텍스트, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명1.

2: pow()를 사용하기 위해 선언

6: 차례대로 입력 받은 실수, 합계, 평균, 표준편차를 저장할 변수 선언

7: 입력 받은 자료를 저장할 배열 선언

8: 10번 반복

10: 실수형으로 입력 받음

11: 배열 num에 차례대로 저장

12: 입력 받은 값을 더해 sum에 저장

14: 평균을 구함

15: 표준편차에 쓰일 합계를 저장하기 위해 sum 초기화

16: 10번 반복

18: 실수 값에서 평균을 뺀 후 pow()를 사용해 제곱한 뒤 sum에 더함

20: 18번째 줄에서 나온 sum값을 10으로 나눈 뒤 제곱근을 취하여 stdDev에 저장함

21: 결과 출력

텍스트이(가) 표시된 사진

자동 생성된 설명2.

2: rand()를 위해 선언

3: persentage를 30으로 정의

7: 문자열 minesweeper 선언

8, 10: 문자열 탐색을 위해 반복

12: 나올 수 있는 0~99의 난수가 30보다 작을 때로 조건을 걸어 30%확률을 만듦

14: 30% 확률로 ‘#’이 해당 위치에 들어감

15, 20: 지뢰 판을 출력하기 위해 #이나 .을 출력함

19: 난수가 30보다 크거나 같으면 ‘.’을 넣음

23: 9\*10 사이즈의 지뢰판을 만들기 위해 줄바꿈

3. 텍스트, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

2: rand() 사용을 위해 선언

3: srand() 사용을 위해 선언

6: 전역 변수로 문자열 map 선언

8: map을 출력하는 함수

텍스트이(가) 표시된 사진

자동 생성된 설명

사진 : 변수 선언 및 초기 랜덤 필드 설정

20: 각 역할의 변수 설정

21: 차례로 게임이 끝났는지 확인하는 변수와 몬스터1과 몬스터2의 위치를 결정할 난수 값을 저장하는 변수

22: rand() 값이 고정되지 않게 하기 위함

24~30: map을 전부 ‘.’으로 채움

32: 각 역할이 map의 어느 위치에 있는지 저장하는 배열 선언

38: 각 역할의 위치를 선정 ( map이 10\*10 이므로 0부터 9까지의 값을 저장 )

39: location[0]은 ‘#’, location[1]과 location[2]는 ‘M’, location[3]은 ‘G’의 위치를 저장하고, 이 위치 정보를 통해 map에 각 역할들 표시

텍스트이(가) 표시된 사진

자동 생성된 설명

53: 만들어진 map을 출력함

55: end가 0일 동안만 돌아감. 후에 게임이 끝날 조건이 만족되면 end가 1로 바뀌어 while을 벗어남

59: 실시간 키 입력을 받은것으로 switch문을 시작

텍스트이(가) 표시된 사진

자동 생성된 설명

사진 : 입력받은 키별로 이동하는 코드. map에서 원래 있었던 위치는 ‘.’으로 변환하고, location을 통해 이동한 정보를 저장한 뒤, 이를 토대로 map 상의 이동한 부분에 ‘#’을 저장한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

98: case 이외의 키를 입력받으면 이동키가 아니라고 알림

102~103: 몬스터가 어디로 움직일지 랜덤으로 결정

105~187: 난수값을 토대로 이동. ‘#’이 이동한 것과 같은 방식으로 움직인다.

텍스트이(가) 표시된 사진

자동 생성된 설명

189: G를 찾을 경우 게임을 끝냄

193, 197: 몬스터를 만났을 경우 게임을 끝냄

202: 이동하기 전 화면을 지워준다

203: 이동한 후 map을 출력

4. 텍스트이(가) 표시된 사진

자동 생성된 설명

swap 함수 내에서 포인터를 사용해 a와 b의 주소로 직접 접근하여 값을 변경하였다.

5: a = 100+200 = 300, b = 200

6: a = 300, b = 300-200 = 100

7: a = 300-100 = 200, b = 100

5. 텍스트이(가) 표시된 사진

자동 생성된 설명

14번 째 줄에서 get\_int함수에 x, y의 주소값을 보냈다. 3번째 줄에서 포인트 변수를 선언해주어 주소를 받는다. \*px와 \*py는 해당 주소에 있는 값을 나타내고 px와 py는 주소를 나타낸다. scanf에서는 뒷부분에 입력받은 값을 저장할 주소를 써줘야 하므로 px와 py를 써주면 입력 받은 정수가 px와 py, 즉 x와 y의 주소에 저장된다.

6.

함수 get\_sum\_diff()에서 포인트 변수 int \*p\_sum과 int \*p\_diff를 선언해주어 여기에 sum과 diff의주소를 전달해주었다. 따라서 \*p\_sum와 \*p\_diff는 0이고 p\_sum과 p\_diff는 각각 sum과 diff의 주소이다. x와 y를 더하거나 빼서 저장해주면 되기 때문에 \*p\_sum = x+y와 \*p\_diff = x-y라고 하여 주소에 들어있는 값을 직접 바꾸어준다. 주소에 접근해서 값을 변경하였으므로 main 함수의 sum과 diff의 값도 바뀌게 된다.