**C프로그래밍1\_챕터8**

**1분반**

**학번: 2011136024**

**성명: 김재환**

1. **챕터 8\_문제 6번**

난수(random number)는 컴퓨터를 이용한 문제 해결에서 많이 사용된다, 특히 수학적인 분석이 너무 복잡한 경우에 시뮬레이션을 사용하면 실제로 제품을 제작하지 않고서도 많은 실험을 할 수 있다. 비주얼 C++의 경우, rand()가 한번 호출될 때마다 0에사 32767까지의 정수를 같은 확률로 선택하여 반환한다. Rand()함수를 이용하여 0 또는 1값을 무작위로 반환하는 함수 b\_rand()를 작성하고 5번 호출하여 보자.

Ex) 1 1 0 0 1.

비겼음

1. **챕터 8\_문제 7번**

앞에서 작성한 b\_rand()함수를 이용하여 간단한 동전 던지기 게임을 시뮬레이션하여 보자. 컴퓨터가 동전을 던지고 사용자는 앞 뒤를 말한다. 컴퓨터는 b\_rand()를 이용하여 생성된 난수가 1이면 동전의 앞면으로 간주하고 0이면 동전의 뒷면으로 간주한다.

Ex) 앞면 또는 뒷면(1 또는 0): 1

맞았습니다.

계속하시겠습니까? (y 또는 n): n

1. **챕터 8\_문제 8번**

0.0 부터 1.0까지의 난수를 반환하는 함수 f\_rand()를 작성하고 5번 호출하여 본다.

Ex) 0.001251 0.563585 0.193304 0.808741 0.585009

1. **챕터 8\_문제 9번**

화면에 세로로 막대 그래프를 그리는 프로그램을 작성하여 보자. 인수의 값만큼의 별표 문자를 출력하는 함수 print\_value(int n)을 작성하라. 반복적으로 사용자로부터 값을 입력 받아서 print\_value()를 호출하여 입력 값만큼 막대를 그리는 프로그램을 완성하라. 사용자가 음수를 입력하면 반복을 중단하라.

Ex) 값을 입력하세요(종료는 음수 입니다): 5

\*\*\*\*\*

값을 입력하세요(종료는 음수 입니다): -1

1. **챕터 8\_문제 10번**

두개의 정수 n, m을 입력 받아서 n이 m의 배수이면 1을 반환하고 그렇지 않으면 0을 반환하는 함수 is\_multiple (int n, int m)를 작성하고 테스트하여 보자

Ex) 첫 번째 정수를 입력 하시오 : 30

두 번째 정수를 입력 하시오 : 5

30은 5의 배수 입니다.

1. **챕터 8\_문제 11번**

두 점 사이의 거리를 계산하는 함수를 작성하여 보자. 2차원 공간에서 두 점 (x1, y1)과(x2, y2) 사이의 거리를 계산하는 get\_distance(double x1, double y1, double, x2. double y2)를 작성하라. 다음과 같은 두 점 사이의 거리를 계산하는 공식을 사용하라, 제곱근은 sqrt()라이브러리 함수를 사용하라.

Ex) 첫 번째 점의 좌표를 입력하세요 (x, y): 1 1

두 번째 점의 좌표를 입력하세요 (x, y): 10 10

두점 사이의 거리는 12.727922입니다.

1. **챕터 8\_문제 12번**

주어진 정수가 소수인지를 검사하는 함수 is\_prime()을 작성하라. 이 함수를 이용하여 2부터 100 사이의 모든 소수를 출력하라.

Ex) 2 부터 100사이의 소수를 출력합니다.

2, 3 …..

**2-6) 소스코드**

**#include <stdio.h>//표준 입출력함수를 사용하기 위해 선언**

**#include <stdlib.h>// 표준 라이브러리 함수를 사용하기 위해 선언**

**#include <time.h> // srand()함수를 하용하기위해 선언**

**void b\_rand(void); //0 또는 1을 출력하는 함수**

**void main(void)**

**{**

**b\_rand();// 함수 호출**

**}**

**void b\_rand(void)**

**{**

**int random\_number;// 0 또는 1을 랜덤하게 대입하는 변수**

**int i; // 반복문을 사용을 위한 변수**

**srand((unsigned)time(NULL));//출력 할 때마다 변화된 숫자를 출력하기 위한 함수**

**for (i = 0; i < 5; i++)// 반복문 5번 수행**

**{**

**random\_number = rand() % 2;// 0 또는 1만 숫자 생성**

**printf("%d ", random\_number); //0또는 1 출력**

**}**

**printf("\n");// 개행**

**}**

**2-7) 소스코드**

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

**#include <stdlib.h>// 표준 라이브러리 함수 사용을 위한 헤더**

**#include <time.h>// srand()함수 사용을 위한 헤더**

**#include <Windows.h>// 유틸리티 함수 사용을 위한 헤더**

**int b\_rand(void);//0 또는 1을 출력하는 함수**

**void main(void)**

**{**

**int user\_choice;// 사용자의 0또는 1 선택을 위한 변수**

**char choice\_continue;// 프로그램을 계속할것인지 여부 판단을 위한 변수**

**while(1)**

**{**

**printf("앞면 또는 뒷면(1 또는 0): ");**

**scanf\_s("%d", &user\_choice);// 사용자의 선택을 입력받는 변수**

**while (getchar() != '\n');// scanf를 사용함에 있어 버퍼로 인안 오류를 해결하기 위해 버퍼를 비워준다.**

**if (user\_choice == b\_rand())// 사용자의 선택과 랜덤에 의한 숫자가 같은 경우**

**printf("컴퓨터의 동전은 %d로 사용자의 선택과 맞았습니다\n", b\_rand());**

**else// 사용자의 선택과 랜덤에 의한 숫자가 다른 경우**

**printf("컴퓨터의 동전은 %d로 사용자의 선택과 틀렸습니다\n", b\_rand());**

**printf("계속하시겠습니까? (y 또는 n): ");**

**//scanf\_s("%c", &choice\_continue);**

**choice\_continue = getchar();// 프로그램을 계속할것인지에 대한 여부 판단.**

**system("cls");// cmd창의 화면에 출력되어 있는것을 지워준다.**

**if (choice\_continue == 'n')// 계속할것인지의 여부에서 n을 선택했을시 프로그램을 종료한다.**

**break;**

**}**

**}**

**int b\_rand(void)//0 또는 1을 출력하는 함수**

**{**

**int n; //랜덤함수를 통한 변수 저장**

**srand((unsigned)time(NULL)); // 출력을 할때마다 다른 숫자로 바꿔주는 기능 수행**

**n = rand() % 2; // 0또는 1을 랜덤하게 n변수에 저장**

**return n;// 0 또는 1 반환**

**}**

**2-8) 소스코드**

**#include <stdio.h>// 표준입출력함수 사용을 위한 헤더**

**#include <stdlib.h>// 표준 라이브러리함수 사용을 위한 헤더**

**#include <time.h>// srand()함수 사용을 위한 헤더**

**double f\_rand(void);//0.0 에서 1.0사이의 숫자를 랜덤하게 출력하기 위한 함수**

**void main(void)**

**{**

**int i;// 반복문 사용을 위한 변수**

**srand((unsigned)time(NULL));// 프로그램 실행시 마다 다른 숫자를 반화하기 위해 사용**

**for (i = 0; i < 5; i++) // 5번의 함수호출을 위한 반복문 사용**

**printf("%lf \n", f\_rand());**

**}**

**double f\_rand(void)//0.0 에서 1.0사이의 숫자를 랜덤하게 출력하기 위한 함수**

**{**

**double rand\_result; //램덤함수의 값을 저장하기 위한 변수**

**rand\_result = rand() / (double)RAND\_MAX;// 저장되는 값이 0.0에서 1.0사이 숫자가 될수 있게 RAND\_MAX로 나눈다**

**return rand\_result;// 결과값 반화**

**}**

**2-9) 소스코드**

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

**#include <Windows.h>// 유틸리티 함수 사용을 위한 헤더**

**void Print\_value(int n);// 사용자에게 입력받은 수만큼 별문자 출력을 위한 함수**

**void main(void)**

**{**

**int star\_number;// 별을 몇 번 출력 할것인지 입력받기 위한 변수**

**while (1)// 무한 반복문 사용**

**{**

**printf("값을 입력하세요 (종료는 음수 입니다): ");**

**scanf\_s("%d", &star\_number);// 별의 숫자 입력**

**if (star\_number < 0)// 음수를 입력받았을경우 반복문 종료**

**break;**

**else// 음수가 아닌 양수를 입력받았을경우 입력받은 숫자만큼 별문자 출력**

**Print\_value(star\_number);**

**system("pause");// cmd창 일시적인 정지**

**system("cls");// cmd창을 지우는 기능 수행**

**}**

**}**

**void Print\_value(int n)// 사용자에게 입력받은 수만큼 별문자 출력을 위한 함수**

**{**

**int i;//반복문 사용을 위한 변수**

**for (i = 0; i < n; i++)// 입력받은 수만큼 반복문 수행**

**{**

**printf("\*");// 별문자 출력**

**}**

**printf("\n");// 개행**

**}**

**2-10) 소스코드**

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

**int is\_multiple(int n, int m); // 배수인지의 여부를 판단하는 함수**

**void main(void)**

**{**

**int number1, number2;// 변수를 입력받기 위한 변수**

**printf("첫 번째 정수를 입력 하세요 ");**

**scanf\_s("%d", &number1);// 첫 번째 변수 입력**

**printf("두 번째 정수를 입력 하세요 ");**

**scanf\_s("%d", &number2);// 두 번째 변수 입력**

**if (is\_multiple(number1, number2) == 1)// 두 수가 배수 일경우**

**printf("%d는 %d의 배수 입니다 \n", number1, number2);**

**else// 두 수가 배수가 아닐경우**

**printf("%d는 %d의 배수가 아닙니다 \n", number1, number2);**

**}**

**int is\_multiple(int n, int m)// 배수인지의 여부를 판단하는 함수**

**{**

**return (n % m == 0) ? 1 : 0; // 입력받은 변수가 배수일 경우 1반환 아닐 경우 0 반환**

**}**

**2-11) 소스코드**

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

**#include <math.h>// 수학 라이브러리 사용을 위한 함수**

**double get\_distance(double x1, double y1, double x2, double y2);//두 좌표사이의 거리를 구하는 함수**

**void main(void)**

**{**

**double x1, y1, x2, y2;// 좌표값을 입력받기 위한 변수**

**printf("첫 번째 좌표를 입력하세요 ");**

**scanf\_s("%lf %lf", &x1, &y1);// 첫 번째 좌표 입력**

**printf("두 번째 좌표를 입력하세요 ");**

**scanf\_s("%lf %lf", &x2, &y2);// 두 번째 좌표 입력**

**printf("두 좌표 사이의 거리는 %lf 입니다\n", get\_distance(x1,y1,x2,y2));// 반환된 거리 값 출력**

**}**

**double get\_distance(double x1, double y1, double x2, double y2)//두 좌표사이의 거리를 구하는 함수**

**{**

**return sqrt(pow((x1 - x2), 2.0) + pow((y1 - y2), 2.0));// 두좌표사이의 거리를 구하는 공식을 통한 값 반환**

**}**

**2-12) 소스코드**

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

**int is\_prime(int n);// 소수를 구하는 함수**

**void main(void)**

**{**

**int number, i;// 소수를 구하기 위한 변수 number, 반복문 수행을 위한 변수 i**

**printf("소수를 구할 정수를 입력하세요 ");**

**scanf\_s("%d", &number);// 소수를 구하기 휘한 변수 입력**

**for (i = 2; i <= number; i++)// 반복문 수행**

**{**

**if(is\_prime(i) == 1)// 반환되는 수가 1일경우 소수이다**

**printf(" %d ", i);// i값 출력**

**}**

**printf("\n");// 개행**

**}**

**int is\_prime(int n)// 소수를 구하는 함수**

**{**

**int i;// 반복문 사용을 위한 변수**

**for (i = 2; i < n; i++)//반복문을 통한 소수 판별**

**{**

**if ( n % i == 0)// 2 에서 n-1까지의 수 중 나눠서 나머지가 0인 수가 있을경우 소수가 성립이 되지 않아 반복문을 빠져나간다.**

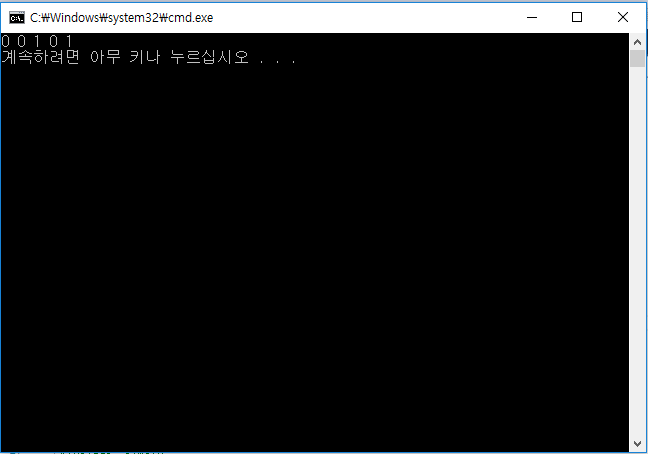
**break;**

**}**

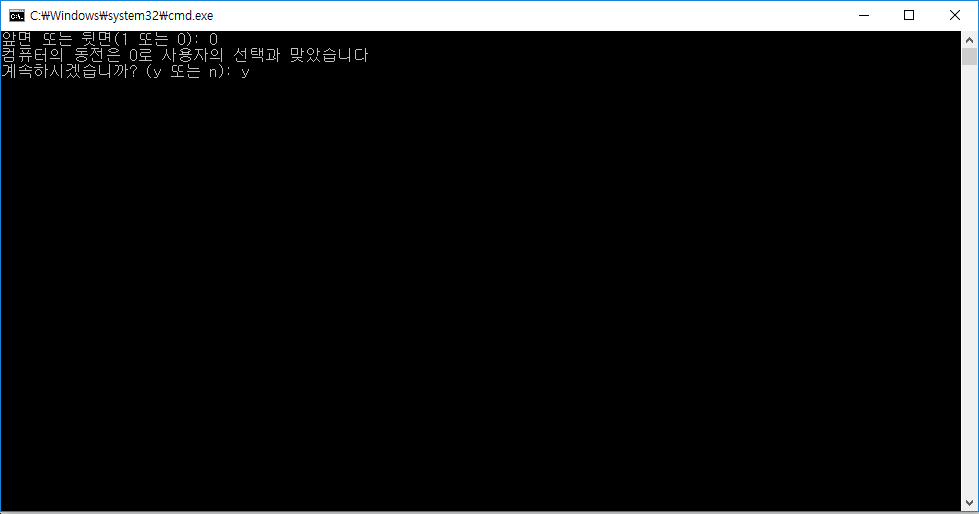
**return (i == n) ? 1 : 0;// 소수일경우 1반환 아닐경우 0반환**

**}**

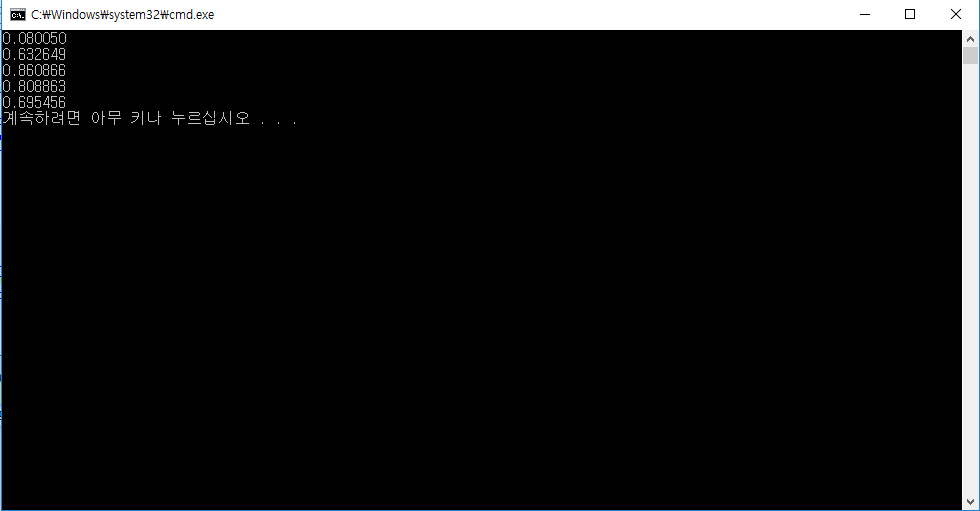
**3-6) 실행결과**



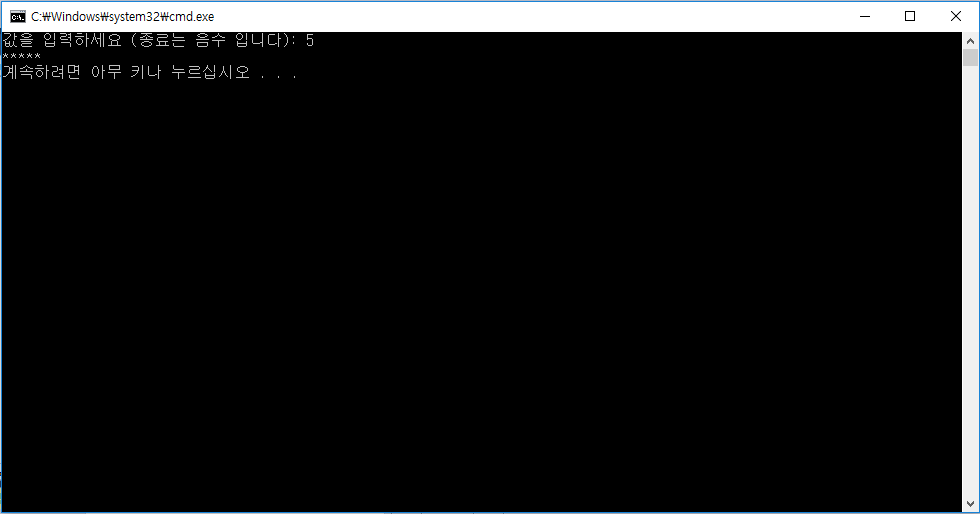
**3-7) 실행결과**



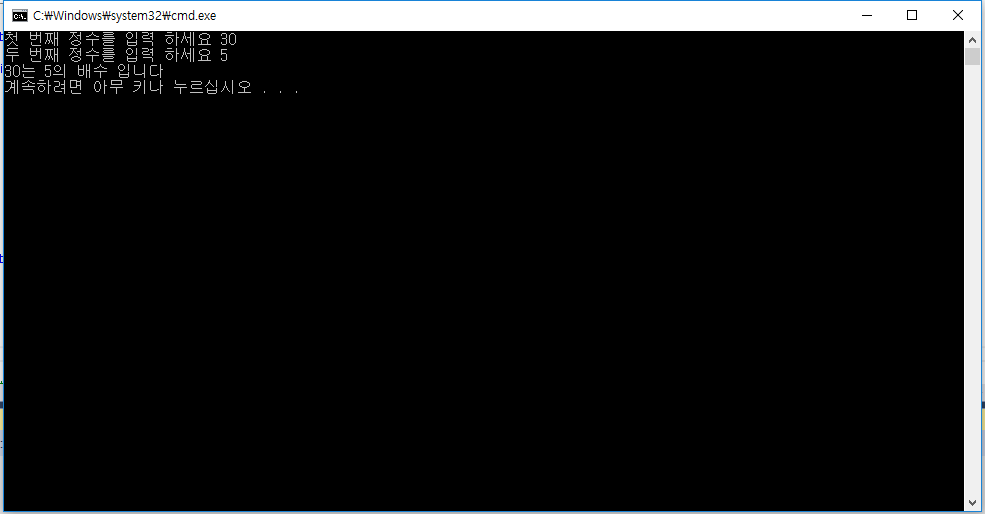
**3-8) 실행결과**



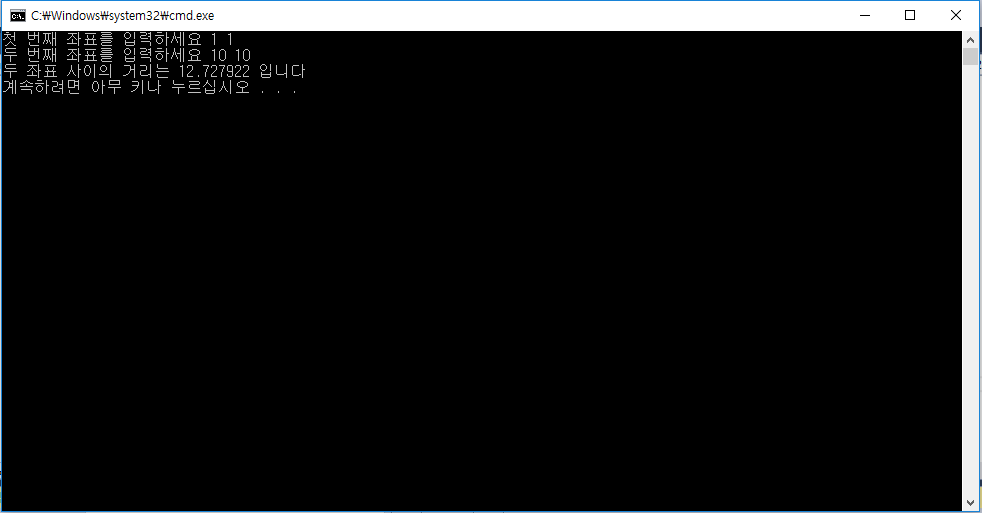
**3-9) 실행결과**



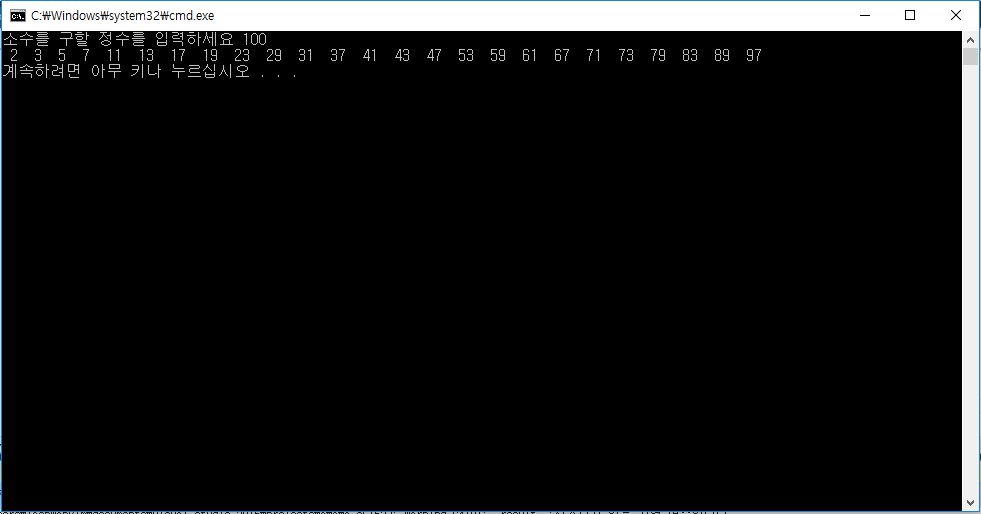
**3-10) 실행결과**



**3-11) 실행결과**



**3-12) 실행결과**



**4-6) 알고리즘**

**Rand()함수에서 %2의 연산을 통해 0또는 1의 2가지의 경우의 수만 random\_number변수에 대입을 하고 이를 반복 문을 통해 5번 출력을 하게 하는 함수 b\_rand()를 만들고 이는 main에서 호출하게 됩니다.**

**4-7) 알고리즘**

**Rand()함수에서 %2의 연산을 통해 0또는 1의 2가지의 경우의 수만 random\_number변수에 대입을 하고 이를 반복 문을 통해 5번 출력을 하게 하는 함수 b\_rand()를 만들고 이는 main에서 호출하게 됩니다. main에서는 이를 이용하여 while(1)을 통해 사용자가 입력하는 수와 비교를 하게 되고 또한 사용자가 프로그램을 계속 사용 할 것인지의 여부 판단을 하고 ‘n’문자를 입력 했을 시 프로그램을 종료하게 됩니다.**

**4-8) 알고리즘**

**F\_rand()함수를 통해 0.0에서 1.0 사이의 랜덤 한 함수를 얻는 식(rand()/(double)RAND\_MAX)을 반환하고 이를 main에서 srand()를 사용하여 계속해서 출력되는 수를 변환을 하게 되고 이를 for을 통해 5번 출력하게 됩니다.**

**4-9) 알고리즘**

**main에서 별을 출력하고자하는 수를 입력 받고 이를 Print\_value함수에 매개변수로 전달하게 되고 이를 함수에서 반복 문을 통해 별을 출력하게 됩니다. main에서는 반복 문에서 계속 수행하게 되고 Print\_value함수를 호출하게 되고 전달되는 매개 변수가 음수 일 경우 반복 문을 빠져나가게 됩니다.**

**4-10) 알고리즘**

**main에서 배수 여부 판단을 위해 2개의 정수를 입력 받고 이를 is\_multiple함수에 매개 변수로 전달하게 합니다. 이를 두수를 나누고 나머지가 0일경우 1을 반환하게 되고 아닐 경우 0이 반환되게 됩니다. 이후 main에서 함수에서 1이 반환 되었을 경우 배수가 되고, 0일 경우 배수가 아니게 됩니다.**

**4-11) 알고리즘**

**main에서 2개의 좌표를 입력 받고 이를 get\_distance함수로 전달하고 이를 2개의 좌표 사이의 거리를 구하는 공식을 통해 구해진 값을 반환하게 됩니다 이를 다신 main함수에서 함수 호출에 따른 출력을 하게 됩니다.**

**4-12) 알고리즘**

**main에서 수를 입력 받은 뒤 이를 is\_prime()함수에 매개변수로 전달된다. is\_prime()함수에선 매개변수와 함수의 지역 변수와 나누어 0일경우 소수가 아니게 되고 이때 반복 문을 나오게 된다. 이후 i와 매개변수를 비교하여 참일 경우 1을 반환하고 거짓일 경우 0을 반환을 하게 되고 이를 main에서 반환된 함수가 1일경우 이를 소수라고 판단하고 이를 출력을 하게 됩니다..**