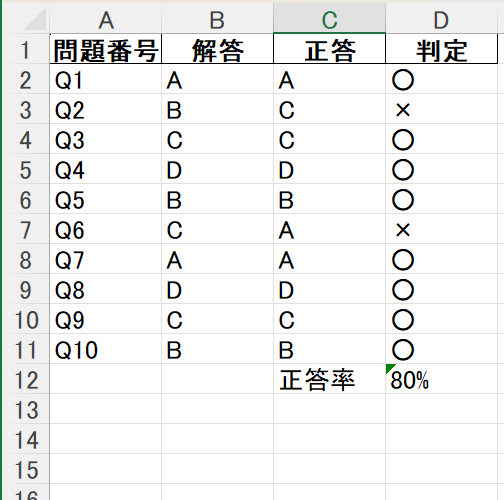
# VBA 最終課題（2025/9/19(金)出題）

## 課題内容

あなたは、毎日100名の受験者から送られてくる「小テスト（Excelファイル）」を採点する仕組みを作成することになりました。

* 各ファイルには60問の問題があります。
* 問題はシートごとに1問ずつ配置されています。
* 受験者は、各シートのセル **A20** に解答を入力します。
* 採点結果は、新たに**「点数」シート** を作成してまとめてください。

【サンプルの点数シート（完成イメージ図）】



こちらが **「点数」シートの完成イメージ図（サンプル）** です。

例として5問分を表示し、最後に正答率をまとめています：

* **問題番号**：Q1〜Q5
* **解答**：受験者の入力（例：A, B, C...）
* **正答**：模範解答
* **判定**：○×判定
* **正答率**：最後にまとめて表示

## 必須要件（全員が取り組む内容）

1. 全60問分の受験者の回答を「点数」シートを作成し、一覧化すること。
2. 正答と照合し、○×の判定を表示すること。
3. 「点数」シートを作成し、正答率を計算して表示すること。

👉 ここまでできれば合格です。

## 応用課題（できる人は挑戦してください）

1. 複数ファイルを一括で処理し、まとめて採点できるようにすること。
2. 「点数」シートのレイアウトを見やすく整えること（罫線・色付けなど）。
3. エラーハンドリングを追加すること（解答が未入力の場合など）。
4. 受験者ごとの結果を集計し、ランキングを出すこと。
5. 設定シートを作り、正答一覧や対象フォルダを切り替えられるようにすること。

## 提出形式

* ExcelファイルにVBAマクロを組み込み、.xlsm　と、.bas ファイルを提出してください。
  + 特に説明はしてこなかったのですが、.bas　ファイルはGitHub上で内容が確認できるため、コードの共有(GitHubを使用する目的)に大変有用です。可能な限り、この提出形式に慣れるようにしてください。
* 必須要件に加え、発展課題を実装した場合は、コメントで工夫点を説明してください。

## 評価基準

* 必須要件を満たしているか（基礎力）
* コードの正確さ・効率性（実装力）
* 発展課題への取り組み（応用力）
* コメントや構造化の工夫（可読性・設計力）

👉 この課題を通じて、「要件定義 → 設計 → 実装」の流れを意識しながら、**実務で役立つVBA力** を身につけてください。

# VBA 発展課題

## テーマ：複数ファイルの一括採点（バッチ処理）

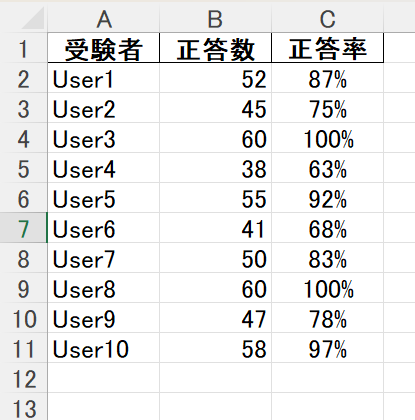
### 課題内容

あなたは、受験者100名が提出した「小テスト（Excelファイル）」を一括採点する仕組みを作成することになりました。

* 受験者は 60問入りのExcelファイルを1人1つ 提出します。
* 提出されたファイルは、1つのフォルダにまとめられています。
* 各ファイルの構成は共通で、各シートのセル A20 に解答が入力されています。
* 採点ルールは、前課題（点数シート作成）と同様に「正答と照合 → ○×判定 → 正答率を計算」とします。

**今回の目標は、外部にVBA入りの「集計用マクロファイル」を用意し、そのファイルから一括採点処理を行うこと です。**

**「集計シート」の完成イメージ図（サンプル）**

****

### 必須要件

1. ユーザーにフォルダを選択させ、その中のExcelファイルをすべて対象とすること。
2. 各受験者ファイルを開き、全60問を採点すること。
3. 採点結果を、マクロ実行元の「集計」シートにまとめること。  
   * 1人につき1行に記録（例：ファイル名／受験者名、正答数、正答率）。

### 追加の発展課題（挑戦できる人向け）

1. エラー処理を追加する（解答が未入力／ファイルが壊れている場合など）。
2. 受験者ごとに「点数シート」をコピーして保存する機能を作成する。
3. 集計結果に基づき、ランキング表 を自動生成する。
4. 成績の分布（ヒストグラム、平均点など）を「統計シート」に出力する。
5. 設定シートを作成し、模範解答や対象フォルダを切り替えられるようにする。

### 提出形式

* **「集計用マクロファイル（xlsm形式）」と、そのマクロだけを取り出した .bas をGitHubに提出してください。**
* **必須要件に加えて発展課題を実装した場合は、コメントや説明シートに工夫点を記載してください。**
* **さらに改善提案がある場合は、「別途提案書」としてまとめて提出してください。**
  + **例：画面設計やフォームを使った入力方式**
  + **より効率的な処理方法のアイデア**
  + **実務での活用を想定した改善案**
  + **別途提案書サンプル　>>**

### 評価基準

* 必須要件：一括処理が正しく動作しているか。
* コードの正確性：エラーなく処理できるか。
* 効率性・可読性：処理速度やコードの見やすさ。
* 応用力：発展課題への取り組みや工夫。

👉 この課題を通じて、「1つのファイルを処理する」から「複数ファイルを一括処理する」へとステップアップし、

業務で通用する自動化スキル を身につけてください。

**>> 提案書テンプレート >>**

### 追加チャレンジ（任意）

なお、細かい要件定義は与えていませんが、**要件定義書を作成できる人は、別途提出してみてください**。  
 実務ではコードを書き始める前に要件定義を行うことが多いため、良いトレーニングになります。

👉 要件定義書のテンプレートを配布しますので、参考にしてください。

**>> 要件定義書テンプレート　>>>**

**※なお、「別途提案書サンプル」や「要件定義書テンプレート」はあくまでも簡略版です。**

**より実戦的に詳しく知りたい、または作成したい場合は、ネットで「提案書」や「要件定義書」などで検索してみてください。**

**オススメリンク。（興味がある人はこちら👇を参考にしてみてください）**

**>> IPA >>** [はじめての要件定義で読むべきIPAドキュメント - Qiita](https://qiita.com/cask_st/items/5fd70bda63f9acf22a30) **>> デジタル庁 >>** [デジタル社会推進標準ガイドライン](https://www.digital.go.jp/resources/standard_guidelines)