```c
/*
    (C) OOMusou 2008 http://oomusou.cnblogs.com

    Filename    : DS_linked_list_simple.c
    Compiler    : Visual C++ 8.0
    Description : Demo how to use malloc for linked list
    Release     : 03/22/2008 1.0

*/
 #include <stdio.h>
 #include <stdlib.h>
 #include <string.h>

 #define SLEN 255

 struct list {
   int  no;
   char name[SLEN];
   struct list *next;
 };
 /*
    linked list的基礎就是struct，所以先建立一個自訂的struct型別，因為linked
    list是靠struct串聯起來，所以最後要多一個struct pointer指向下一個struct。
 */

 int main(){
    int no;
    char s[255];

    struct list *head    = NULL;
    struct list *current = NULL;
    struct list *prev    = NULL;
    /*
        建立linked list最基本需要三個指標，head 指向 linked list 的第一個
        struct，current 指向目前剛建立的 struct，prev 則指向前一個
        struct，目的在指向下一個 struct，對於未使用的 pointer，一律指定為
        NULL，這是一個好的 coding style，可以藉由判斷是否為NULL判斷此 pointer
        是否被使用。
    */

    while(1){
        printf("No. = ");
        scanf("%d", &no);

        if (no == 0)
            break;

        printf("Name = ");
        scanf("%s", s);

        current = (struct list *)malloc(sizeof(struct list));
        if(current == NULL)
            exit(EXIT_FAILURE);

        current->next = NULL;
        /*
            每當有新資料，需要建立一個新的 struct 時，就用 malloc()
            要一塊記憶體，由於 malloc() 傳回的是 void *，所以要手動轉型成 struct
```

```
            list *。但 malloc()
            並不是一定會成功，若記憶體不足時，仍然會失敗，所以必須判斷是否傳回
            NULL。由於一個新的 node，一定是 linked list 最後一個 node，所以將
            current->next 接 null。
53          */
54          current->no = no;
55          strncpy(current->name, s, SLEN -1);
56          current->name[SLEN -1] = '\0';
57          /*
58              正式將輸入的資料填進 struct，至於為什麼要用 strncpy() 而不用 strcpy()
            呢?雖然 strcpy() 也可以，但 strncpy() 比較安全，若輸入的字串大小超過
            struct 所定義的字串大小，則會只接受 struct
            所接受的字串大小，而不會因為找不到'\0'而造成程式錯誤。
59          */
60          if (head == NULL)
61              head = current;
62          else
63              prev->next = current;
64          /*
65              判斷若是第一個 node，則將目前的 node 當成 head，若不是第一個
            node，則將前一個 node 指向目前的 node，完成 linked list
            的連接。最後將目前的 node 當成前一個 node，以備指向下一個 node。
66          */
67          prev = current;
68      }
69      // display linked list
70      current = head;
71      while(current != NULL){
72          printf("No. = %d, Name = %s\n", current->no, current->name);
73          current = current->next;
74      }
75      /*
76          要重新顯示 linked list，所以將指標再度指向第一個 node，每當顯示一個 node
            後，就指向下一個 node，直到指到 NULL 為止。
77      */
78      // free linked list
79      current = head;
80      while(current != NULL){
81          prev = current;
82          current = current->next;
83          free(prev);
84      }
85      /*
86          由於 malloc() 是將記憶體放在 heap，而不是放在 stack，所以並不會隨著
            function 的結束而釋放，必須要手動使用 free() 釋放記憶體，否則會造成 memory
            leak。
87      */
88      return 0;
89  }
```