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。

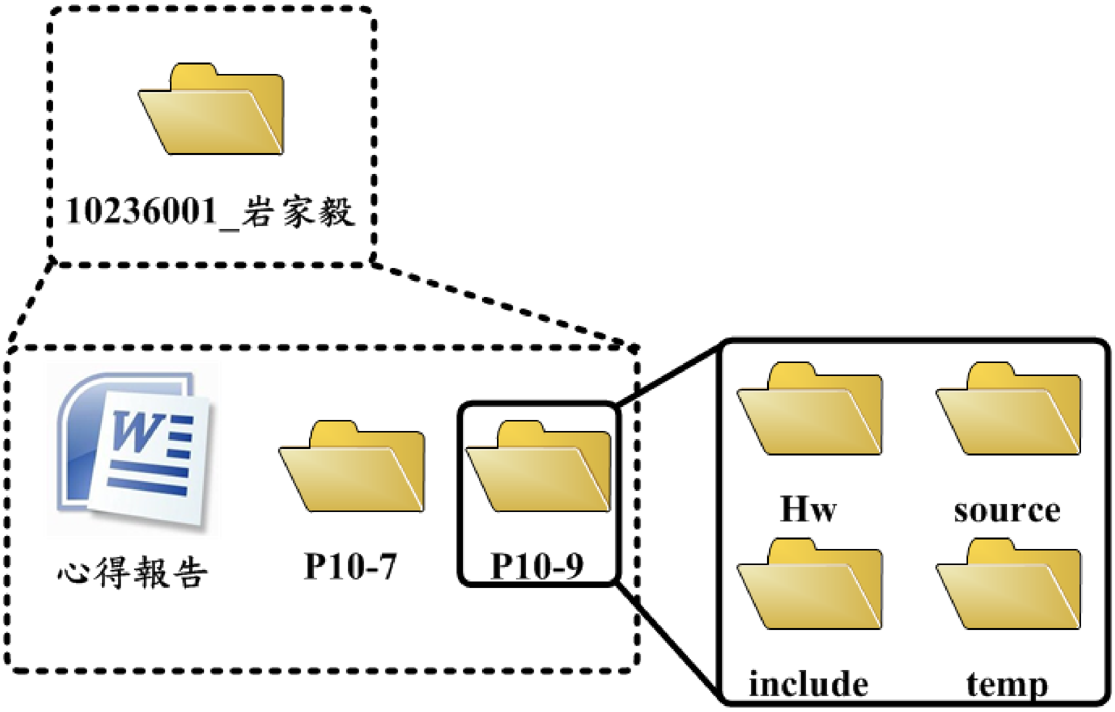
備註：

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：12/21 (四) 晚上 11:59 前

繳交格式：103360001_李奇樺.zip

繳交內容：心得報告(包含:上傳到 GitHub 的截圖)和 Lab 所檢查之程式檔案，如下圖



上傳位置：Homework\Upload

帳號、密碼：CC

如無法上傳可 Mail 繳交的作業檔案(zip)至

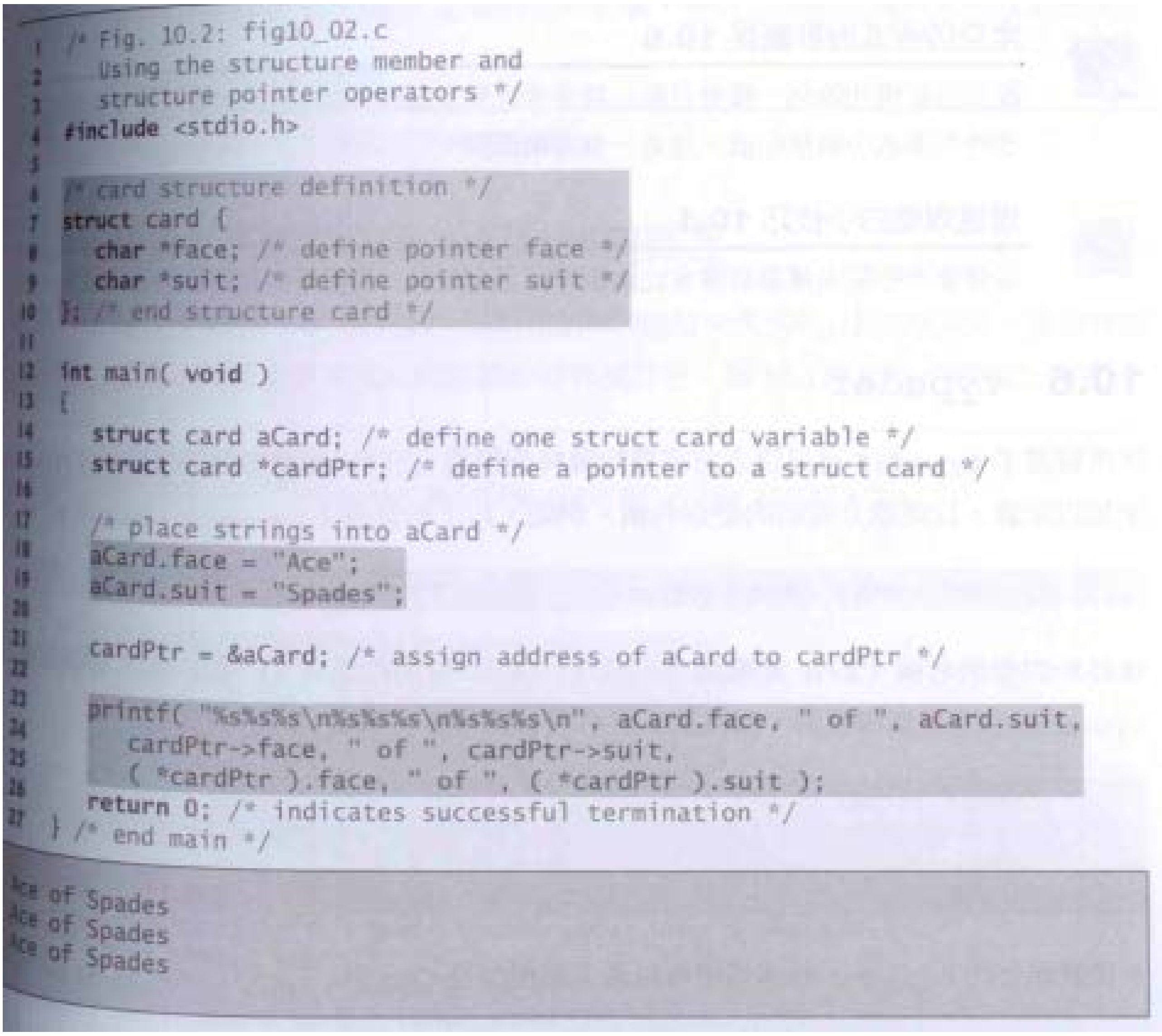
張力元 john19940815@gmail.com

林裕軒 shane.sandy123@gmail.com

何家維 jw80301@gmail.com

龔昱庭 davidkung007@gmail.com

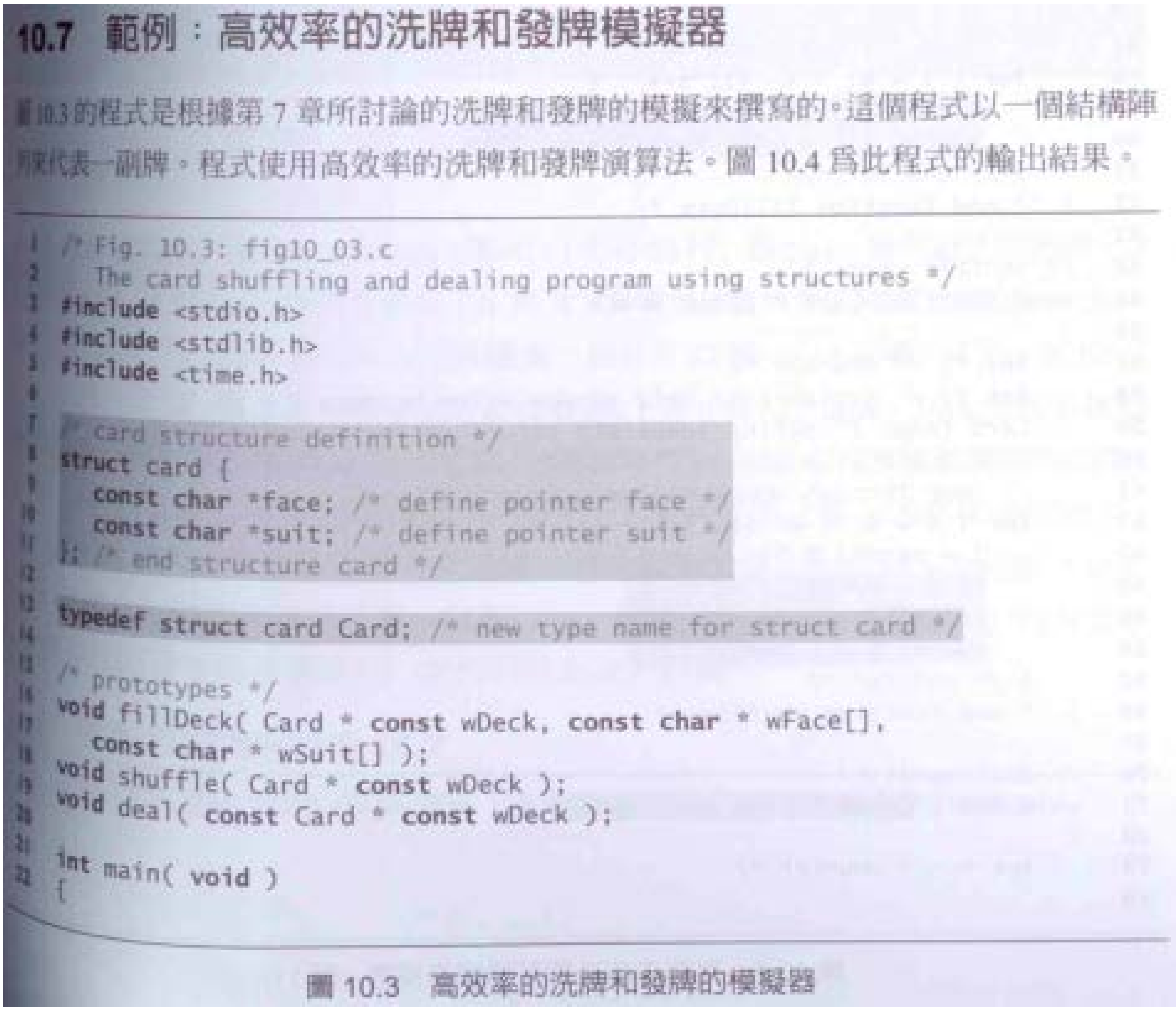
P10-7(Fig 10.2)：



```
/* Fig. 10.2: fig10_02.c
   Using the structure member and
   structure pointer operators */
#include <stdio.h>

/* card structure definition */
struct card {
   char *face; /* define pointer face */
   char *suit; /* define pointer suit */
}; /* end structure card */

int main( void )
{
   struct card aCard; /* define one struct card variable */
   struct card *cardPtr; /* define a pointer to a struct card */

   /* place strings into aCard */
   aCard.face = "Ace";
   aCard.suit = "Spades";

   cardPtr = &aCard; /* assign address of aCard to cardPtr */

   printf( "%s%s%s\n%s%s%s\n%s%s%s\n", aCard.face, " of ", aCard.suit,
      cardPtr->face, " of ", cardPtr->suit,
      ( *cardPtr ).face, " of ", ( *cardPtr ).suit );
   return 0; /* indicates successful termination */
} /* end main */
```

```
Ace of Spades
Ace of Spades
Ace of Spades
```

P10-9(Fig 10.3)：

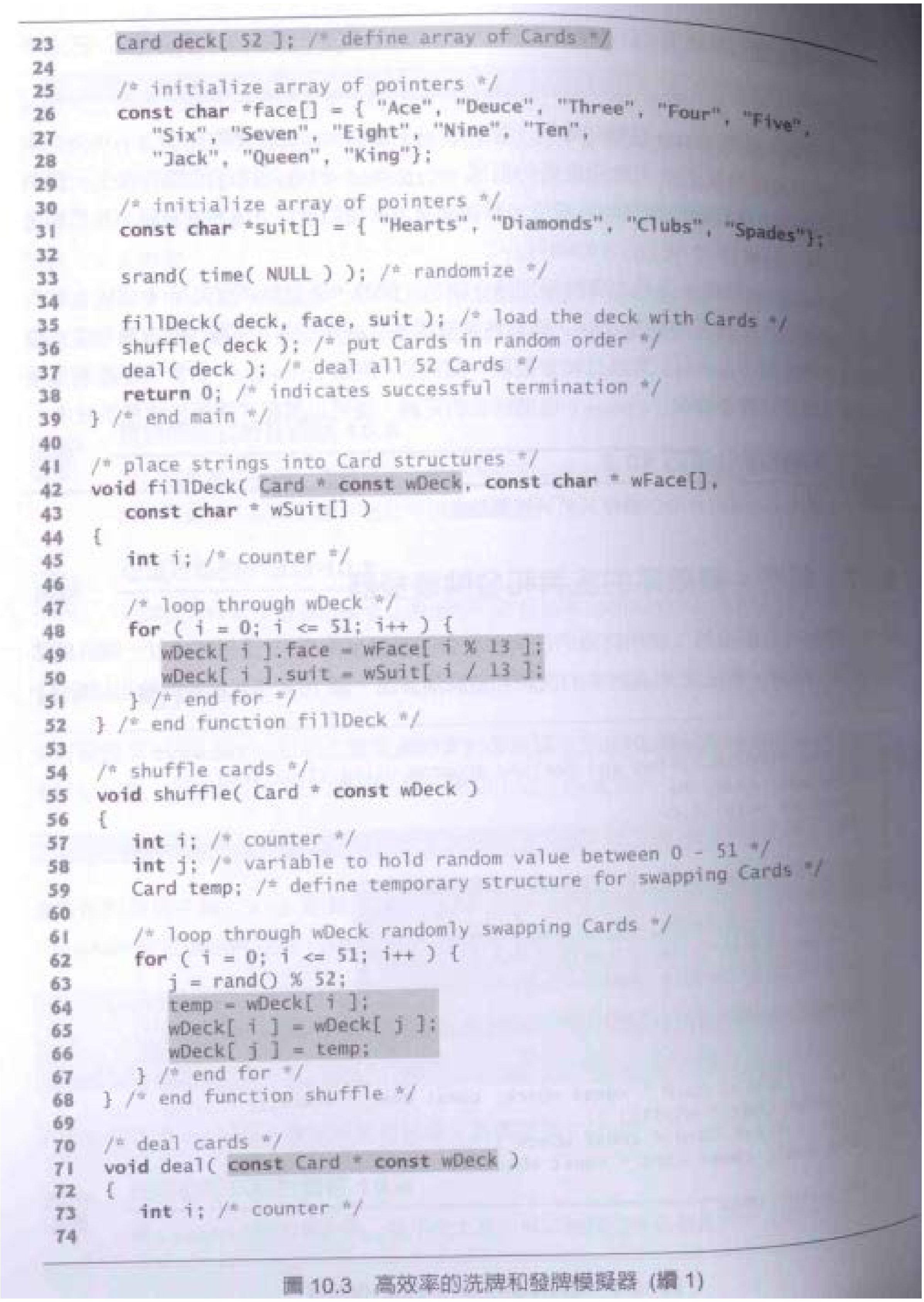


10.7 範例：高效率的洗牌和發牌模擬器

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來代表一副牌。程式使用高效率的洗牌和發牌演算法。圖 10.4 爲此程式的輸出結果。

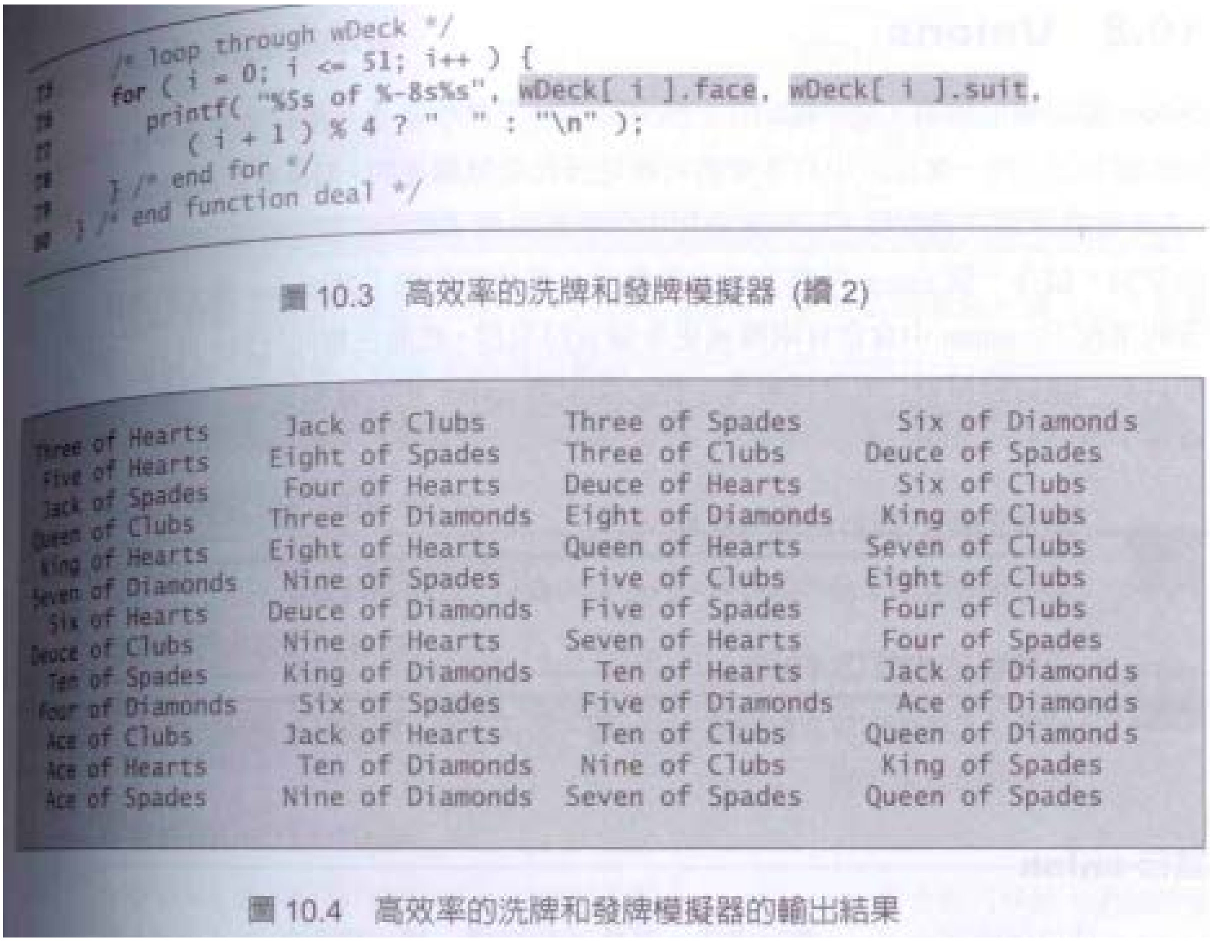
```
/* Fig. 10.3: fig10_03.c
   The card shuffling and dealing program using structures */
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

/* card structure definition */
struct card {
   const char *face; /* define pointer face */
   const char *suit; /* define pointer suit */
}; /* end structure card */

typedef struct card Card; /* new type name for struct card */

/* prototypes */
void fillDeck( Card * const wDeck, const char * wFace[],
   const char * wSuit[] );
void shuffle( Card * const wDeck );
void deal( const Card * const wDeck );

int main( void )
{
```

圖 10.3　高效率的洗牌和發牌的模擬器



```
   Card deck[ 52 ]; /* define array of Cards */

   /* initialize array of pointers */
   const char *face[] = { "Ace", "Deuce", "Three", "Four", "Five",
      "Six", "Seven", "Eight", "Nine", "Ten",
      "Jack", "Queen", "King"};

   /* initialize array of pointers */
   const char *suit[] = { "Hearts", "Diamonds", "Clubs", "Spades"};

   srand( time( NULL ) ); /* randomize */

   fillDeck( deck, face, suit ); /* load the deck with Cards */
   shuffle( deck ); /* put Cards in random order */
   deal( deck ); /* deal all 52 Cards */
   return 0; /* indicates successful termination */
} /* end main */

/* place strings into Card structures */
void fillDeck( Card * const wDeck, const char * wFace[],
   const char * wSuit[] )
{
   int i; /* counter */

   /* loop through wDeck */
   for ( i = 0; i <= 51; i++ ) {
      wDeck[ i ].face = wFace[ i % 13 ];
      wDeck[ i ].suit = wSuit[ i / 13 ];
   } /* end for */
} /* end function fillDeck */

/* shuffle cards */
void shuffle( Card * const wDeck )
{
   int i; /* counter */
   int j; /* variable to hold random value between 0 - 51 */
   Card temp; /* define temporary structure for swapping Cards */

   /* loop through wDeck randomly swapping Cards */
   for ( i = 0; i <= 51; i++ ) {
      j = rand() % 52;
      temp = wDeck[ i ];
      wDeck[ i ] = wDeck[ j ];
      wDeck[ j ] = temp;
   } /* end for */
} /* end function shuffle */

/* deal cards */
void deal( const Card * const wDeck )
{
   int i; /* counter */

```

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