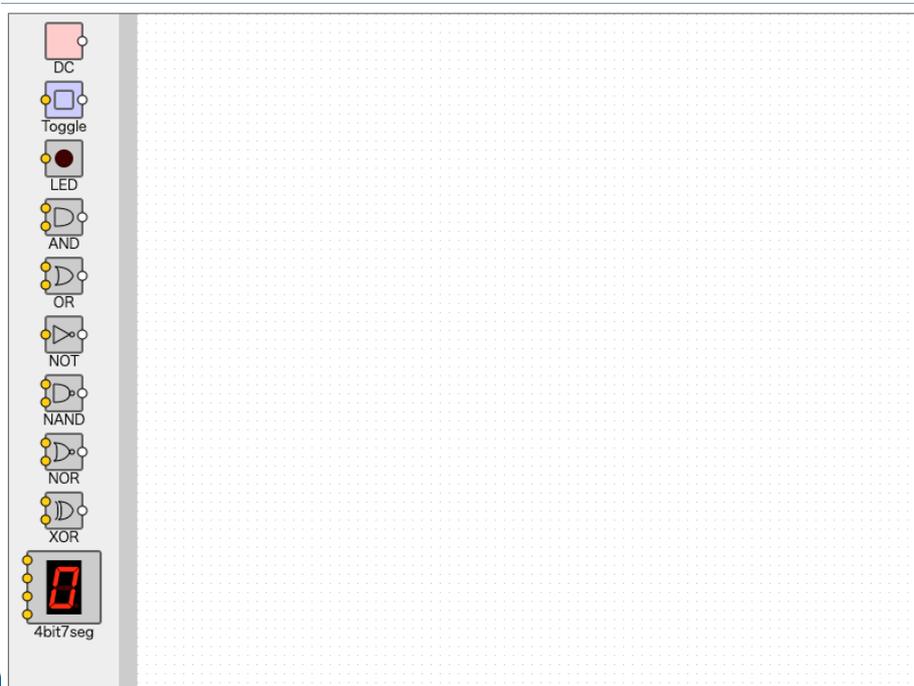
試作問題 第1問 問3

論理回路の問題

論理回路シミュレータとは

Webブラウザで利用できる論理回路のシミュレーションツール
ブラウザで、次のURLにアクセスする

<https://anko.education/tools/simcirjs/>



(参考) あんこエデュケーション
<https://anko.education>

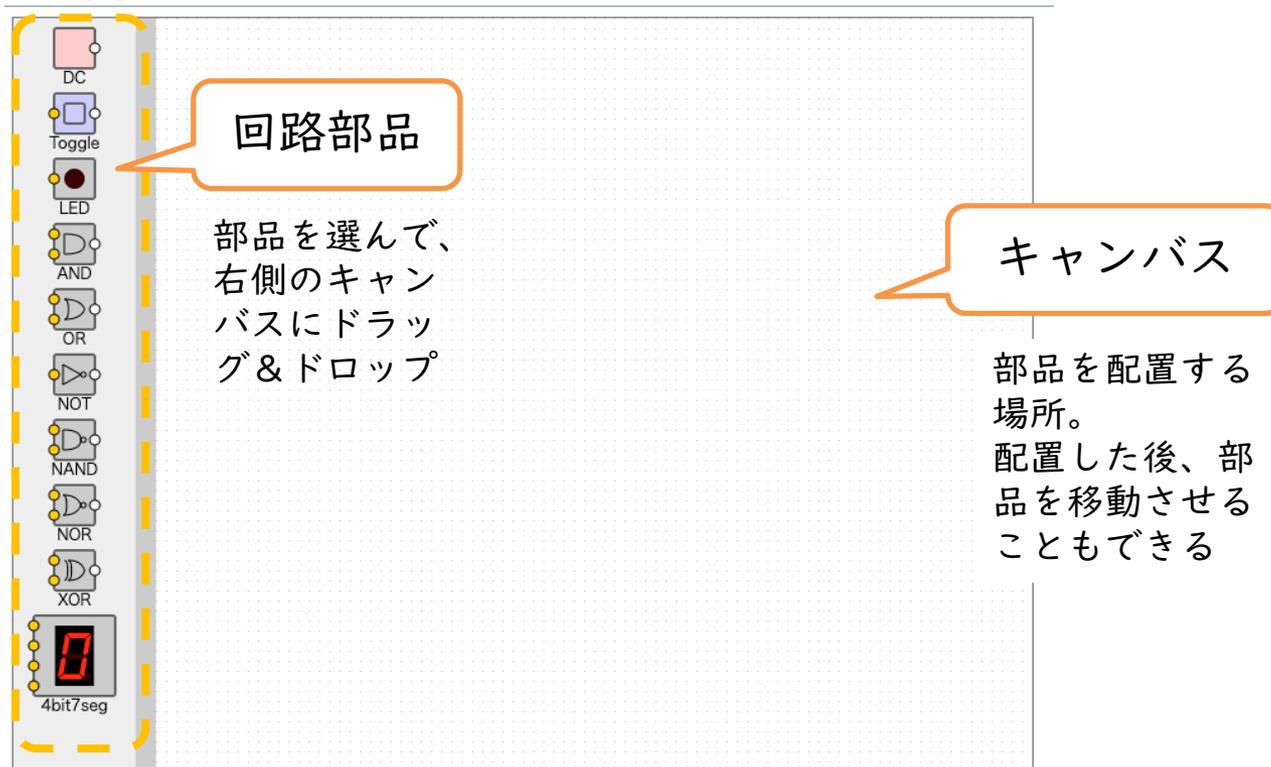
アシアルが運営するWebサイト
です。
ツールや資料、情報科関連のブ
ログ記事等を掲載しています。

授業中も利用しやすくするため、
広告無しにしております。

試作問題 第1問 問3

論理回路シミュレータの使い方①

論理回路シミュレータの画面

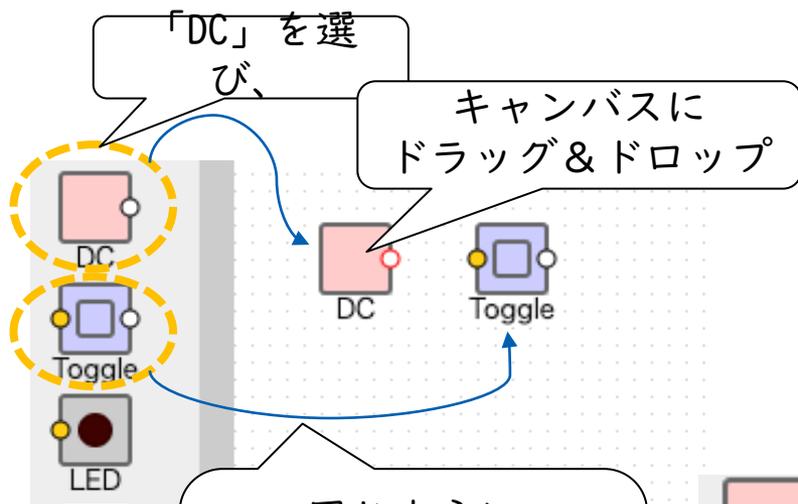


Copyright (c) 2014 Kazuhiko Arase
Licensed under the MIT license

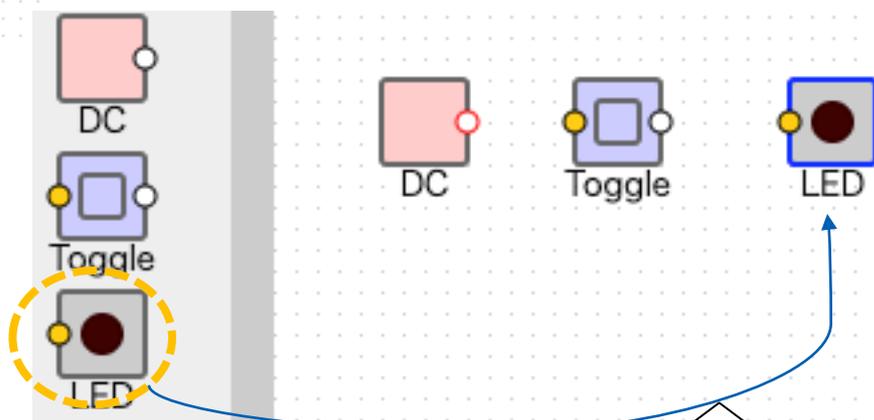
試作問題 第1問 問3

論理回路シミュレータの使い方②

論理回路シミュレータで、
最初の回路を作ろう



同じように
「Toggle」を選び、
キャンバスに
ドラッグ&ドロップ

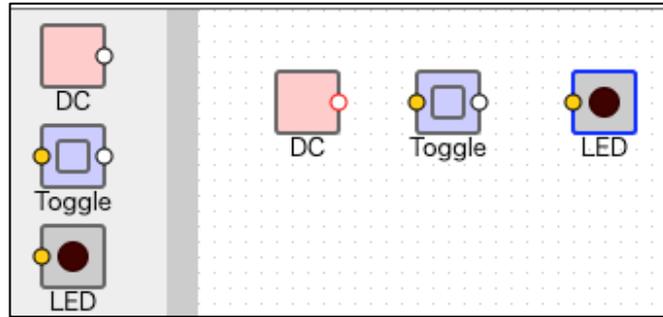


最後に、「LED」を
ドラッグ&ドロップ

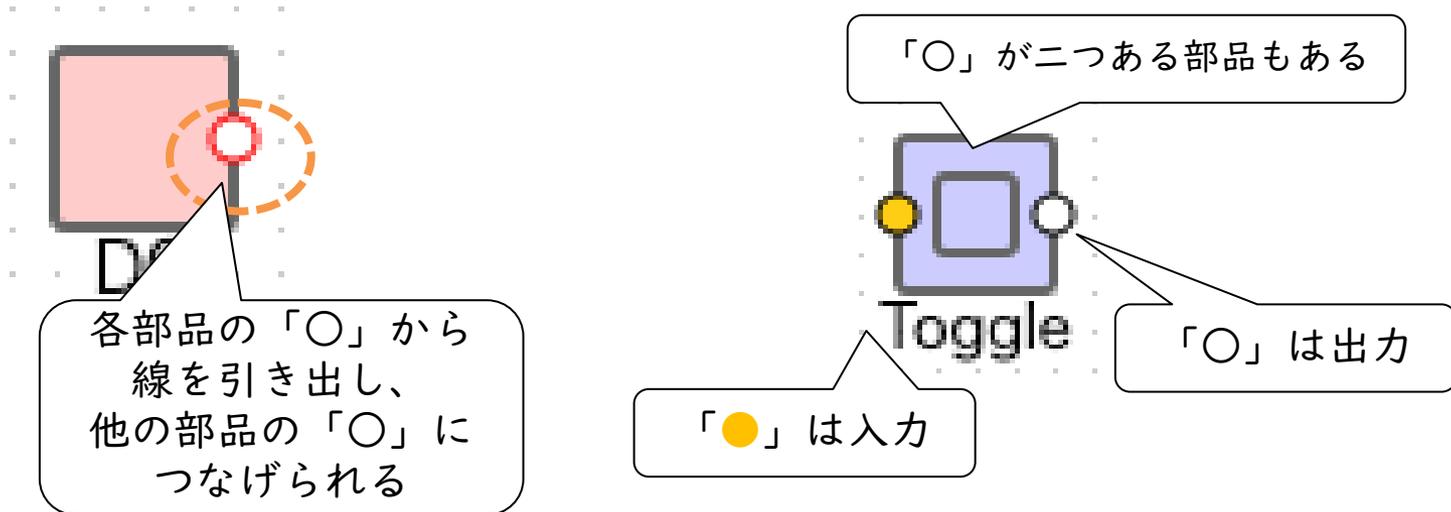
試作問題 第1問 問3

論理回路シミュレータの使い方③

部品を並べたが、部品はつながっていない



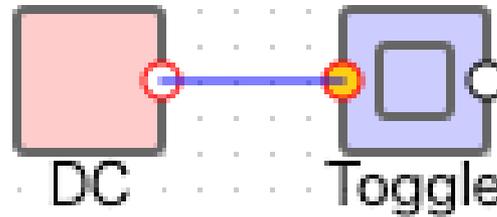
つなげるには



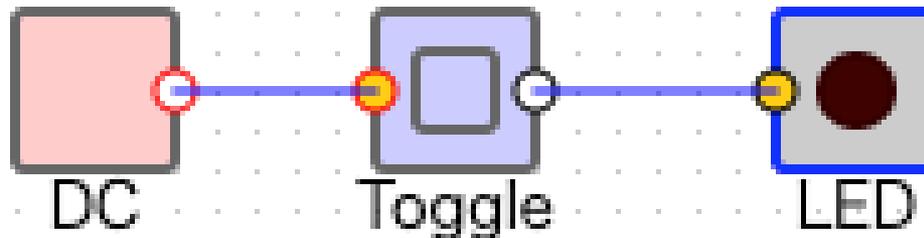
試作問題 第1問 問3

論理回路シミュレータの使い方④

「○」から「○」へドラッグし、部品をつなぐ



三つの部品をつないだら、最初の回路の完成!

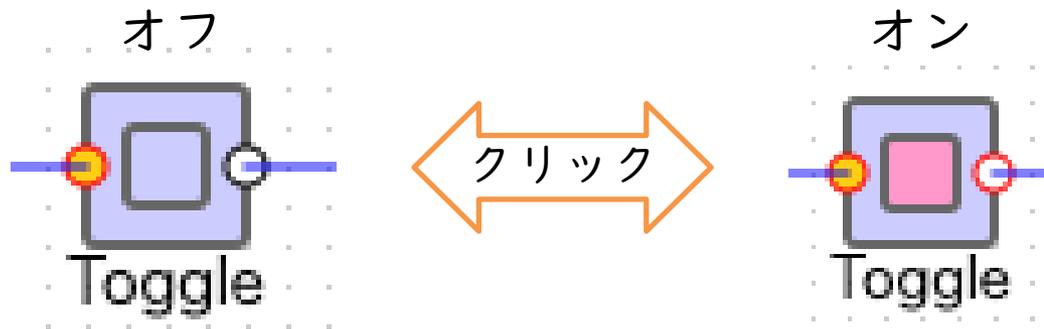


試作問題 第1問 問3

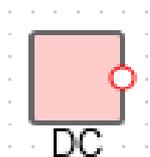
論理回路シミュレータの使い方⑤

最初の回路を動作させよう

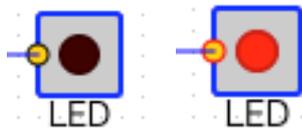
部品Toggleは、「スイッチ」。クリックで、スイッチのオン/オフを切り替える



他の部品



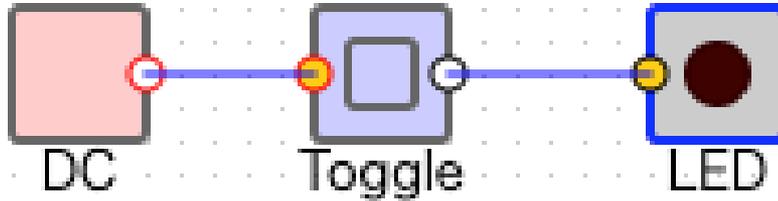
DC: 電源のこと。他の部品（回路）に電流を送る



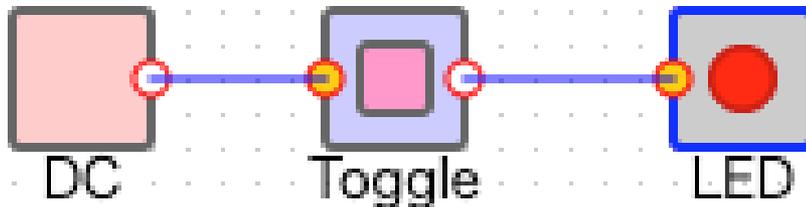
LED: LED電球のこと。電流が流れると、赤くなる（点灯する）

試作問題 第1問 問3

論理回路シミュレータの使い方⑥



トグルスイッチをクリックすると、
スイッチがオン/オフされる。
電流の有無で、
LEDが点いたり消えたりする



※ オフ → 0
オン → 1
消灯 → 0
点灯 → 1
と見なすことができる

試作問題 第1問 問3(2)の論理回路を作ってみよう

(2) S航空会社では新しい旅客機を購入することにした。この旅客機では、トイレを三つ(A・B・C)に増やし、三つのうちどれか二つ以上が使用中になったら混雑を知らせるランプを点灯させる。入力や出力は(1)と同様とする。この場合の真理値表は で、これを実現する論理回路は図3である。

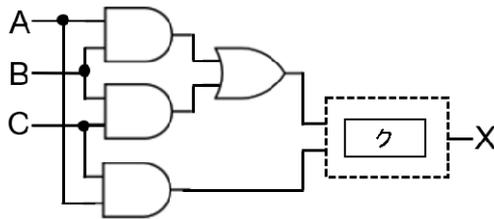


図3 (2)の論理回路

ヒント:

- 少しずつ作る
- 真理値表を作りながら、付け足していく

, の解答群

①	②	③
④	⑤	⑥

※試作問題から引用

※あんこエデュケーションに設置してあるシミュレータは、「保存」機能が無いため、大きく複雑な回路を作るのには向いていません。

ツール集：光の三原色 (RGB)

ブラウザで、次のURLにアクセスする

https://anko.education/tool/app_rgb

3つの入力 (R、G、B) の
組み合わせで、
1つの出力 (色) を得る

サンプルプログラム「光の三原色」

カラーコード

カラーコード：#

RGB(赤・緑・青)の値を入力

R:赤(0-255)

G:緑(0-255)

B:青(0-255)

色の見本



あんこエデュケーションには、他にも多数のツール、サンプルを掲載しています

あんこエデュケーション ブログ

Monaca Educationで大学入学共通テストの試作問題に取り組む～釣り銭が少ない上手な払い方

<https://anko.education/archives/2979>

Monaca Educationにログイン済みのWebブラウザで別にタブを開き、上記URLにアクセスする。

または、あんこエデュケーション(<https://anko.education/>) トップページから「ブログ」をたどる



あんこエデュケーション サンプルアプリ

アシアルが公開しているサンプルアプリ (APS) の カスタマイズ例を紹介・解説



Monaca Education上で簡単に動かせるアプリを紹介しています。

生徒・学生の方へ

大半のアプリはパラメーターを変えたり1行追加するだけでカスタマイズできます。

授業でMonaca Educationを使う方は指導者の指示に従って下さい。

なお、指導者がいない方はMonaca Educationの以下の記事やMonaca Educationの市販の
[はじめてのMonaca Education \(一般向け\)](#)

ソーメンタイマー

ソーメンタイマー(BASIC)

01:20

スタート!

ソーメンの茹で時間をカウントしてくれるアプリです。

複数の音声ファイルが内包されており、カウントダウンとともに用意した音声ファイルを順々に再生しています。