プログラミング II 第 10 回課題

以下のソース・コードは，線形リストを逆順に作成した後，その線形リストを一括消去するものである．一括消去を行う関数 DestroyList() について，空欄のソース・コードを埋めよ．「一括消去」とは，ノードのために確保されたメモリ領域をすべて関数 free() を用いて開放することを意味することに注意．なお，ノードの型名は struct tfield 型を typedef によって NODE 型と呼び替えている．また，ノードに記載するデータは int 型の data のみであり，ポインタの変数名は pt とした．一部の関数では，先頭ノードへのポインタ変数 head (main 関数がもつローカル変数) をポインタ渡しで受け取っていることに注意．
下記のコードは Teams にアップロードされているのでコンパイルしてテストすることが可能である．

```
//
// 第 10 回課題：リストの一括消去
//

#include <stdio.h>
#include <stdlib.h>

struct tfield {
  int data;
  struct tfield *pt;
};

// tfield 構造体の型名を「NODE 型」に呼び替える
typedef struct tfield NODE;

NODE *talloc(void);
void GenerateList(NODE **);
void DisplayList(NODE *);
void DestroyList(NODE **);


void main(void)
{
  NODE *head, *p, *old;

  GenerateList(&head);
  DisplayList(head);

  DestroyList(&head);
  DisplayList(head);
}

void GenerateList(NODE **head) // 逆順リスト生成
{
  NODE *p;
  int   data;

  *head = NULL;
  while (printf("¥nData: "),scanf("%d", &data)!=EOF){
    p = talloc();
    p->data = data;
```

```
    p->pt = *head;
    *head = p;
  }
  printf("¥n");
}

void DisplayList(NODE *head) // リスト一覧表示
{
  NODE *p;

  printf("¥n");

  if (head == NULL) {
    printf("List is empty.¥n");
    return;
  }

  p = head;
  while (p != NULL) {
    printf("->%d", p->data);
    p = p->pt;
  }

  printf("¥n¥n");
}

void DestroyList(NODE **head) // リスト消去
{
  NODE *p, *old;

  //
  // (ここを作成する)
  //
}

NODE *talloc(void) // ノード新規作成
{
  NODE *p;

  p = (NODE *)malloc(sizeof(NODE));

  if (p == NULL) {
    fprintf(stderr, "Cannot allocate memory for node.¥n");
    exit(1);
  }

  return p;
}
```

以上