P10-7

#include<stdio.h>

#include<stdlib.h>

struct card

{

char \*face;

char \*suit;

};

int main(void)

{

struct card aCard;

struct card \*cardPtr;

aCard.face = "Ace";

aCard.suit = "Spades";

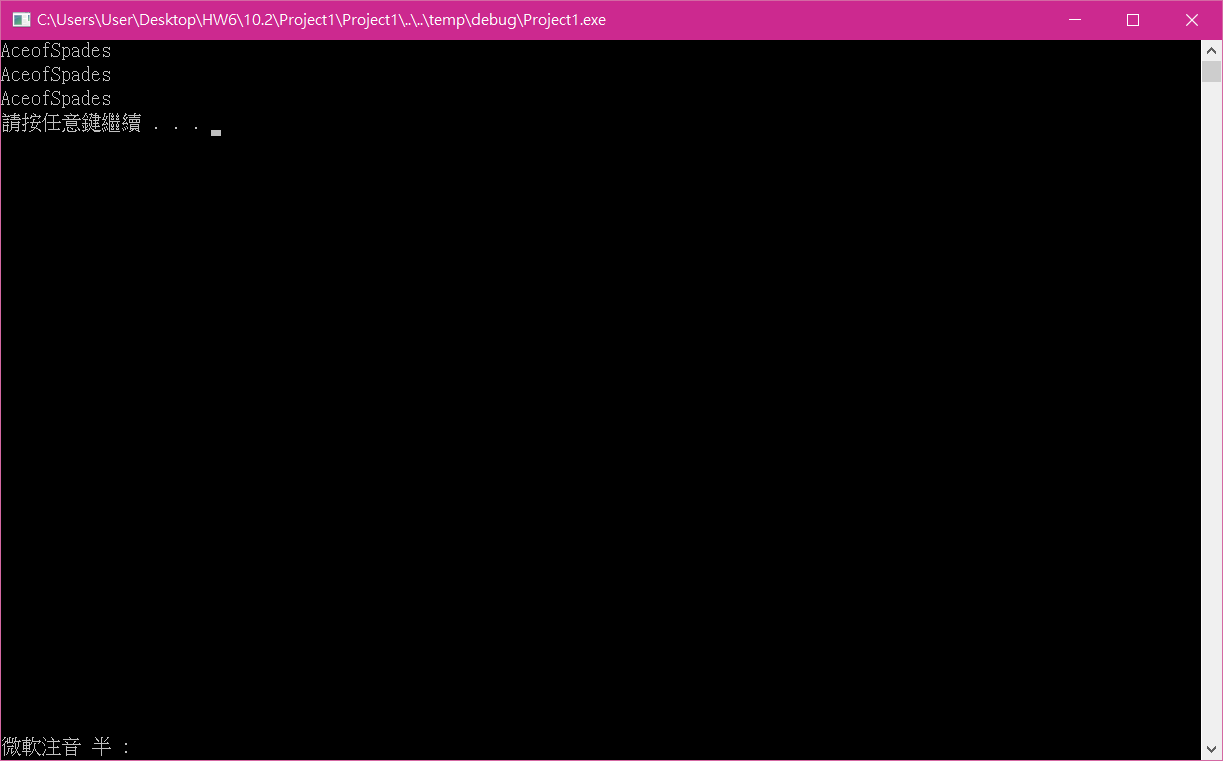
cardPtr = &aCard;

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);

system("pause");

return 0;

}



P10-9

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#define CARDS 52

#define FACES 13

struct card

{

const char \*face;

const char \*suit;

};

typedef struct card Card;

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)

{

Card deck[CARDS];

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

const char \*suit[] = { "Hearts","Diamonds","Clubs","Spades" };

srand( time ( NULL ) );

fillDeck(deck, face, suit);

shuffle(deck);

deal(deck);

system("pause");

return 0;

}

void fillDeck(Card \* const wDeck, const char \* wFace[], const char \* wSuit[])

{

size\_t i;

for ( i = 0; i < CARDS; ++i)

{

wDeck[i].face = wFace[i%FACES];

wDeck[i].suit = wSuit[i/FACES];

}

}

void shuffle(Card \* const wDeck)

{

size\_t i;

size\_t j;

Card temp;

for ( i = 0; i < CARDS; ++i)

{

j = rand() % CARDS;

temp = wDeck[ i ];

wDeck[i] = wDeck[j];

wDeck[j] = temp;

}

}

void deal(const Card \* const wDeck)

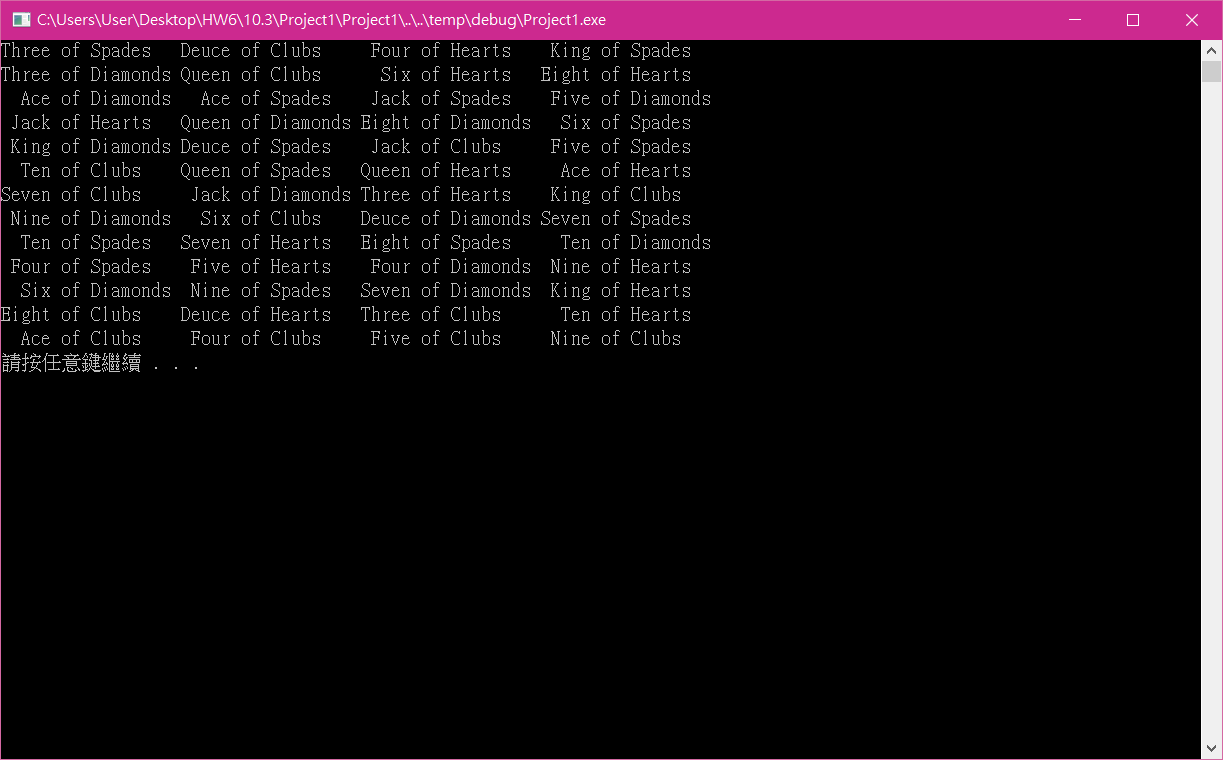
{

size\_t i;

for (i = 0; i < CARDS; ++i)

printf("%5s of %-8s%s", wDeck[i].face, wDeck[i].suit, (i + 1) % 4 ? " " : "\n");

}



這次使用結構來簡化原本複雜的迴圈。

Git

心得

這次我對程式的增進效率以及改良有很大的幫助