|  |  |  |  |
| --- | --- | --- | --- |
| 2022/2 『자료구조』실습 보고서 | | | |
| 제목 | 3장 실습(O ) 과제( ) | 제출일자 | 2022.   09.  22 . |
| 학번 | 201911608 | 이름 | 김지환 |

|  |
| --- |
| **실습 3장** |
| 1. [강의 TP] Call by Value, call by reference 에제  #include <stdio.h>  void copy\_array(int a[], int b[], int len) {  for (int i = 0; i < len;i++) b[i] = a[i];  }  void copy\_variable(int a, int b) {  b = a;  }  int main(void) {  int A[5] = { 10, 20, 30 };  int B[5], i, x = 2018, y = 0;  copy\_variable(x, y);  copy\_array(A, B, 5);  printf("변수 복사 결과: x=%d, y=%d\n", x, y);  printf("배열 복사 결과: \n");  for (i = 0; i < 5;i++) {  printf("A[%d] = %d\t", i, A[i]);  printf("B[%d] = %d\n", i, B[i]);  }  return 0;  }  실행결과    ->  변수는 call by value 이다. 함수 내 파라미터 a와 b에 2018과 0이 들어왔다.  b는 2018로 변경했지만 함수 내에 존재하는 변수이므로 함수의 연산이 끝나도  기존 x, y 변수의 값은 그대로이다.  배열은 값을 저장 할 때 A[0]의 메모리 주소부터 A[n]까지 +n의 메모리 주소를 갖는다. 그래서 배열은 메모리 주소를 전달하므로 함수 내에서 전달받은 메모리 주소에 직접적으로 연산을 수행한다. 그래서 배열 복사 결과에서 A의 배열 값대로 B의 배열 값이 변경되었다. |
| 2. [강의 TP] 구조체 매개변수 전달 예제  #include <stdio.h>  typedef struct {  double real;  double imag;  } Complex;  void print\_complex(Complex c) {  printf("%4.1f + %4.1f;\n", c.real, c.imag);  }  void reset\_complex(Complex c) {  c.real = c.imag = 0.0;  }  int main(void) {  Complex a = { 1.0, 2.0 };  printf("초기화 이전 : ");  print\_complex(a);  reset\_complex(a);  printf("초기화 이후 : ");  print\_complex(a);  return 0;  }  실행결과    -> 구조체에서는 reset\_complex 함수를 실행하기 전과 실행하기 후의 값이 같음을 보아 call by value임을 알 수 있다.  함수 내에서 새로운 Complex 구조체를 정의하면서 새로운 메모리를 사용하므로 함수 내 Complex c의 객체 값만 변경된다. 함수를 벗어난 complex a의 값은 변동이 없다. |
| 3. 프로그램 3.1  #include <stdio.h>  typedef struct studentTag {  char name[10];  int age;  double gpa;  } student;  int main(void) {  student a = { "kim", 20, 4.3 };  student b = { "park", 21, 4.2 };  printf("student a - 이름 : %s, 나이: %d, 학점: %.1f\n", a.name, a.age, a.gpa);  printf("student a - 이름 : %s, 나이: %d, 학점: %.1f\n", b.name, b.age, b.gpa);  return 0;  }  실행결과 |
| 4. 프로그램 3.1 Quiz (75쪽)를 구현하여 실행  01. 2차원 좌표 공간에서 하나의 점을 나타내는 구조체 Point를 정의하여 보라. typedef도 사용하여서 구조체 Point를 하나의 타입으로 정의한다.  02. 01에서 정의한 구조체에서 변수인 p1과 p2를 정의하여 보라.  03. p1과 p2를 각각(1,2)와 (9,8)로 초기화하라.  04. 점을 나타내는 두개의 구조체 변수를 받아서 점 사이의 거리를 계산하는 함수 get\_distance(Point p1, Point p2)를 작성하여 보자.  #include <stdio.h>  #include <math.h>  typedef struct {  int x;  int y;  } Point;  double get\_distance(Point p1, Point p2) {  int dx = pow(p1.x - p2.x, 2);  int dy = pow(p1.y - p2.y, 2);  double distance = sqrt((double)dx + dy);  return distance;  }  int main(void) {  Point p1, p2;  p1.x = 1; p1.y = 2;  p2.x = 9, p2.y = 8;  printf("%f", get\_distance(p1, p2));  return 0;  }  실행결과 |