

2164204

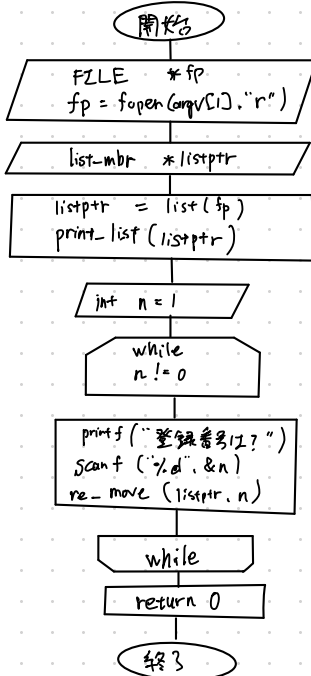
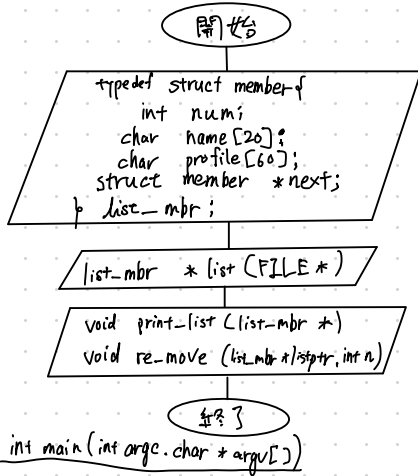
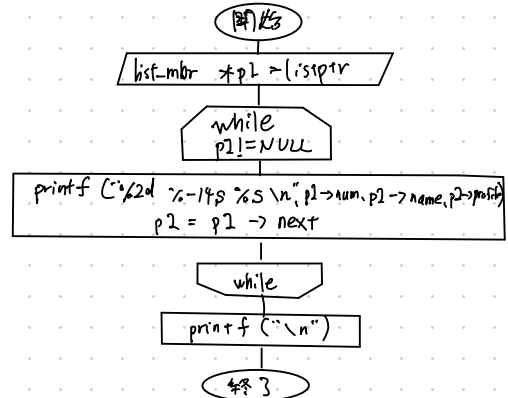
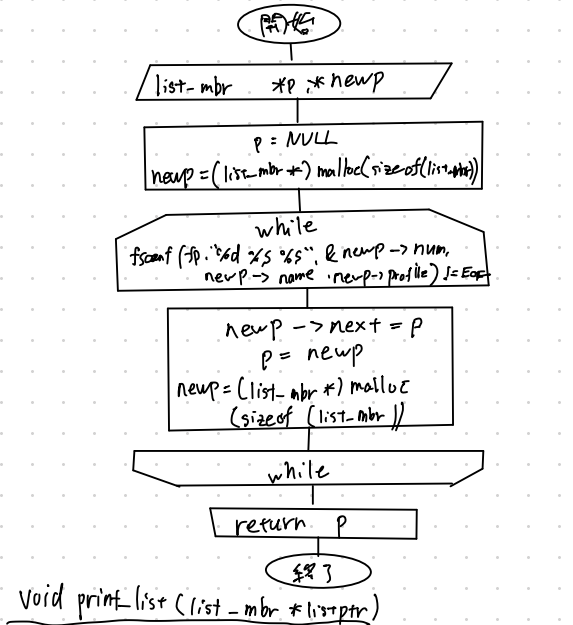
野村 瑛吉

(1) 基本課題3

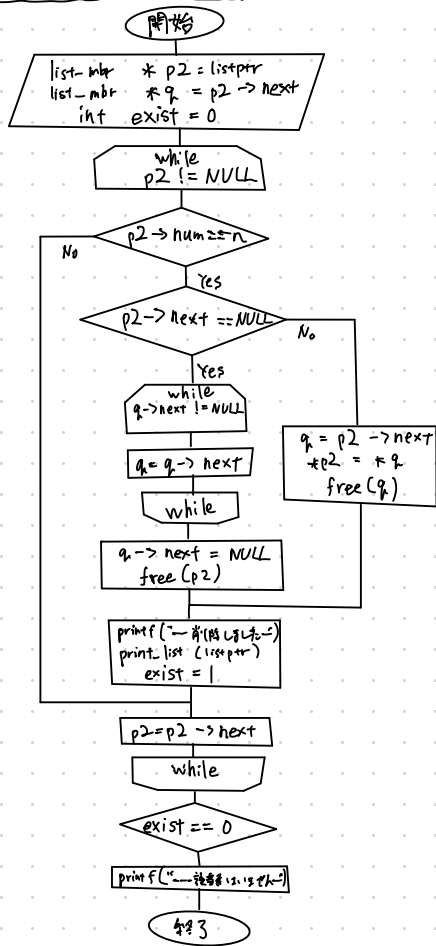
2164204-b3c

線形リストの削除

(2) 課題4

list_mbr */list (FILE *fp)

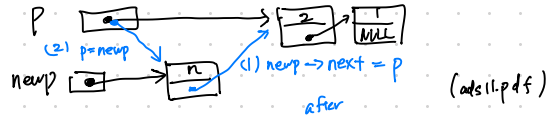
void re-move (list_mbr *listptr, int n)



(3) main関数内で list_mbr の構造体を作成

*listptr を定義。コマンド第三引数の zac-japan.txt を開き、list 関数に代入してファイル名を list に変更し、そのリストを listptr に保存する。

list 関数では、*p (保持用)、*newp (追加用) として、
 2. newp に malloc 関数を用いて 2 行のファイル分の
 保存領域を確保し、ファイルを読み込んでリスト追加。
 malloc 関数で 2 行のデータ保存領域の確保。
 保存したリスト p を戻り値にする。



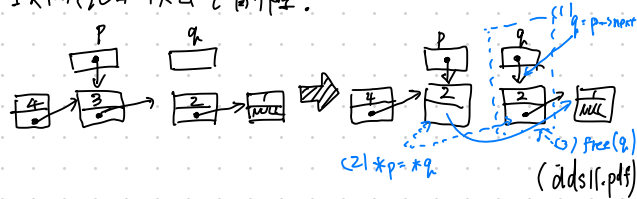
リスト生成後、main 関数では listptr を print_list 関数に
 代入。listptr に保存されたリストを表示。

main 関数で n = 1 (二行目まで while でエラー) としていき、

n = 0 を入力することで、番号を入力させ、その入力した

番号の 2 行目のデータを削除する re-move 関数に n を代入。

while で 番号を入力し、見つかった場合は最後の要素以外で
 1 以下の 2 行目のリストで削除。



最後の要素の場合、最後の要素の一つ前の next を NULL にして

からメモリを解放する。その後、リストを全部表示。

見つからない場合は、その番号を返す。

(4) 遊歩情報の構造体 struct member (name: 登録名, 番号, name[20]: 氏名, 登録名、profile[60]: フォト、+next = 次遊歩名) を list-member と定義。リスト作成関数、リスト表示関数、番号削除関数、フックタイフ宣言、main関数では、ファイルオープン、構造体ポインタ宣言、ポインタのリスト作成。list(fb) では、構造体ポインタのポインタ用、new(追加用) 保存用の最初は NULL にし、追加用でデータ保存領域の確保、while文でファイルを読み込み、保存用に追加用に入ったデータと保存。その後追加用の保存領域を増やす。保存したリストを返す。

main関数では、次に作成したリストの表示をする。print-list関数では、リストが空になるまで表示し、次の番号へというものを繰り返す。

main関数で、番号の更新 n (初期値1) を定義し、nが0が入るまで入力させ続け、re-move関数でリスト内に番号が存在するか探索し、存在したら番号を削除し削除後のリストを返す。

(5) 今回は、線形リストを作るプログラムだった。動的データ構造は初めてだったので、理解に時間がかかった。また慣れないが、今回の構造も死蔵を防止し理解に時間をかけてからプログラム作成できるようにしたい。

(6) アドバイスとデータ構造 11回 ads11.pdf から図を取り。