**과제1 사다리/미로**



|  |  |
| --- | --- |
| **과 목 :** | **프로그래밍** |
| **제출 일자 :** | **2018. 9. 30** |
| **담당 교수 :** | **김승태** |
| **학 과 :** | **사회복지학과** |
| **학 번 :** | **20170299** |
| **이 름 :** | **김유빈** |

**(1) 사다리**

<순서>

1. 사다리를 어떻게 저장할 것인지 -> 배열
2. 세로선과 가로선을 각각 어떻게 저장할 것인지 -> 0, 1
3. 이미 존재하거나 양 옆에 가로선이 존재하는 경우
4. 세로줄에 가로선 최소 3개 이상
5. 사다리 모양 출력
6. 결과 출력

**<교수님 힌트 동영상 전> - project1**

1. 사다리를 어떻게 저장할 것인지 -> 배열
2. 세로선과 가로선을 각각 어떻게 저장할 것인지 -> ‘|’, 1

char ladder[20][9] = { 0 } -> **세로선과 가로선 모두 저장**

int array[4][5] -> **가로선 위치 저장하는 배열** 따로 만듦

④ 세로줄에 가로선 최소 3개 이상

int wcount = rand() % 3 + 3 -> 각 줄에 랜덤으로 가로선 개수 정함(3~5)

width = rand() % 20; -> 가로선을 생성

array[j][x] = width; -> 그 가로선 위치를 저장

**<array 배열>**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 14 | 16 | 4 |
| 2 | 14 | 5 | 6 |
| 17 | … |  |  |
|  |  |  |  |

<- 1번째 가로선

<- 2번째 가로선

<- 3번째 가로선

<- 4번째 가로선

1. 양 옆에 가로선이 존재하는 경우

if ((width == array[j - 1][i]) && j != 0) {

ladder[width][j \* 2 + 1] = 0;

x--;

break;

}

★2번째 가로선을 생성할 때

3번째 가로선은 아직 만들어지기 전이므로

그 전인 1번째 가로선만 확인하면 됨

⑤사다리 모양 출력

if (ladder[i][j] == 1)

printf("----------"); -> 가로선 출력

else if (ladder[i][j] == 0) -> 공백 출력

printf(" ");

else printf("%c", ladder[i][j]); -> 세로선 출력

사다리 전부를 배열(ladder)로 저장했기 때문에 출력 순서 상관 없다!!(쉬움)

1. 결과 출력

A

B

C

D

E

while (i != 20) {

if ((ladder[i][j] == '|') && (ladder[i][j - 1] == 1))

j-=2,i++;

else if ((ladder[i][j] == '|') && (ladder[i][j + 1] == 1))

j+=2,i++;

else

i++;

}

★★주의점: 단순히 j++/j—만 해서는 안된다. i++도 같이 와야 한다 (그렇지 않으면 갔던 가로선을 또 가는 오류 발생)

**<교수님 힌트 동영상 후> - project1\_2**

1. 사다리를 어떻게 저장할 것인지 -> 배열

char ladder[20][5] = { 0 }; -> 가로선만 저장!!

1. 세로선과 가로선을 각각 어떻게 저장할 것인지 -> 0, 1

for문을 통해 가로선을 20개정도 무작위로 저장

1. 이미 존재하거나 양 옆에 가로선이 존재하는 경우

if문을 이용하여 이를 방지

1. 세로줄에 가로선 최소 3개 이상★

int wcount 변수를 따로 설정하여 각 줄에 몇 개의 가로선이 있는지 카운트!

3개 미만일 경우 가로선 2개를 더 추가

이때!! 2,3번 과정을 다시 반복!! (중요)

(사람) 1 2 3 4 5

1. 사다리 모양 출력 **j= 0 1 2 3 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 0 | x |
| 0 | 1 | 0 | 1 | X |
| 0 | 0 | 0 | 0 | X |
| 1 | 0 | 1 | 0 | X |
| 0 | 1 | 0 | 0 | X |
| 1 | 0 | 1 | 0 | X |
| 0 | 1 | 0 | 1 | X |

1 2 3 4 5

0과 1로 나누어 따로 출력

1. 결과 출력★

옆에 그림은 char ladder[20][5]

사람1 -> j = 0

사람2 -> j = 1

사람3 -> j = 2

사람4 -> j = 3

사람5 -> j = 4

(ex) 사람1을 출발이면 j=0에서 시작

Ladder[i][j]의 값 = 0 -> i++

Ladder[i][j]의 값 = 1 -> j++, i++

사람2(3,4)가 출발하면 j = 1에서 시작

Ladder[i][j]의 값 = 0 && Ladder[i][j-1]의 값 = 0 -> i++

Ladder[i][j]의 값 = 1 -> j++, i++

Ladder[i][j]의 값 = 0 && Ladder[i][j-1]의 값 = 1 -> j--, i++

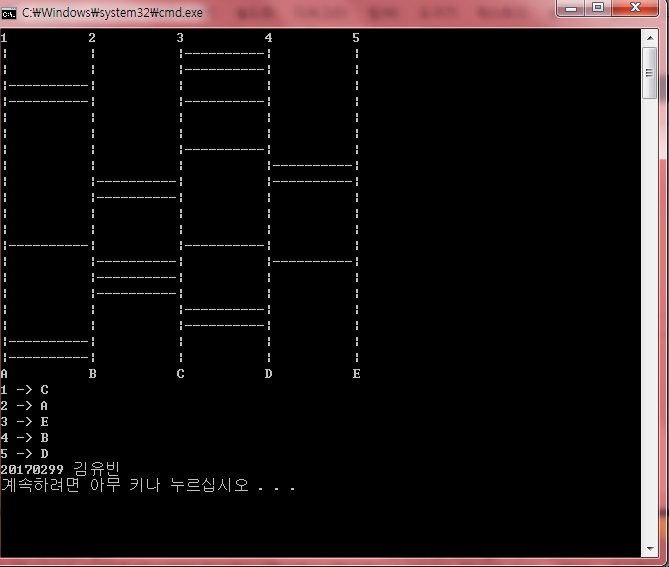
사람5가 출발하면 j=4에서 시작

Ladder[i][j]의 값 = 0 -> i++

Ladder[i][j]의 값 = 1 -> j--, i++

★X(사람)=1, 5 과 2,3,4,를 따로 생각해야함!★

결과= ‘A’ + j



**(2) 미로 그리기**

①공간 or 선 둘 중 무엇을 저장할지 -> **공간**으로 결정

1. 테두리는 저장 안 해도 됨
2. 미로 그리기의 규칙성 찾기 ★

값 = 1 2 3 0

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |

그 외는 저장할 필요가 없다!!

한 가로줄씩 maze[12][9]에 저장

단, 마지막 가로줄일 때는 세로선만 저장하면 됨(값 = 1, 3, 0)

④미로 출력

한 줄마다

1. 세로선 출력 = ‘| ’
2. 가로선 출력 = ‘+---’

단, 처음과 마지막 세로선은 무조건 ‘|’, 가로선은 ‘+’

