

Programming Languages Homework 6

功課習題：[課本]

1. P10-7(Fig 10.2)
2. P10-9(Fig 10.3), 畫流程圖, 並寫報告討論 Homework5 第七題有何不同。

備註：

- 1.) 未購買課本者請參閱此頁後面，將附上功課習題的掃描圖片(第六版)。

繳交期限：12/05 (四) 晚上 11:59 前

繳交格式：103360001_李奇樺.pdf

繳交內容：讀書會(包含:組員、討論時間、地點、照片或 Google Meet)、心得報告(包含: 心得、問 ChatGPT 或 Claude 三個問題、GitHub 程式連結截圖)

上傳位置：Homework\Upload

帳號、密碼：CC

如有問題請 Mail 至

崔士豪 3kkelvin@gmail.com

游亮濤 yu2000.yy@gmail.com

張富順 fushun891214@gmail.com

李育任 lyuren6432@gmail.com

P10-7(Fig 10.2)：

```

1  /* Fig. 10.2: fig10_02.c
2     Using the structure member and
3     structure pointer operators */
4  #include <stdio.h>
5
6  /* card structure definition */
7  struct card {
8     char *face; /* define pointer face */
9     char *suit; /* define pointer suit */
10 }; /* end structure card */
11
12 int main( void )
13 {
14     struct card aCard; /* define one struct card variable */
15     struct card *cardPtr; /* define a pointer to a struct card */
16
17     /* place strings into aCard */
18     aCard.face = "Ace";
19     aCard.suit = "Spades";
20
21     cardPtr = &aCard; /* assign address of aCard to cardPtr */
22
23     printf( "%s%s\n%s%s\n%s%s\n", aCard.face, " of ", aCard.suit,
24           cardPtr->face, " of ", cardPtr->suit,
25           ( *cardPtr ).face, " of ", ( *cardPtr ).suit );
26     return 0; /* indicates successful termination */
27 } /* end main */

```

Ace of Spades
 Ace of Spades
 Ace of Spades

10.7 範例：高效率的洗牌和發牌模擬器

圖 10.3 的程式是根據第 7 章所討論的洗牌和發牌的模擬來撰寫的。這個程式以一個結構陣列代表一副牌。程式使用高效率的洗牌和發牌演算法。圖 10.4 為此程式的輸出結果。

```

1  /* Fig. 10.3: fig10_03.c
2     The card shuffling and dealing program using structures */
3  #include <stdio.h>
4  #include <stdlib.h>
5  #include <time.h>
6
7  /* card structure definition */
8  struct card {
9      const char *face; /* define pointer face */
10     const char *suit; /* define pointer suit */
11 }; /* end structure card */
12
13 typedef struct card Card; /* new type name for struct card */
14
15 /* prototypes */
16 void fillDeck( Card * const wDeck, const char * wFace[],
17              const char * wSuit[] );
18 void shuffle( Card * const wDeck );
19 void deal( const Card * const wDeck );
20
21 int main( void )
22 {

```

圖 10.3 高效率的洗牌和發牌的模擬器

```

23  Card deck[ 52 ]; /* define array of Cards */
24
25  /* initialize array of pointers */
26  const char *face[] = { "Ace", "Deuce", "Three", "Four", "Five",
27      "Six", "Seven", "Eight", "Nine", "Ten",
28      "Jack", "Queen", "King"};
29
30  /* initialize array of pointers */
31  const char *suit[] = { "Hearts", "Diamonds", "Clubs", "Spades"};
32
33  srand( time( NULL ) ); /* randomize */
34
35  fillDeck( deck, face, suit ); /* load the deck with Cards */
36  shuffle( deck ); /* put Cards in random order */
37  deal( deck ); /* deal all 52 Cards */
38  return 0; /* indicates successful termination */
39 } /* end main */
40
41 /* place strings into Card structures */
42 void fillDeck( Card * const wDeck, const char * wFace[],
43     const char * wSuit[] )
44 {
45     int i; /* counter */
46
47     /* loop through wDeck */
48     for ( i = 0; i <= 51; i++ ) {
49         wDeck[ i ].face = wFace[ i % 13 ];
50         wDeck[ i ].suit = wSuit[ i / 13 ];
51     } /* end for */
52 } /* end function fillDeck */
53
54 /* shuffle cards */
55 void shuffle( Card * const wDeck )
56 {
57     int i; /* counter */
58     int j; /* variable to hold random value between 0 - 51 */
59     Card temp; /* define temporary structure for swapping Cards */
60
61     /* loop through wDeck randomly swapping Cards */
62     for ( i = 0; i <= 51; i++ ) {
63         j = rand() % 52;
64         temp = wDeck[ i ];
65         wDeck[ i ] = wDeck[ j ];
66         wDeck[ j ] = temp;
67     } /* end for */
68 } /* end function shuffle */
69
70 /* deal cards */
71 void deal( const Card * const wDeck )
72 {
73     int i; /* counter */
74

```

圖 10.3 高效率的洗牌和發牌模擬器 (續 1)

```

/* loop through wDeck */
for ( i = 0; i <= 51; i++ ) {
    printf( "%5s of %-8s%s", wDeck[ i ].face, wDeck[ i ].suit,
            ( i + 1 ) % 4 ? " " : "\n" );
} /* end for */
} /* end function deal */

```

圖 10.3 高效率的洗牌和發牌模擬器 (續 2)

Three of Hearts	Jack of Clubs	Three of Spades	Six of Diamonds
Five of Hearts	Eight of Spades	Three of Clubs	Deuce of Spades
Jack of Spades	Four of Hearts	Deuce of Hearts	Six of Clubs
Queen of Clubs	Three of Diamonds	Eight of Diamonds	King of Clubs
King of Hearts	Eight of Hearts	Queen of Hearts	Seven of Clubs
Seven of Diamonds	Nine of Spades	Five of Clubs	Eight of Clubs
Six of Hearts	Deuce of Diamonds	Five of Spades	Four of Clubs
Deuce of Clubs	Nine of Hearts	Seven of Hearts	Four of Spades
Ten of Spades	King of Diamonds	Ten of Hearts	Jack of Diamonds
Four of Diamonds	Six of Spades	Five of Diamonds	Ace of Diamonds
Ace of Clubs	Jack of Hearts	Ten of Clubs	Queen of Diamonds
Ace of Hearts	Ten of Diamonds	Nine of Clubs	King of Spades
Ace of Spades	Nine of Diamonds	Seven of Spades	Queen of Spades

圖 10.4 高效率的洗牌和發牌模擬器的輸出結果