**C프로그래밍1\_챕터9**

**1분반**

**학번: 2011136024**

**성명: 김재환**

1. **챕터 9\_문제 2번**

주사위를 던져서 각각의 면이 몇 번 나왔는지를 출력하는 프로그램을 작성하라. 주사위의 면의 난수를 이용하여 생성한다. 주사위를 던지는 함수 get\_dice\_face()를 만들고 이 함수 안에서 각각의 면이 나올 때마다 그 횟수를 정적 지역 변수를 이용하여 기억하게 하라. get\_dice\_face() 호출 횟수가 100의 배수일 때마다 면이 나온 횟수를 출력한다.

Ex) 14 9 18 20 20 19.

34 21 33 37 34 41

1. **챕터 9\_문제 4번**

다음과 같은 무한 수열을 계산하는 순환적인 프로그램을 작성하라.

1/1+1/2+1/3…..1/n

1. **챕터 9\_문제 5번**

은행 계좌에서 저축하고 인출하는 프로그램을 작성하여 보자. Save(int mount)함수는 저금할 금액 amount를 받으며 save(100)과 같이 호출된다. Draw(int amount)함수는 예금한 금액의 인출을 나타낸다. 사용자에게 메뉴를 보여주고 저축 또는 인출을 선택하게 한다.

Ex) 메뉴를 선택하세요 : 저축(1), 인출(2) : 1

저축할 금액: 5000

현재 잔액은 10000입니다.

**2-2) 소스코드**

**#include <stdio.h> //표준입출력함수를 이용하기 위한 헤더**

**#include <stdlib.h> // 표준라이브러리함수를 이용하기 위한 헤더**

**#include <time.h> // srand()함수를 이용하기 위한 헤더**

**void get\_dice\_face(void); // 주사위를 던져서 각각의 숫자가 몇 번 나왔는지를 출력하는 함수**

**void main(void)**

**{**

**get\_dice\_face();// 각각의 숫자가 몇 번 나왔는지를 count하는 함수 호출**

**}**

**void get\_dice\_face(void) // 주사위를 던져서 각각의 숫자가 몇 번 나왔는지를 출력하는 함수**

**{**

**static int count1 = 0; //숫자1이 나온 횟수를 세는 static 변수**

**static int count2 = 0; //숫자2이 나온 횟수를 세는 static 변수**

**static int count3 = 0; //숫자3이 나온 횟수를 세는 static 변수**

**static int count4 = 0; //숫자4이 나온 횟수를 세는 static 변수**

**static int count5 = 0; //숫자5이 나온 횟수를 세는 static 변수**

**static int count6 = 0; //숫자6이 나온 횟수를 세는 static 변수**

**int dice\_number; // 주사위 숫자를 저장하는 변수**

**int sum\_result; // 합을 통한 100의 배수가 되는 count들의 합을 저장하는 변수**

**int num, i; // 반복문의 횟수와 반복문을 위한 변수**

**srand((unsigned)time(NULL));// 난수 발생기가 난수를 발생 할 때 마다 다른 수를 출력**

**printf("횟수를 입력하세요 ");**

**scanf\_s("%d", &num); // 반복문이 반복되는 횟수**

**for (i = 0; i < num; i++)**

**{**

**dice\_number = rand() % 6 + 1;// 1 ~ 6사이의 숫자 중 난수 발생**

**switch (dice\_number)**

**{**

**case 1: // 난수가 1일 경우 1의 카운트를 한번 더한다.**

**count1 += 1;**

**break;**

**case 2: // 난수가 2일 경우 1의 카운트를 한번 더한다.**

**count2 += 1;**

**break;**

**case 3: // 난수가 3일 경우 1의 카운트를 한번 더한다.**

**count3 += 1;**

**break;**

**case 4: // 난수가 4일 경우 1의 카운트를 한번 더한다.**

**count4 += 1;**

**break;**

**case 5: // 난수가 5일 경우 1의 카운트를 한번 더한다.**

**count5 += 1;**

**break;**

**case 6: // 난수가 6일 경우 1의 카운트를 한번 더한다.**

**count6 += 1;**

**break;**

**}**

**sum\_result = count1 + count2 + count3 + count4 + count5 + count6;// count가 100번 되었을 때마다 출력**

**if (sum\_result % 100 == 0)// 난수의 count의 합이 100의 배수 일 때**

**printf("%d %d %d %d %d %d\n", count1, count2, count3, count4, count5, count6);// 1~6까지의 숫자가 각각 몇 번씩 나타났는지 출력**

**}**

**}**

**2-4) 소스코드**

**#include <stdio.h> // 표준 입출력 함수 사용을 위한 헤더**

**void main(void)**

**{**

**double denominator = 1.0; // 분모가 되는 변수**

**double numerator; // 분자의 범위가 되는 변수**

**double sum = 0 , numerator\_input; // 합과 분자의 가장 큰 값이 되는 변수**

**printf("실수를 입력하세요 ");**

**scanf\_s("%lf",&numerator\_input);// 분자의 범위 입력**

**for(numerator = 1.0; numerator<=numerator\_input; numerator++) // 분자는 1부터 numerator\_input까지 1씩 증가**

**{**

**sum +=denominator/numerator; // 분수의 합**

**}**

**printf("%lf\n", sum); // 최종적인 값을 출력**

**}**

**2-5) 소스코드**

**#include <stdio.h> // 표준 입출력 함수를 위한 헤더**

**#include <Windows.h> // 유틸리티 함수를 사용하기 위한 헤더**

**int save(int amount);// 입력 받은 돈을 저축하기 위한 함수**

**int draw(int amount);// 입력 받은 돈을 인출하기 위한 함수**

**int balance = 5000;// 통장의 돈의 인출과 저축을 위한 전역 변수**

**void main(void)**

**{**

**int menu\_select; // 저축을 할 것 인지 인출을 할 것 인지 프로그램을 종료할 것 인지를 입력 받기 위한 변수**

**int save\_money; // 저축할 금액의 돈을 입력 받기 위한 변수**

**int use\_money; // 인출할 금액의 돈을 입력 받기 위한 변수**

**while(1) // 3을 입력 받을 경우 반복문을 빠져나가고 1~3사이의 숫자가 아닌 다른 수를 입력 받았을 경우 반복문을 다시 수행**

**{**

**printf("메뉴를 선택하세요: (1) 저축, (2) 인출, (3) 프로그램종료 ");**

**scanf\_s("%d", &menu\_select); // 메뉴 입력**

**if(menu\_select == 1) // 1을 입력 받았을 경우 돈을 저축**

**{**

**printf("저축할돈을 입력하세요 ");**

**scanf\_s("%d", &save\_money); // 저축하고자 하는 돈의 금액 입력**

**save(save\_money); // 입력 받은 돈의 금액만큼 저축**

**printf("현재 잔액은 %d입니다\n", balance); // 저축 후 현재 금액 출력**

**}**

**else if(menu\_select == 2) //2을 입력 받았을 경우 돈을 인출**

**{**

**printf("인출 할 돈을 입력하세요 ");**

**scanf\_s("%d", &use\_money); // 인출할 금액 입력**

**draw(use\_money); // 입력 받은 돈의 금액만큼 인출**

**printf("현재 잔액은 %d입니다\n", balance); // 인출 후 현재 금액 출력**

**}**

**else if(menu\_select == 3) //2을 입력 받았을 경우 돈을 인출**

**{**

**printf("현재 잔액은 %d입니다\n", balance); // 현재 금액 출력**

**break; // 반복문을 종료한다.**

**}**

**else**

**{**

**printf("잘못된 입력입니다 \n");// 1~3사이의 숫자 입력이 아닐 시 잘못된 입력이라는 경고 문구 출력**

**system("pause");//화면을 지우기 이전에 잠시 멈추는 기능 수행**

**system("cls");// 화면을 지우는 기능 수행**

**continue;// 반복문을 계속한다.**

**}**

**system("pause");//화면을 지우기 이전에 잠시 멈추는 기능 수행**

**system("cls");// 화면을 지우는 기능 수행**

**}**

**}**

**int save(int amount)// 입력 받은 돈을 저축하기 위한 함수**

**{**

**balance +=amount;// amount만큼 금액에 더한다.**

**return balance;// 더한 값을 반환**

**}**

**int draw(int amount)// 입력 받은 돈을 인출하기 위한 함수**

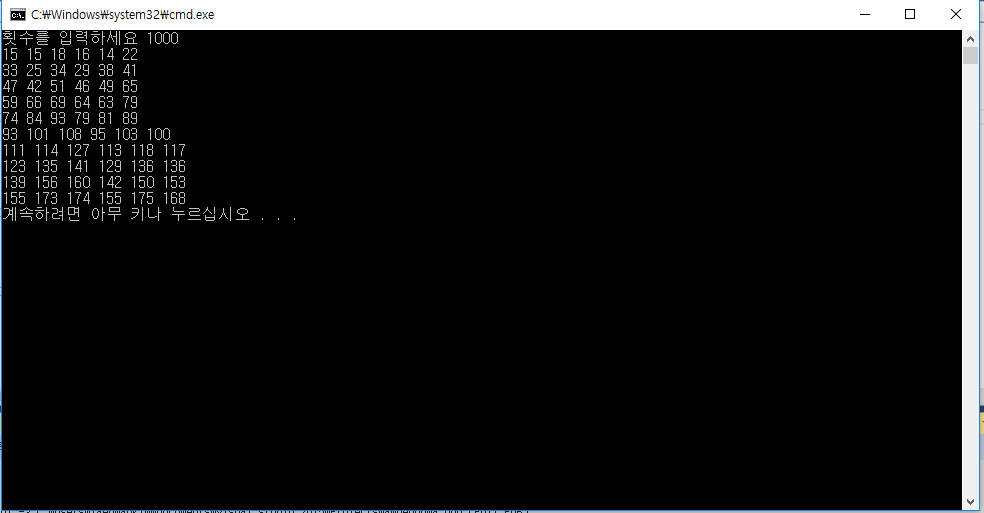
**{**

**balance -=amount;// amount만큼 금액을 빼준다.**

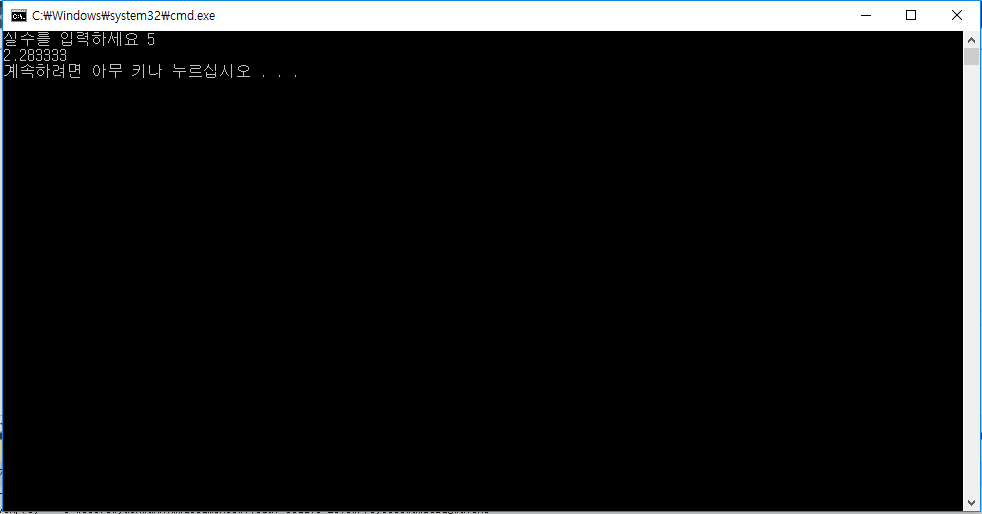
**return balance;// 뺀 값을 반환**

**}**

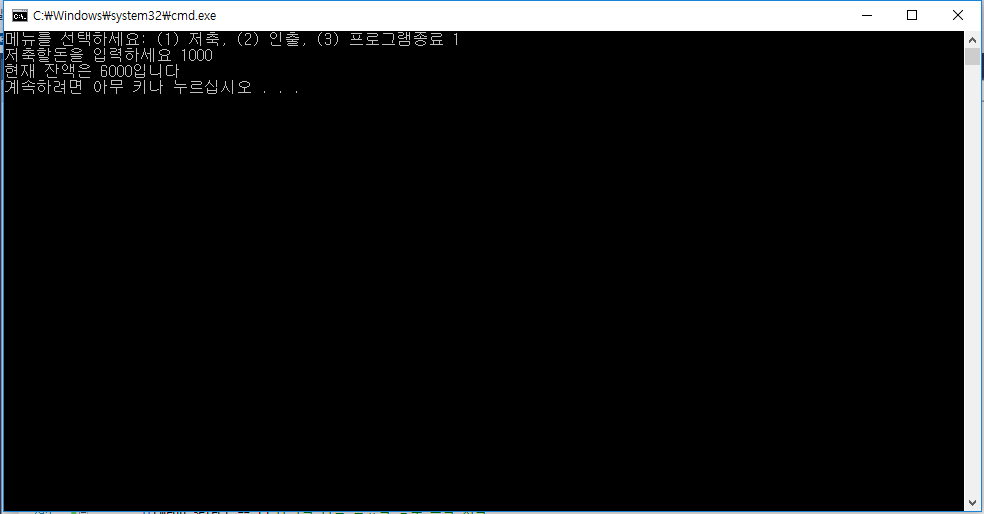
**3-2) 실행결과**



**3-4) 실행결과**



**3-5) 실행결과**



**4-2) 알고리즘**

get\_dice\_face()함수를 이용하여, main에서 함수 호출을 하였습니다. get\_dice\_face()함수에서는 1~6까지의 수의 카운트 세기 위한 count 1~6변수가 선언 되었으며 1~6까지의 난수를 저장하기 위한dice\_number, 100번의 횟수 마다 출력을 위한 count 1~6까지의 합 sum\_result변수, 그리고 총 몇 회 진행을 할 것 인지에 대한 입력을 받기 위한 변수num을 선언 하였습니다. 이는 반복 문을 통해 num번 반복이 되게 되며 rand() % 6 + 1을 이용하여 1~6까지의 난수를 발생 시킵니다. 이후 switch문을 이용하여 난수가 1일경우 count1을 한번 증가시키고, 이처럼 count6까지의 count를 증가 시킵니다. 그리고 이러한 count1~6까지의 변수의 합이 100의 배수를 위한 (%100==0)의 조건문을 이용하였습니다.

**4-4) 알고리즘**

분모가 되는 변수(denominator)을 1.0으로 두고 분자 되는 변수(numerator)는 1.0부터 입력 받는 변수까지(numerator\_input)까지 1씩 증가시키고, 이를 sum이라는 변수가 계속해서 더해집니다.

**4-5) 알고리즘**

처음의 돈의 금액을 저장하는 변수는 전역 변수로 선언이 되고, 이는 5000으로 초기화됩니다. 이후 저축할 돈과 인출할 돈, 그리고 수행할 작업의 번호를 선택하는 변수를 입력 받습니다. 이를 조건문을 이용하여 1이 선택되었을 경우 돈을 저축하게 되고 2를 입력 받았을 경우 돈을 인출하게 되고 3을 입력 받았을 경우, 프로그램 종료, 1~3사이의 숫자가 아닌 엉뚱한 숫자를 입력 받았을 경우 숫자를 다시 입력 받도록 합니다 돈을 저축하는 함수는 save함수가 기능을 수행하며 인출하는 기능은 draw함수가 그 기능을 수행하게 됩니다.