Exercise 1A

왼쪽의 코드는 swap()함수를 실행했을 때 함수 내에서는 값 자체를 서로 바꾸어서 함수 내에서 출력할 때에는 변수 I, j가 바뀌어서 나온다. 하지만 swap함수가 끝나고 main으로 돌아올 때에는 main 내의 변수 I, j와 swap내의 I, j는 서로 영향을 주지 못하므로 값이 바뀌지 않은 형태로 출력된다.

오른쪽의 코드는 swap()함수에서 포인터를 통해 주소를 참조한다. 주소를 인자로 제공하여 pi의 포인터 변수에 pj가 가리키는 값을 저장하고 pj의 주소에 pi가 가리켰던 값을 저장한다. 따라서 swap 함수가 끝난 이후에도 모두 I, j의 주소는 바뀌지 않고 주소가 가리키는 값을 바꾸었기 때문에 main함수에서 두 값이 바뀌어진 형태로 출력된다.

Exercise 1B

포인터 변수 p는 i의 주소를 저장하고 있다. 출력 서식문자 %p는 변수의 주소를 나타낸다. 첫번째 출력은 i의 주소와 p에 저장된 값을 출력한다. P에는 i의 주소가 저장되어 있기때문에 두 출력값은 서로 같다. 두번 째 출력은 i에 저장된 값과 p가 참조하는 값을 출력한다. I에 저장된 값은 100이고 p(i의 주소)가 가리키는 값은 마찬가지로 100이다.
그리고 p에 i의 주소값을 저장하고 i의 값은 1로 저장했다. 따라서 세번째 출력값은 i의 값인 1이고 *P는 i의 주소가 참조하는 값이므로 마찬가지로 1이다.
그리고 *P 값을 2로 저장한 것은 i의 주소가 참조한 값을 2로 바꾼 것과 같으므로 다섯번째 출력값은 2이고 여섯번째 출력값도 2이다.

Exercise 1C

Ptr1은 arr[1]의 주소를 가지고 있기 때문에 (A)는 *(ptr+3)이다.
Ptr2는 arr[0]의 주소를 가지고 있기 때문에 (B)는 *(ptr+4)이다.

Exercise 1D

fillDeck 함수는 deck이라는 구조체 배열에 각각 rank(숫자)와 suit(모양)을 저장시킨다.
Shuffle 함수는 rand( )를 이용해 0~52의 숫자를 랜덤으로 생성한다. 0부터 51번까지 차례대로 해당하는 인덱스의 배열 원소와 랜덤숫자의 인덱스인 배열의 원소를 바꿔 저장하면서 shuffle의 기능을 이용한다.

Main() 함수에서는 while()문에서 사용자에게 몇 개의 카드를 뽑을 지 입력을 받고 shuffle되어 있는 deck의 0번째 원소부터 입력받은 뽑을 카드 개수만큼 출력하여 알려준다. 사용자의 입력이 제한되어 있는 카드의 개수를 초과할 때 프로그램은 종료된다.

Exercis 1E

Data Structure는 데이터를 저장하는 방법에 관한 것이다.
Algorithm은 다양한 data structure의 연산과 동시에 그것들을 실행시키기 위한 명령 집합들이다.
따라서 프로그램은 Data Structure + Algorithm으로 구성되며, 이번 자료구조 수업은 Data Structure 좀 더 포커스하여 공부할 것이다.

#define NAME number
NAME이라는 이름을 가진 변수를 number로 지정함으로써 그 변수가 빈번하게 사용이 될 때 Define을 통해 불편함을 해소할 수 있다.

함수의 인자로써 배열을 전할 때에는 function(array) 하면 된다. Array를 그냥 주면 배열의 주소를 준다.

&(Ampersand) operator
Reference operator
Return the address of a variable