

コンピュータ実習
第13回 2026年1月8日
演習

カーナビゲーシヨン完成へ向けて

担当:相澤 直人, 宍戸 博紀

コンピュータ実習(第13回)

演習

今回は各自カーナビの作成に
没頭してもらいます。

コンピュータ実習(第13回)

コンピュータ実習の単位取得条件

課題の最低限の合格ライン

- 「カーナビゲーションプログラムを完成させること」
 1. 地図が表示されること
 2. 出発地・目的地が選択できること
 3. 出発地から目的地までの経路を計算で求めること
 - ダイクストラ法もしくは、改良した最近傍隣接交差点検索
 4. 経路上にマーカ―移動の表示をすること

コンピュータ実習(第13回)

コンピュータ実習の単位取得条件

課題の最低限の合格ライン

- 「カーナビゲーションプログラムを完成させること」
 1. 地図が表示されること
 2. 出発地・目的地が選択できること
 3. 出発地から目的地までの経路を計算で求めること
 - ダイクストラ法もしくは、改良した最近傍隣接交差点検索
 4. 経路上にマーカー移動の表示をすること

mobile.c

コンピュータ実習第9回

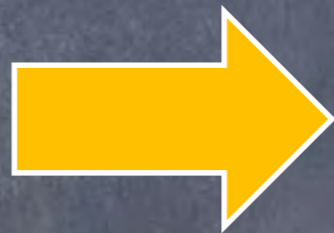
コンピュータ実習(第13回)

地図の表示, マーカーの移動表示

mobile.cで重要な部分

/* 経路を通過する交差点IDの例として, 配列pathに設定する */

```
void path_set(void)
{
    int i=0;
    path[i++] = 60;
    path[i++] = 85;
    path[i++] = 48;
    path[i++] = 49;
    path[i++] = 50;
    path[i++] = 21;
    path[i++] = 22;
    path[i++] = 23;
    path[i++] = 16;
    path[i++] = 17;
    path[i++] = -1; /* -1 は経路の終りを表す */
}
```



path[0]から順に通過する交差点ID(最後は-1)を納めてやれば, その順番に地図上をマーカーが移動していくプログラムになっている.

まず, この部分を消して, 代わりにdijkstra関数とpickup_path関数を書けばよい.

pickup_path関数はpath[n]の中に通過交差点を納めてくれる.

コンピュータ実習(第13回)

コンピュータ実習の単位取得条件

課題の最低限の合格ライン

- 「カーナビゲーションプログラムを完成させること」
 1. 地図が表示されること
 2. 出発地・目的地が選択できること
 3. 出発地から目的地までの経路を計算で求めること
 - ダイクストラ法もしくは、改良した最近傍隣接交差点検索
 4. 経路上にマーカー移動の表示をすること

dijkstra2.c

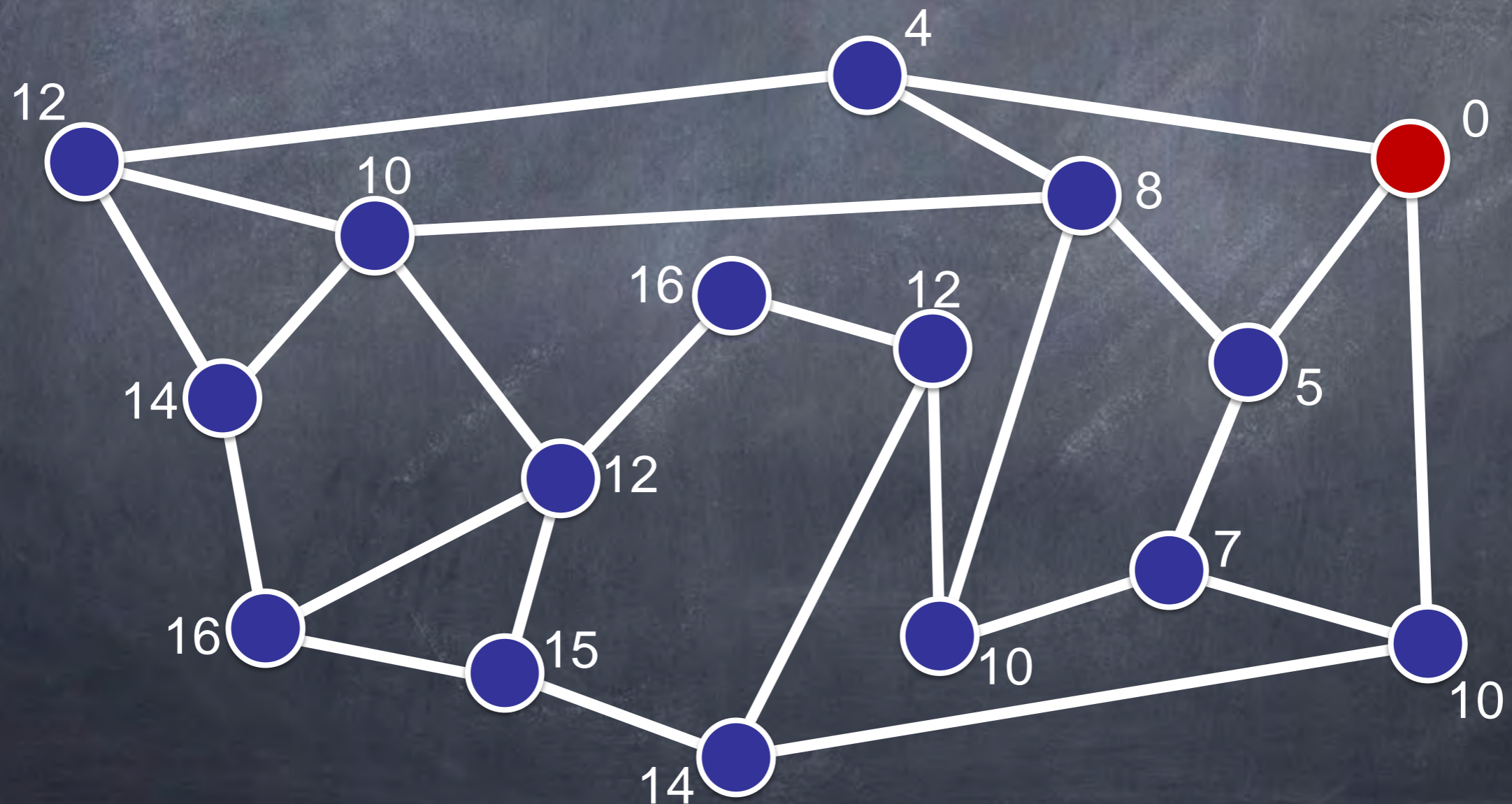
コンピュータ実習第10回 演習2

コンピュータ実習(第13回)

出発地から目的地までの経路計算

dijkstra関数

- dijkstra関数は、指定した基準交差点とそれ以外のすべての交差点との最短距離のみを計算し、cross構造体内の変数に納めている。

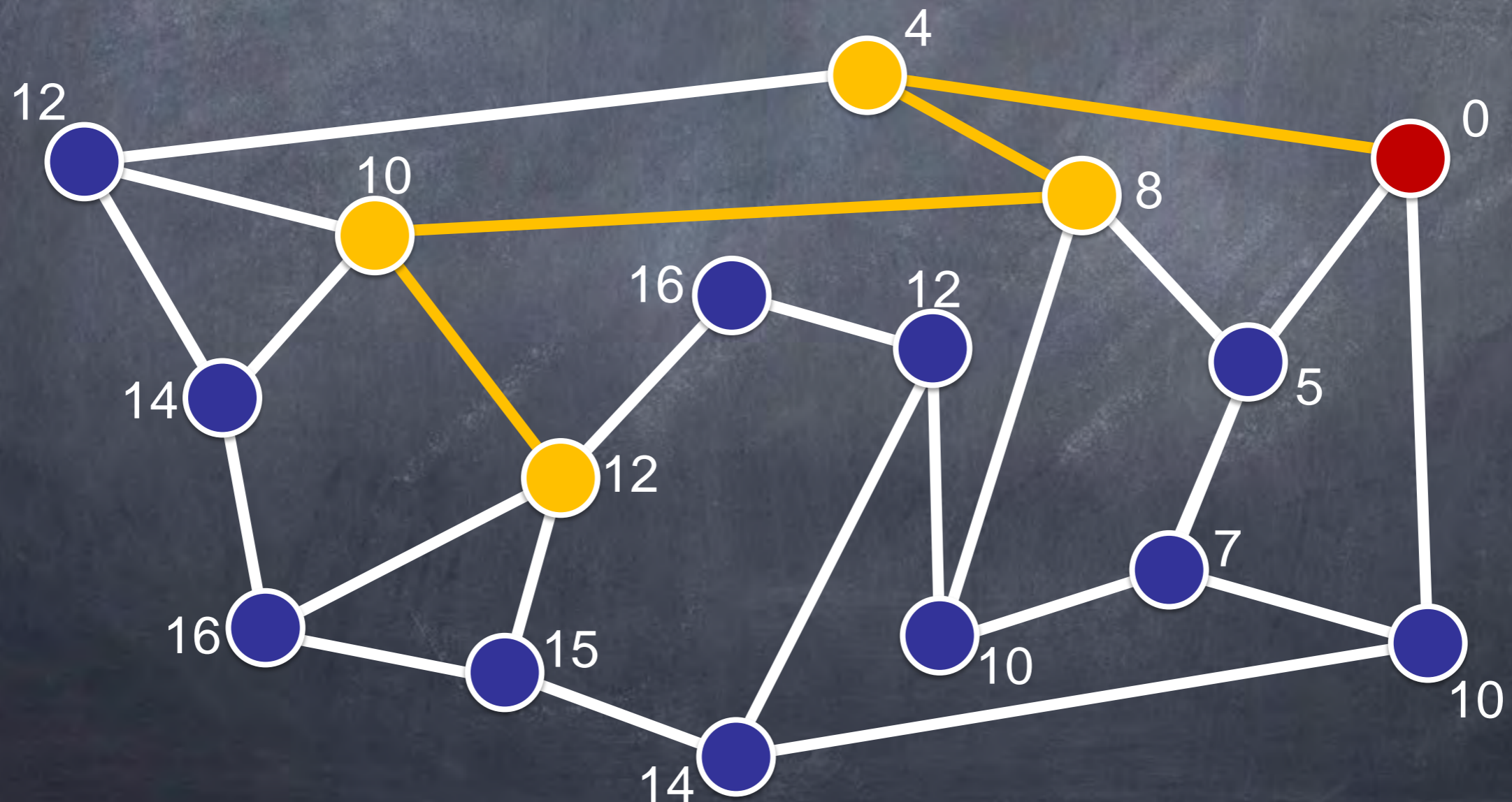


コンピュータ実習(第13回)

出発地から目的地までの経路計算

pickup_path関数

- pickup_path関数は、dijkstra関数の計算結果を基にして、指定した交差点から基準交差点までの最短距離を示す交差点のつながりを抽出し、path[]の中に納めている。



コンピュータ実習(第13回)

出発地から目的地までの経路計算

具体的には・・・(mobile.c main関数)

```
・  
・  
/* マップファイルの読み込み */  
crossing_number = map_read("map2.dat");  
if (crossing_number < 0) {  
    fprintf(stderr, "couldn't read map file¥n");  
    exit(1);  
}
```

path_set(); /* 経路の設定 */

```
/* 移動体の位置を初期化 */  
vehicle_pathIterator = 0;  
vehicle_stepOnEdge = 0;
```

```
for(;;){  
    int width, height;
```



```
・  
・  
/* マップファイルの読み込み */  
crossing_number = map_read("map2.dat");  
if (crossing_number < 0) {  
    fprintf(stderr, "couldn't read map file¥n");  
    exit(1);  
}
```

**dijkstra関数の実行
pickup_path関数の実行**

```
/* 移動体の位置を初期化 */  
vehicle_pathIterator = 0;  
vehicle_stepOnEdge = 0;
```

```
for(;;){  
    int width, height;
```

```
・  
・
```

コンピュータ実習(第13回)

コンピュータ実習の単位取得条件

課題の最低限の合格ライン

- 「カーナビゲーションプログラムを完成させること」
 1. 地図が表示されること
 2. 出発地・目的地が選択できること
 3. 出発地から目的地までの経路を計算で求めること
 - ダイクストラ法もしくは、改良した最近傍隣接交差点検索
 4. 経路上にマーカ移動の表示をすること

ex8-1.c

コンピュータ実習第8回

コンピュータ実習(第13回)

出発地・目的地の選択

具体的には・・・(mobile.c main関数)

```
・  
・  
/* マップファイルの読み込み */  
crossing_number = map_read("map2.dat");  
if (crossing_number < 0) {  
    fprintf(stderr, "couldn't read map file¥n");  
    exit(1);  
}
```

dijkstra関数の実行
pickup_path関数の実行

```
/* 移動体の位置を初期化 */  
vehicle_pathIterator = 0;  
vehicle_stepOnEdge = 0;
```

```
for(;;){  
    int width, height;  
    ・  
    ・
```

```
・  
・  
/* マップファイルの読み込み */  
crossing_number = map_read("map2.dat");  
if (crossing_number < 0) {  
    fprintf(stderr, "couldn't read map file¥n");  
    exit(1);  
}
```

dijkstra関数に与える交差点を入力する関数の実行
dijkstra関数の実行
pickup_path関数に与える交差点を入力する関数の実行
pickup_path関数の実行

```
/* 移動体の位置を初期化 */  
vehicle_pathIterator = 0;  
vehicle_stepOnEdge = 0;
```

```
for(;;){  
    int width, height;  
    ・  
    ・
```

main文に付け足した関数は当然定義する必要があります。さらに変数名をdijkstra関数やpickup_path関数に渡せるように合わせてやる必要があります。

コンピュータ実習(第13回)

コンピュータ実習の単位取得条件

課題の最低限の合格ライン

- 「カーナビゲーションプログラムを完成させること」
 1. 地図が表示されること
 2. 出発地・目的地が選択できること
 3. 出発地から目的地までの経路を計算で求めること
 - ダイクストラ法もしくは、改良した最近傍隣接交差点検索
 4. 経路上にマーカー移動の表示をすること

mobile_s.c

コンピュータ実習第9回

コンピュータ実習(第13回)

出発地から目的地までの経路を計算(最近傍隣接交差点検索)

具体的には・・・(mobile_s.c main関数)

```
・
・
/* 移動体の状態を初期化 */
#if 1
    vehicle_edgeFrom = 5; /* 泉大橋から */
    vehicle_goal = 19; /* 道路資料館へ */
#else
    vehicle_edgeFrom = 5; /* 泉大橋から */
    vehicle_goal = 71; /* 八木山動物公園へ */
#endif
vehicle_edgeTo = vehicle_edgeFrom;
vehicle_stepOnEdge = 0;
・
・
```



```
・
・
/* 移動体の状態を初期化 */
vehicle_edgeFromに出発交差点IDを与える関数の実行
vehicle_goalに目的交差点IDを与える関数の実行

vehicle_edgeTo = vehicle_edgeFrom;
vehicle_stepOnEdge = 0;
・
・
```

これで一応は不十分ながらカーナビのできあがりです。しかし、mobile_s.cで使っているアルゴリズムは不完全ですので、search_nearest関数の中身を考え直す必要があります。

コンピュータ実習(第13回)

演習

- 以上をきちんと組み合わせることができれば最低限のカーナビは完成します。後は使いやすく機能を増やしていきましょう。

オリジナリティを評価します。

色やサイズが違うなどのパラメータを変えただけの新しさではなく、機能としての新しさを高く評価する。
よって、他人のコピーでは高得点は望めない。

コンピュータ実習(第13回)

演習

工夫すべき点の例

- **インターフェース**
 - ナビゲーションの操作が破綻しないインターフェース
 - マウスによる交差点選択
 - 目的地までの距離や時間の表示
- **グラフィックス**
 - 地図の拡大縮小・自由な方向へのスクロール
 - 決定した交差点や経路, あるいは道路状態の判りやすい表示
 - 三次元表示
 - ナビゲーション時の現在位置を中心とした地図スクロール
 - 移動方向に頂点を向けた三角形によるマーカー表示
- **アルゴリズム**
 - 文字入力による交差点選択時のインクリメントサーチ(ただしポイントの理解が必要)
 - 混雑度などを考慮に入れた経路計算(混雑度は乱数などを使用して任意に設定), さらには混雑度の変化に伴うナビゲート中での経路再計算
 - 経由地の指定

コンピュータ実習(第13回)

最終面接について(再掲)

最終面接の実施について

最終面接は、1月15日(木)の授業時間に対面で実施します

面接では、1人ずつ順番に作成したカーナビの動作の確認を行います

その際、工夫点、改良点について、事前に提出されたレポートを元に質問します

1月8日までにカーナビゲーションプログラムが完成した人は1月8日に面接を受けても構いません。その場合は、教員に面接希望の旨を知らせてください。1月8日に面接を受けた結果、課題の内容が十分だと判断される場合は1月15日の実習の出席を免除します。

コンピュータ実習(第13回)

最終面接について(再掲)

最終面接に関する提出物について

最終面接に先立ち、作成したカーナビゲーションプログラムのソースファイルと工夫点をまとめたレポートを Classroomの課題提出機能を使って提出すること。提出期限は1月15日14:40までとする。Classroomの課題提出ページは後日連絡します。

(1月8日に面接を受ける場合は、面接前までに提出すること)

- レポートの題名は「2025コンピュータ実習最終課題_学籍番号_名前」とすること
- レポート中にはカーナビプログラムを完成させた際の工夫点を書くこと
どんな些細なことでも良いので自分が考えたところをできる限り書くこと
 - (例)〇〇の色を緑に変えた
- ソースファイルの名前は「report_学籍番号.c」とすること
- 優秀作品に選ばれた場合にお手本として公開することに同意しない場合は、その旨を書くこと。