#include <stdio.h>

#include <stdlib.h>

//単方向リストの構造体を定義します。

struct node {

/\*要 素 の デ ー タ \*/

int data;

// 次の要素

struct node \*next;

};

/\* 新しい要素を作ります。\*/

struct node \*create\_node (int);

/\* 末尾要素を取得します。\*/

struct node \*get\_tail(struct node \*);

/\* 要素を末尾へ追加します。\*/

void add\_node(int, struct node \*);

/\* 単方向リストの記憶領域を解放します。\*/

void release\_list(struct node \*);

int main(void) {

struct node \*head;

/\* 先頭要素をを作ります\*/

head = create\_node (300);

/\* 2つの要素を追加します\*/

add\_node (100, head);

add\_node (200, head);

/\* 単方向リストの記憶領域を解放します \*/

printf("\n");

release\_list (head);

return 0;

}

//新しい要素を作ります。

struct node \*create\_node (int data){

//要素

struct node \*element;

/\* 要素の記憶領域を確保します\*/

element = (struct node \*)malloc(sizeof(struct node));

/\* 要素のメンバを設定・初期化します\*/

element->data = data;

element->next = NULL;

/\* 動作確認用メッセージを表示します\*/

printf("data = %d が作られました.\n",element->data) ;

/\* 要素を返します \*/

return element;

}

struct node \*get\_tail (struct node \*element){

if (element->next == NULL) {

/\*次の要素がない= 末尾要素なので、この要素を返します\*/

return element;

}else{

return get\_tail(element->next);

}

}

void add\_node(int data, struct node \*head)

{

/\* 新しい要素 \*/

struct node \*element;

/\*末尾要素 \*/

struct node \*tail;

/\* 新しい要素を作ります \*/

element = create\_node (data);

/\*末尾要素を取得します\*/

tail= get\_tail(head);

/\* 末尾要素の後ろに新しい要素を接続します \*/

tail->next= element;

}

void release\_list(struct node \*element){

/\* 末尾要素でないかを調べます \*/

if (element->next != NULL){

/\*次の要素から引き続き記憶領域を解放します(再帰呼び出し)\*/

release\_list(element->next);

}

/\* 動作確認用メッセージを表示します \*/

printf("data = %d が解放されました.\n",element->data);

/\* この要素の記憶領域を解放します\*/

free(element);

}

**実行結果**

data = 300 が作られました.

data = 100 が作られました.

data = 200 が作られました.

data = 200 が解放されました.

data = 100 が解放されました.

data = 300 が解放されました.