아주대학교 프로그래밍언어

201821048 미디어학과 이서영

2019/11/06 재귀함수, static 사용

# 소스코드 및 주석

/\*

[실습#3] 주사위 각 면 횟수 출력 프로그램

요구사항 : 주사위를 던져서 각각의 면이 몇 번 나왔는지를 출력하는 프로그램을

작성하시오. 주사위를 던지는 함수 get\_dice\_face()함수를 만들고

이 함수 안에서 각각의 면이 나올 때마다 그 횟수를 정적 지역변수를

이용하여 저장한다. get\_dice\_face()함수 호출회수가 100이 되면

각 면의 횟수를 출력한다

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

100번을 돌리는 것 / n을 입력받아 n번 돌리는 것으로 둘 다 작성함.

100번 :

1. for문으로 주사위함수(100)을 100번 실행한다

2. 123456을 랜덤하게 생성해서 맞으면 각 숫자의 개수를 올려 저장한다

3. 100번 굴렸으면 종료하고 각 숫자의 개수를 출력한다.

n번 :

1. for문으로 주사위함수(n)을 n번 실행한다

2. 123456을 랜덤하게 생성해서 맞으면 각 숫자의 개수를 올려 저장한다

3. n번 굴렸으면 종료하고 각 숫자의 개수를 출력한다.

핵심 내용 :

static으로 변수의 값이 맞은 횟수를 카운트 하기

static으로 각 함수의 불린 횟수 카운트 하기

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void get\_dice\_face(int count);

void get\_dice\_face\_100();

int main(void)

{

srand(time(NULL));

int i, n;

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

get\_dice\_face\_100();

/\*

위에는 100번, 밑에는 입력 받은 만큼

\*/

printf("\n주사위를 몇 번 굴릴 것인가?");

scanf\_s("%d", &n);

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

get\_dice\_face(n);

return 0;

}

//100번 입력 함수

void get\_dice\_face\_100()

{

int number = (rand() % 6) + 1;

static int one = 0, two = 0, three = 0, four = 0, five = 0, six = 0;

static call\_count = 0;

call\_count++;

switch (number) {

case 1:

one++;

break;

case 2:

two++;

break;

case 3:

three++;

break;

case 4:

four++;

break;

case 5:

five++;

break;

case 6:

six++;

break;

}

if (call\_count == 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);

}

}

//n번 입력 함수

void get\_dice\_face(int count)

{

int number = (rand() % 6) + 1;

static int one = 0, two = 0, three = 0, four = 0, five = 0, six = 0;

static call\_count = 0;

call\_count++;

switch (number) {

case 1:

one++;

break;

case 2:

two++;

break;

case 3:

three++;

break;

case 4:

four++;

break;

case 5:

five++;

break;

case 6:

six++;

break;

}

if (call\_count == count) {

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);

}

}

/\*

[실습#4] 로그인 시에 비밀번호 검사 프로그램

로그인 시에 비밀번호를 검사하는 함수 int check()를 작성해서

테스트 하시오. check()함수가 한번 호출될 때마다 비밀번호를

질문하고 일치 여부를 0과 1로 반환한다. 비밀번호는 숫자 1234로

고정되어 있다고 가정한다. check()가 3번 이상 호출되고 비밀번호가

일치하지 않으면 check() 는 “로그인 시도 횟수 초과” 메시지를

출력한다. check()함수 안에 정적변수를 선언하여 사용해보자

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 비밀번호 입력

2. 맞는지 확인

3-1. 맞으면 그 후 부를 시, 이미 맞음 출력

//맞으면 바로 종료하도록 발전

3-2. 아니면 그 후 불러서 확인 시,

4번째로 함수를 부르면 로그인시도횟수초과메시지 출력

핵심 판단 :

함수 부른 숫자 3보다 큼 -> 로그인 시도횟수 초과 출력, 입력 안받음

맞아서 더이상 로그인 필요 없음 -> (이미 비밀번호 맞춤 출력,) 입력 안받음

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

#include <stdio.h>

#include <stdlib.h>

int check()

{

static int call\_count = 0;

call\_count++;

int static truth = 0, password = 1234, input\_password = 0;

/\*

if (truth == 1) {

printf("이미 비밀번호를 맞추었습니다.\n");

}

//처음에 맞추고 다시 시도하면 이미 맞추었다고 출력 안내

//하지만 맞추면 바로 종료하는 내용으로 발전시킴

\*/

if (call\_count <= 3 && truth == 0) {

printf("비밀번호 : ");

scanf\_s("%d", &input\_password);

if (input\_password == password) {

printf("로그인 되었음\n");

truth++;

exit(1);

//조교님이 맞으면 바로 종료하도록 발전시킬 수 있다고 하심.

}

}

else if (call\_count > 3) {

printf("로그인 시도횟수 초과");

}

return truth;

}

int main(void)

{

check();

check();

check();

check();

return 0;

}

/\*

[실습#5] 순환기법을 이용한 계산 프로그램

요구사항 : 1부터 N까지의 합(1+2+3+…+N)을 계산하는 문제를

순환기법을 이용해서 작성해보자.

힌트 : sum(3)은 3+sum(2)로 바꿀 수 있다

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 메인함수에서 값을 입력한다

2. 더해주는 함수에서 -1씩 하면서 재귀적으로 호출하며 더한 값을 리턴한다

3. 점점 줄어들며 1이 되면 위의 함수로 올라가며 최종 값을 반환한다.

핵심 판단 :

n에서 1씩 빼주면서 재귀함수 사용, 값을 더해준다.

\*/

#include <stdio.h>

int sum(int n);

int main()

{

int number, result;

printf("정수를 입력하시오: ");

scanf\_s("%d", &number);

result = sum(number);

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

}

int sum(int num)

{

if (num == 1) {

return 1;

}

else {

return num + sum(num - 1);

}

}

/\*

[실습#6] 정수 자리수 계산 프로그램

요구사항 : 주어진 정수가 몇 개의 자리수를 가지고 있는지를 계산하는 프로그램을

순환을 이용하여 작성해보자. 예를 들어서 12345의 경우에는

5가 출력된다

힌트 : 순환 호출 시에 어떤 데이터를 저장하려면 정적 지역변수를

사용할 수 있다.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 메인에서 숫자를 자리수 함수로 넘긴다.

2. 자리수 함수에서 10으로 나누어지면 static으로 저장한 값++ 해준다.

3. 자리수 함수에서 계속 10으로 나누다가 0이면 1을 반환한다.

핵심 내용 : 계속 10으로 나누어 자리수를 줄이고 자리수를 더해 저장한다

한 자리 수를 10으로 나누어 몫이 0이 되면 지금까지 더한 자리수를 반환한다.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

#include<stdio.h>

int get\_no\_digits(int n);

int main()

{

int n, ctr;

printf("정수를 입력하시오: ");

scanf\_s("%d", &n);

ctr = get\_no\_digits(n);

printf("자리수의 개수: %d\n", ctr);

return 0;

}

int get\_no\_digits(int n)

{

int static digit = 0;

if (n == 0) {

return digit;

}

else {

digit++;

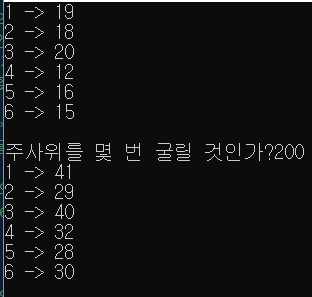
return get\_no\_digits(n / 10);

}

}

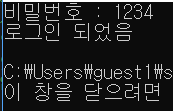
# 실행파일

No3

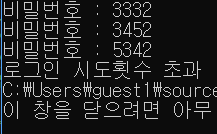


No4

1. 비밀번호를 맞춤 -> 바로 종료



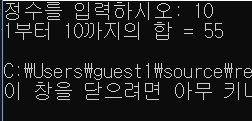
1. 비밀번호를 3번 틀림 -> 4번째 시도에서 오류 메시지 발생



1. 비밀번호 2번 틀리고 3번째에 맞춤 -> 맞음 표시 후 종료

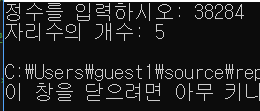


No5

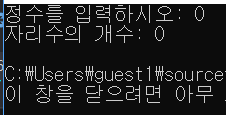


No6

1. 자연수 입력



1. 0 입력



1. 음수 입력

