

資訊工程學系	資料結構應用	文件編號：	LAB5
		發佈日期：	2022/03/28

LAB 5

以結構自訂資料型別建構鏈結串列資料結構

長榮大學
資訊工程學系

班級：資工 2B

姓名：郭智榮

學號：109B30612

日期：2022/03/28

版本校定紀錄：

版本	更新紀錄	發佈日期
0.0.0.0	初版完成	2022/03/28

資訊工程學系	資料結構應用	文件編號：	LAB5
		發佈日期：	2022/03/28

一. 實驗需求：

(一). 題目說明

描述：

使用結構(Struct)建立 Linked List 資料結構，Linked List 需儲存名字(name)及年齡(age)，資料為 20 歲的 David、24 歲的 Tom 及 30 歲的 Fred。

輸出說明：

依照題目格式依序輸出題目給予的三筆資料。

(二). 演算法

i. 演算法

定義最大人數為 3 人，以利限制人數；

定義結構資料(struct)型別 people，且成員含有一個儲存字元(char)、一個儲存整數(int)及一個儲存結構資料的型態變數 name、age 及 next，其中 name 的大小為 16；

宣告結構資料型態的變數 haed、ptr 及 newptr；

```
int main(){
```

```
    宣告題目已預先決定的名字及年齡陣列；
```

```
    迴圈 i=0 ~ i=MAX_NAME-1 (for i ; i++){
```

```
        為 newptr 要求記憶體空間，並用於創建 people 結構資料；
```

```
        將 newptr 的成員 name 設為 name[i]；
```

```
        將 newptr 的成員 age 設為 age[i]；
```

```
        將 newptr 的成員 next 設為 NULL；
```

```
        如果 i 不為 0{
```

```
            將 ptr 的成員 next 設為 newptr；
```

```
        }
```

```
        否則{
```

```
            將 head 設為 newptr；
```

```
        }
```

資訊工程學系	資料結構應用	文件編號：	LAB5
		發佈日期：	2022/03/28

```
        將 ptr 設為 newptr；  
    }
```

```
    將 ptr 設為 head；
```

```
    宣告一個整數型態變數 time 為 1，用於計算第 time 個節點；
```

```
    while(ptr 不為 NULL){  
        輸出題目要求格式；  
        將 ptr 設為 ptr 的成員 next；  
    }
```

```
    return 0；
```

```
}
```

資訊工程學系	資料結構應用	文件編號：	LAB5
		發佈日期：	2022/03/28

二. 完整程式碼：

```
#include<stdio.h>
#include<stdlib.h>
#define MAX_NAME 3 //定義最大人數為 3
struct people{
    char name[16];
    int age; //year old.
    struct people *next; //儲存下一個結構位置
} *head, *ptr, *newptr;

int main(int argc, char *argv[]){
    char name[MAX_NAME][10] = {"David", "Tom", "Fred"}; //題目輸入的名字
    int age[MAX_NAME] = {20, 24, 30}; //題目輸入的年齡

    for(int i = 0; i < MAX_NAME; i++){
        newptr = (struct people*)malloc(sizeof(struct people));
        strcpy(newptr->name, name[i]); //儲存 name[i] 到 newptr 的成員 name
        newptr->age = age[i]; //儲存 age[i] 到 newptr 的成員 age
        newptr->next = NULL; //將 newptr 的成員 next 設為 NULL

        if(i != 0){
            //若 i 不為 0 代表非首節點
            ptr->next = newptr;
            //將前一個節點的成員 next 設為新節點的位置
        }
        else{
            //若 i 為 0 代表為首節點
            head = newptr;
            //將首節點的位置存放到 head 內
        }

        ptr = newptr;
        //將 ptr 設為新節點
    }
}
```

資訊工程學系	資料結構應用	文件編號：	LAB5
		發佈日期：	2022/03/28


```

ptr = head;
//將 ptr 設為 head，代表將 ptr 設為首節點
int time = 1;
//宣告 int 變數 time，用於計算第 time 個節點
while(ptr != NULL){
    //當 ptr 不為 NULL 時進入迴圈
    printf("\n People %d ==> name : %s , age : %d \n", time++,
ptr->name, ptr->age);
    //輸出資料
    ptr = ptr->next;
    //將 ptr 設為 ptr 的下一個節點
}
return 0;
}

```

資訊工程學系	資料結構應用	文件編號：	LAB5
		發佈日期：	2022/03/28

三. 輸入及輸出結果：



```

D:\Program\C&CPP\LAB5\LAB5.exe
People 1 ==> name : David , age : 20
People 2 ==> name : Tom , age : 24
People 3 ==> name : Fred , age : 30
-----
Process exited after 0.005458 seconds with return value 0
請按任意鍵繼續 . . .

```

四. 心得與討論：

本次實作結構型態的資料，跟 Java 中的 class 相似，只是沒有在 struct 裡面寫 function，概念平常上學期在物件導向程式語言時就大概都釐清了，不過 C/C++ 與 Java 的語法差異較大，所以花了一些時間看網路資料。

在參考網路資料時，也有找到一些我個人更偏好的寫法，但另一個寫法就不太適合讓 head 跟 ptr 儲存位置，而是直接儲存下一個 struct，後來考量到如果要新增新的資料需要另外再處理會比較麻煩就沒有使用了；也因為考量到新增資料的方便性，因此又將老師的寫法改寫成只需要更新資料在陣列就能自動完成所有資料儲存，除了節省後續新增的時間，也能縮短程式碼的篇幅。