小テスト #6

今回の小テストの目的

- ポインタ型を理解する.
- リスト（特に線形リスト）を理解する

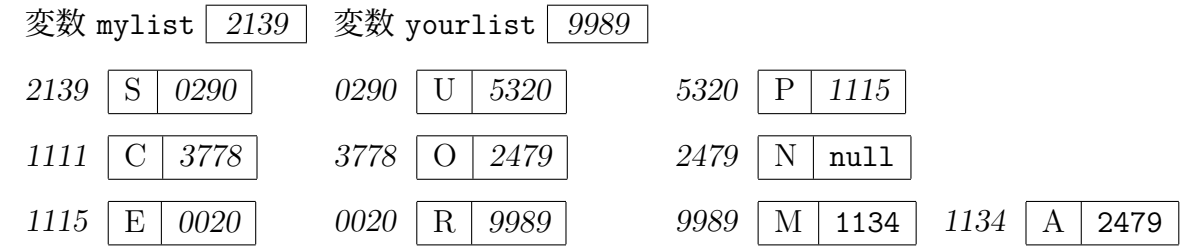
次のような線形リストを考える.

```
struct chrlist {    char data;
                    struct chrlist *next; };
struct chrlist *p;
```

この struct 型 chrlist のリストからなるデータへのポインタ p が与えられたとき，その p が参照するリスト中のデータをすべて出力（画面表示）する手続き `outall` を，次のように定義する.（注：`NULL` は「未定義」を意味する定数.）

```
void outall(chrlist *p) {
// 入力： 線形リストへのポインタ.
// 作業： p が参照するリスト内のデータを順次すべて出力する.
   while(p != NULL) {
      画面に (*p).data を出力;
      p = (*p).next;
   } return;
}
```

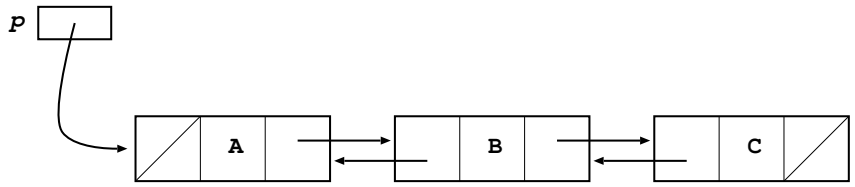
いま，次のような線形リストが与えられているとする．箱の外等に書いてある斜めのフォントの数字は，その箱のポインタ（参照，レファレンス）を表わしているものとする.



(1) 変数 `mylist` が参照している線形リストを図示せよ.

(2) `outall(yourlist)` を実行したとき，`outall` の局所変数 `p` がどのように変化し，また，何が画面に表示されるかを説明せよ.

(3) 次の図のような線形リストの１レコード分のデータ構造を，我々の記述法に従って定義せよ．



(4) 上のようなリストを示すポインタ p をもらったとき，リスト上の文字を逆順（この例では C B A の順）で出力する関数 out_reverse を示せ．