|  |
| --- |
| 2020 1학기 |
| 프로그래밍 언어 레포트 |
| 8주차 학습활동 |

|  |
| --- |
| 송 지민  사회학과 201721899 |

**[과제#1] 실습#3, #4 실습수행 및 실행화면 캡처**

1. **실습#3**

#include <stdio.h>

#include <stdlib.h>

void get\_dice\_face();

int main(void)

{

int i;

for(i = 0; i < 100; i++)

{

get\_dice\_face();

}

return 0;

}

void get\_dice\_face()

{

static int check = 0;

static int one = 0;

static int two = 0;

static int three = 0;

static int four = 0;

static int five = 0;

static int six = 0;

int dice\_face = (rand()%6) + 1;

switch(dice\_face){

case 1 :

one++;

break;

case 2 :

two++;

break;

case 3 :

three++;

break;

case 4 :

four++;

break;

case 5 :

five++;

break;

case 6 :

six++;

break;

}

check++;

if(check==100)

{

printf("1 -> %d\n", one);

printf("2 -> %d\n", two);

printf("3 -> %d\n", three);

printf("4 -> %d\n", four);

printf("5 -> %d\n", five);

printf("6 -> %d\n", six);

}

}



1. **실습#4**

#include <stdio.h>

#include <stdlib.h>

int check()

{

static int call\_count = 0;

call\_count++;

if(call\_count>3){

printf("로그인 시도횟수 초과\n");

return 0;

}

printf("비밀번호: ");

int password;

scanf("%d", &password);

if( password == 1234 )

{

printf("성공적으로 로그인 되었습니다.\n");

return 1;

}

return 0;

}

int main(void)

{

check();

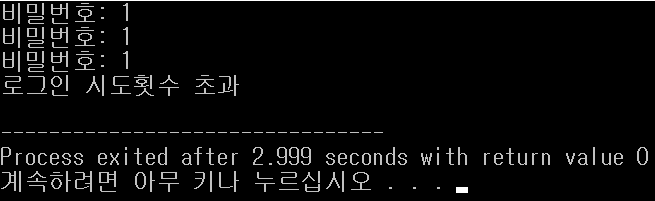
check();

check();

check();

return 0;

}



* **전역변수와 함수를 이용하여 보완 프로그램 작성**

#include <stdio.h>

void get\_password();

int password;

int main(void)

{

int call\_count;

for(call\_count = 0; call\_count < 4; call\_count++)

{

if(call\_count == 3)

{

printf("로그인 시도횟수를 초과하였습니다.");

break;

}

get\_password();

if(password == 1234)

{

printf("비밀번호가 일치합니다. 로그인 되었습니다.\n");

break;

}

else

printf("비밀번호가 일치하지 않습니다.\n\n");

}

return 0;

}

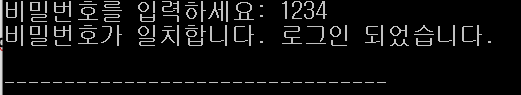
void get\_password()

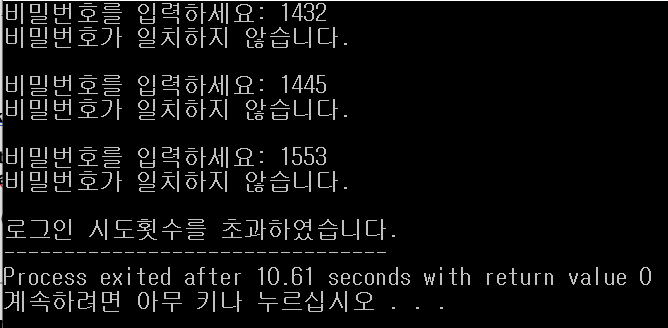
{

printf("비밀번호를 입력하세요: ");

scanf("%d", &password);

}





**[과제#2] 순환기법을 이용한 계산 프로그램 작성**

// 순환기법을 이용한 계산 프로그램

#include <stdio.h> // 표준입출력 헤더파일

int sum(int num); // 함수 원형, 1부터 N까지의 합을 계산해줄 함수

/\* 반환형이 정수이며, 매개변수로 정수형 변수 num을 가짐

main()함수보다 뒤에 나오기 때문에 미리 원형을 정의해주어야 한다.

함수 원형은 함수 헤더에 세미콜론(;)을 추가한 것과 같으나, 함수 원형에서는 매개변수의 이름은 적지 않아도 된다.\*/

int main() // 메인 함수의 선언부, 반환유형은 정수형이며 매개변수는 없음

{

// 변수 선언 및 초기화

int number = 0; // 사용자로부터 입력받은 정수를 저장할 변수

int result = 0; // 1부터 number까지의 합을 저장할 변수

// 정수 입력받기

printf("정수를 입력하시오: "); // 정수 입력 안내하기

scanf("%d", &number); // 정수 입력받기, 주소연산자 이용해 number에 저장

result = sum(number); // 합계 계산하는 함수 호출하여 입력값이 number 일 때의 반환값을 result에 저장

/\* 함수를 호출할 때는 반환형을 쓰지 않고 이름과 인수(매개변수가 있을 때)를 쓰는데여기서 int sum(int num)의 경우 매개변수가 정수형이므로 인수 또한 정수형이 와야 한다. 또한 반환형이 정수형이므로 sum(number)의 결과로 정수형이 반환되며 그 값이 정수형 변수 result에 저장된다.\*/

printf("1부터 %d까지의 합 = %d\n", number, result); // 1부터 number까지의 합 result를 출력

return 0; // 메인 함수에 0을 반환하여 끝내기

}

int sum(int num) // 1부터 num(매개변수)까지의 합을 계산할 함수, 반환유형과 매개변수는 정수형

{

if(num != 0) // num이 0이 아닐 때

{

return num + sum(num - 1); // sum(3) = 3 + sum(2) 이므로 sum(num) = num + sum(n-1)

}

else

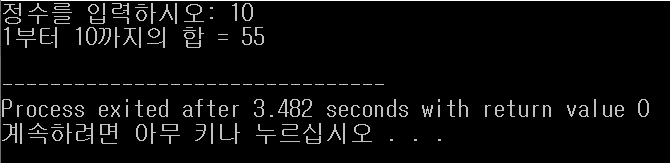
return num; // num가 0이면 더할 것이 없으므로 num 반환

/\* 자기 자신을 호출하는 것을 순환(recursion)이라고 부르며, 순환은 함수가 자기 자신을 호출하여 문제를 해결하는 프로그래밍 기법이다.

순환 알고리즘은 자기 자신을 순환적으로 호출하는 부분과 순환 호출을 멈추는 부분으로 되어있다.

위의 코드에서 else문(num == 0)이 순환 호출을 멈추는 부분이다.\*/

}



**[과제#3] 정수 자리수 계산 프로그램 작성**

// 정수 자리수 계산 프로그램

#include <stdio.h> // 표준입출력 헤더파일

int get\_no\_digits(int n); // 정수의 자리수를 계산해줄 함수 원형 정의

/\* 메인함수에서 호출되지만 메인함수 뒤에 있기 때문에 함수 원형을 정의해주어야 한다.

원형을 정의할 때는 반환형, 함수이름은 필수이지만 매개변수 이름은 생략해도 된다.\*/

int main() // 메인함수 선언부, 정수형을 반환하며 매개변수가 없음

{

// 변수 선언

int n, ctr; // 사용자로부터 입력받은 정수를 저장할 변수 n, 자리수 저장할 변수 ctr

// 사용자로부터 정수 입력받기

printf("정수를 입력하시오: "); // 정수 입력 안내하기

scanf("%d", &n); // 정수 입력받아서 n에 저장하기, 정수이기 때문에 변환명세 %d 사용

// 자료 처리

ctr = get\_no\_digits(n); // 자리수 구하는 함수 호출하여 입력값 n을 주고 반환값을 ctr에 저장

/\* 함수 안의 문장들은 호출되기 전까지는 실행되지 않는다.

함수를 호출하게 되면 현재 실행하고 있는 코드는 잠시 중단되고, 호출된 함수로 이동하여 함수 몸체 안의 문장들이 순차적으로 실행된다.\*/

// 결과 출력하기

printf("자리수의 개수: %d\n", ctr); // ctr변수에 저장한 자리수를 출력

return 0; // 0반환하여 끝내기

}

int get\_no\_digits(int n) // 정수 자리수를 계산할 함수, 반환형과 매개변수 모두 정수로 존재함

{

static int digit = 1; // 정적 할당을 사용하여 자리수가 누적되도록 함, 자리수는 최소1이므로 초기값은 1

/\* 지역변수는 함수 또는 블록 안에서 정의되는 변수로 해당 블록이나 함수 안에서만 사용이 가능하다.

정적 지역변수라는 것은 지역변수에 정적 할당을 이용하여 프로그램이 실행되는 동안에는 계속하여 변수에 저장공간이 할당되게 한 것이다.

정적 할당을 이용하지 않으면 일반적으로 블록이 시작되며 변수에 저장 공간이 할당되고 블록이 종료되면 저장공간이 회수되어 값이 누적되지 않는다.

지역 변수를 정적 변수로 만들기 위해서는 변수의 자료형 앞에 static을 붙이면 된다.\*/

if(n/10 != 0) // 정수를 10으로 나눈 몫이 0이 아니라면

{

get\_no\_digits(n/10); // 10진수의 자리수를 구하기 위해서는 그 수가 한 자리 수가 될 때 까지 10으로 계속 나눠줘야 함

digit++; // 한 번 나눌 때마다 1을 추가

}

else // n을 10으로 나눴을 때 몫이 0이라는 것은 n이 한 자리 수라는 것

return digit; // digit을 반환

}

/\* 함수 이름 또한 변수명과 마찬가지로 식별자에 따른 규칙만 따른다면 어떤 이름이라도 가능하지만 소스코드를 읽기 쉽게 하기 위해 함수의 기능을 암시하는 이름을 부여하는 것이 좋다.

C에서는 함수의 길이에 아무런 제한을 두지 않지만 함수의 길이가 지나치게 길어진다면, 하나 이상의 작업을 하고 있다고 봐야 한다.

절대적인 기준은 없으나 30행을 넘지 않도록 하는 게 좋다.\*/

