#8주차 정리노트(하혜정, 노진경)

1. 함수 사용하기- 함수 주소

#include <stdio.h>

void print\_hello()

{

printf("Hello, world!n");

}

int main()

{

//메모리상에 저장된 함수의 주소값

printf("함수의 주솟값 : %p\n,", print\_hello);

return 0;

}

예제 6-3

함수호출 연습

조건 : 중복된 for문(같은 역할을 하는) 함수로 뺌으로써 코드를 짠다. 함수의 이름을 짓고, (SumXXX..)

>>함수의 유용성

1차 시도)

#include <stdio.h>

int SumF(int a, int b);

int main(void)

{

int SumF(int a, int b);

{

int sum = 0, i;

for(i= a; i<= b; i++)

sum +=i;

printf( "1부터 10까지 합은 %d다.\n, SumF(1, 10)");

printf("7부터 17까지 합은 %d다.\n, SumF(7,17)");

return sum;

}

}#include <stdio.h>

int SumF(int a, int b);

int main(void)

{

int SumF(int a, int b);

{

int sum = 0, i;

for(i= a; i<= b; i++)

sum +=i;

printf( "1부터 10까지 합은 %d다.\n, SumF(1, 10)");

printf("7부터 17까지 합은 %d다.\n, SumF(7,17)");

return sum;

}

A : 왜 이 코드에서 오류가 난 걸까요?

B : 음.. SumF(1, 10)부분의 함수 호출 방식이 다른 것 같아요.

A : 그러면 예제에 있는 것처럼 함수 호출과 반환한 값을 변수에 저장한 뒤에 출력으로 수정해 볼까요?

B : 일단 처음 것을 저장해 놓고 수정해서 실행해 봐요.

A : 그러고 다시 문제가 생기면 그때 다시 수정해 볼까요?

A : 그런데, 수정하면서 보니까 printf문에서 %d가 빠진 것 같아요. 이것도 수정해서 한번 다시 실행해 보겠습니다.

수정된 코드)

#include <stdio.h>

int SumF(int a, int b);

int main(void) {

int result1 = SumF(1, 10);

int result2 = SumF(7, 17);

printf("1부터 10까지 합은 %d다.\n", result1);

printf("7부터 17까지 합은 %d다.\n", result2);

return 0;

}

int SumF(int a, int b) {

int sum = 0, i;

for (i = a; i <= b; i++)

sum += i;

return sum;

}

A : 중복된 반복문에서 함수를 사용하면 유용해짐을 느꼈다.

B : 동감한다. 특히 반복문의 반복, 그러니까 반복문의 중첩이 일어날 때, 반복적인 노동을 줄일 수 있어서 좋았고, 함수로 대체하면서 코드내에서의 함수의 쓰임새와 주석을 이용하면 코드의 가독성 또한 올라갈 수 있을 것 같다.

1. 수행

조건)

정수열 배열 b[]={20, 34,12,24,54,91,9,40,81, 10}의 최댓값, 쵯소값을 출력하는 코드를 짜라. 함수 선언을 이용하라.

#include <stdio.h>  
  
int fMax(int arr[], int size) {  
int max = arr[0];  
for (int i = 1; i < size; i++) {  
if (arr[i] > max) {  
max = arr[i];  
}  
}  
return max;  
}  
  
int fMin(int arr[], int size) {  
int min = arr[0];  
for (int i = 1; i < size; i++) {  
if (arr[i] < min) {  
min = arr[i];  
}  
}  
return min;  
}  
  
int main() {  
int b[] = {20, 34, 12, 24, 54, 91, 9, 40, 81, 10};  
int size = sizeof(b) / sizeof(b[0]);  
  
int max\_value = fMax(b, size);  
int min\_value = fMin(b, size);  
  
printf("최댓값: %d\n", max\_value);  
printf("최솟값: %d\n", min\_value);  
  
return 0;  
}