

プログラミングB

第3回レポート課題

担当教員 武政 淳二

令和7年12月4日

■ 課題の目的

構造体、文字列、ファイルポインタの操作に慣れ、理解すること。

■ 課題内容「住所検索」

以下に挙げる2種類の検索を実現するプログラムを作成し、レポートにまとめよ。

(検索1) 郵便番号を入力すると、対応する住所を出力する。

(検索2) 住所名の一部を表す文字列を入力すると、その文字列を含む一連の住所およびそれらに対応する郵便番号のリストを出力する。ただし、リストは郵便番号で昇順にソートすること。

ただし、プログラムは以下の2段階の処理を実行すること。

1. 前処理

検索を容易にするための前処理を担当する。郵便番号と住所を含むテキストファイル（付録参照）を入力として、構造体を利用してメモリに格納する。

2. 検索処理

(検索1) もしくは(検索2)を処理し、検索結果を画面に出力する。

郵便番号データおよび入力仕様については本資料末尾の付録を参考すること。

また、その他のプログラム仕様については、本資料後半の章”プログラムの仕様”にまとめているのでよく読むこと。

プログラムの作成にあたっては、特段の理由がない限り、CLEの本課題フォルダに同梱

のテンプレートプログラムに加筆して完成させること。特に、入出力の際には、本テンプレートプログラムで定められている入出力の仕様を変更しないこと。

本プログラムは日本語の文字操作を含むので、文字コードに気を付けること。テンプレートプログラムは UTF-8 の文字コードを使用して、作成している。テキストエディタで開く際、あるいは保存する際、は文字コードの設定を UTF-8 に変更すること。

また、実行環境として、推奨環境である Msys2/MinGW、他に Windows Subsystem for Linux (WSL)、Linux を想定している。もし、他の環境を利用する場合、文字コードに関する問題が生じても、自身で調査し解決すること。なお、Windows のコマンドプロンプト、PowerShell の利用者については、これらの環境に対応するためのコードをテンプレートプログラムにコメントとして追加しているので、よく読んでコメントを外して利用すること。

■ 提出物

□ プログラム & レポート

作成したソースプログラム、および、本課題フォルダに同梱の「レポートの書き方」にしたがって作成したレポート（PDF ファイルとする）を CLE に提出すること。また、ソースプログラムの提出後、動作確認を slack で依頼すること。

動作確認依頼の方法は CLE にアップロードしている通りである。動作確認依頼時に、「/動作確認 課題〇-学籍番号-氏名-OS 名-文字コード名」をメッセージ欄に入力し送信すること。「動作確認を依頼しました」と表示されることを確認すること。OS 名には、動作確認に使用した PC の OS 名を、Windows・Mac・Linux・その他、のいずれから選択し記入すること（Ubuntu, Debian, CentOS, Red Hat 等は Linux に含まれる）。文字コード名には、sjis あるいは utf8 のうち、選択した方を記入すること。

提出ファイルの名前は「学籍番号下 4 行.c」、「学籍番号下 4 行.pdf」とすること。例えば、学籍番号が 09B16099 であれば、ファイル名は 6099.c, 6099.pdf とする。ソースプログラムとレポートは異なるタイミングで提出してよい。（むしろ、動作確認を早期に行えるので推奨。）

■ 締切

■ 項目	期日
ソースコード	1/4 23:59 (演習の講義 14 回目の前日)
教員・TA による動作確認	基本的には、12/4~1/5 の演習の講義中
レポート	1/8 23:59 (演習の講義 16 回目の前日)

締切後の課題提出は大幅な減点を覚悟すること。ただし、期限内に提出したプログラムであって、第 14 回演習の時間内に教員・TA から誤り（バグ）を指摘されたものについては、当日中に修正・提出し、後日動作確認が取れれば減点対象とはならない。

■ 発展課題

以下の発展課題は義務ではないが、取り組んだ場合はプラス評価する。

(発展課題) 絞り込み検索の機能を実装せよ。ここでいう絞り込み検索とは、(検索 2) の検索結果に対して、さらに新たな文字列を用いて検索を行うことである。たとえば、「大阪」で検索して抽出した住所の集合のうち、さらに「吹田」を含む住所を抽出して提示する行為を指す。なお、絞り込み検索は 1 度だけでなく、ユーザが望む限り、何度も繰り返し行えるように実装すること。たとえば、「大阪」、「吹田」、「東」、「山」をすべて含むような住所の検索が行えねばならない。

また、本課題は、以下の 2 つの観点で工夫のしどころがある課題である。レポートに、どのような工夫を行い、どの程度の効果が得られた（と考えられるか）を記載した場合、その程度に応じて加点評価を行う。

・ 時間計算量（検索時間）の削減

検索 1、検索 2、絞り込み検索の検索時間をどのようにすれば短縮できるか。たとえば、郵便番号による検索（検索 1）については前処理段階で工夫を施しておこくことで効率の良い検索が可能になるかもしれない。また、単純なプログラムであれば、検索 2 において入力された文字列（クエリ文字列）と「大阪」といった頻出地名の比較を多数回行うことになるが、この無駄を省くことによって検索時間を削減できる余地があるかもしれない。

・ 空間計算量（メモリ消費量）の削減

「広島県」や「大阪市」といった都道府県名、市町村区名は、住所データのなかで大量に現れる。この特性を活かすことで、メモリ消費量を少なく抑えることができるかもしれない。

■ プログラムの仕様

□ 実行モード

本プログラムは、実行時に与える引数の有無に応じて、以下の 2 つのモードで動作する。

- インタラクティブモード（引数無し）
 - プログラム実行中に、検索モードと検索クエリを選択し、住所検索を行うモード。
 - 引数に何も指定せずに実行すると本モードで動作する。例えば、(プログラム名を kada13 と仮定し) ./kada13
 - コーディング中の動作確認などに使うことを推奨
- ファイルモード（引数有り）

- プログラム実行時に、検索モードと検索クエリの列を記入した入力ファイル名を引数として指定し、住所検索を行うモード
- 引数に入力ファイル名を指定して実行すると本モードで動作する。例えば、./kadai3 input_a_utf.txt
- 動作確認時は、本モードでテストを行う。

□出力の仕様

以下の通り、検索結果を出力すること。

- [郵便番号]:[住所] の形式で出力すること。":の前後に空白を加えないこと
 - 例えば、5650871:大阪府吹田市山田丘
- 検索結果以外は、printf 等で標準出力しないこと。

■ テストケース

提出するプログラムは、下記の手順でテストケースをクリアすることを提出前に確認しておくこと。

- (プログラム名を kadai3 と仮定し) ./kadai3 input_a_utf.txt > myoutput.txt
 - kadai3 に検索クエリを入力し、標準出力の内容を myoutput.txt に保存
- diff myoutput.txt output_a_utf.txt を実行し、相違がないことを確認

なお、冒頭の課題内容でも述べた通り、検索結果は郵便番号で昇順にソートして出力したこと。全ての住所を出力できいても、ソートできていない場合、動作確認が通過することは限らないので注意すること。

テストケースの入力ファイル、出力ファイルは本課題フォルダにアップロードしている。

項目	入力ファイル名	出力ファイル名
基本 a	input_a_utf.txt	output_a_utf.txt
基本 b	input_b_utf.txt	output_b_utf.txt
発展	input_hatten_utf.txt	output_hatten_utf.txt

参考までに、本テストケースにおいて、入力ファイルで実施している検索内容と、出力ファイルの内容（期待する出力結果）は以下の通りである。プログラム作成中は、インターラクティブモードで動作させ、以下の検索クエリに対し逐次出力を確認することで、デバッグするのが良いかもしれない。

項目	検索内容	期待する出力結果
基本 a	(検索 1) 5650871	5650871:大阪府吹田市山田丘
基本 b	(検索 2) 小野原	3691804:埼玉県秩父市荒川小野原 5620031:大阪府箕面市小野原東 5620032:大阪府箕面市小野原西

		6692132:兵庫県篠山市今田町上小野原 6692133:兵庫県篠山市今田町下小野原 8912312:鹿児島県鹿屋市小野原町
発展	(検索 2) 石橋 の実行後、絞り込み検索で、 大阪	5600042:大阪府豊中市石橋麻田町 5630032:大阪府池田市石橋

■付録

□郵便番号データ（テキストファイル）の配布元

CLE から data_utf.csv をダウンロードすること。本データの出典元は日本郵政株式会社の Web ページである。

http://www.post.japanpost.jp/zipcode/dl/kogaki/zip/ken_all.zip

下記のページには上記の全国一括版だけでなく、都道府県ごとの縮小版データも配布されている。デバッグなどの検証時には縮小版データを用いた方が効率的な場合もある。

<http://www.post.japanpost.jp/zipcode/dl/kogaki-zip.html>

CSV の各列が何を意味しているかは、下記ページを参照すること。

<http://www.post.japanpost.jp/zipcode/dl/readme.html>

本課題にとって必須な情報は 3,7,8,9 列目であることが分かる。

このデータに関して、注意すべき点を以下にまとめる。

1. 1 つの郵便番号が複数の町域に割り当てられている場合がある。

例：5700043,大阪府,守口市,高瀬旧大枝,0,0,0,1

5700043,大阪府,守口市,高瀬旧世木,0,0,0,1

5700043,大阪府,守口市,高瀬旧馬場,0,0,0,1

5700043,大阪府,守口市,南寺方東通,0,0,1,1

2. カラム 5～8 の値が正しくない場合があり、データそのものに間違いがある。この誤りが原因でプログラムが誤った解を返すのは認める（データが正しいと信じて、プログラムは動作してよい）。

3. 郵便番号データは、住所に漢字や平仮名などの日本語を含んでいる。文字コード (UTF-8) に注意してプログラムを作成すること。