Day02 (20 01 16) \_ 자료구조

함수(function)

-재사용이 가능

- 관리가 쉽다

- 큰 프로그램을 작성할 때 공동작업이 유리

- 가독성을 높일 수 있다

반환형 함수이름 (매개변수 / parameter / 인자)

{

}

1. 전달인자(o) , 반환값(o)
2. 전달인자(o) , 반환값(x)
3. 전달인자(x) , 반환값(o)
4. 전달인자(x) , 반환값(x)

|  |
| --- |
| 1. int Add1(int n1, int n2) { 2. return n1 + n2; 3. } 4. void Add2(int n1, int n2) { 5. int num; 6. num = n1 + n2; 7. printf("%d\n", num); 8. } 9. int Add3() { 10. int n1 = 1, n2 = 2; 11. return n1+n2; 12. } 13. void Add4() { 14. printf("%d\n", 3); 15. } |

Int main(){ // caller

}

Void MasxFunc(){ // callee

}

지역변수 – 메인 함수에 int I 가 있는데 함수에서 int i를 또 쓸 수 있나요? 넹!

재귀함수

재귀 (recursion)

-원래의 자리로 되돌아가거나 되돌아옴. 반복, 되풀이

-재귀함수(recursive function)

-함수 내에서 자기 자신을 다시 호출하는 함수

=============================

사용자가 입력한 수까지 1부터 더하기

5입력 -> 1+2+3+4+5..

n입력->1+2+3+4..(n-1)+n

n + (n-1) + (n-2)…+2+1

|  |
| --- |
| Int main() {  int n1;  printf("정수 입력 : ");  scanf("%d", &n1);  printf("%d\n", Recursive(n));  return 0;  }  int Recursive(int n) { //1~사용자가 입력한 합까지 출력    if (n == 1) {  return 1;  }  else {  return n + Recursive(n - 1);  }      } |