

Objective Memo

준혁이는 스티커 메모를 사용하는데 불편함을 느끼고 있다. 메모 안에서 포함관계와 연관관계를 표현할 수 없기 때문이다. 그래서 메모를 직접 만들어서 사용하기로 했다.

아래와 같이 기능을 명세했다.

기능	설명
방향	메모 모드가 세로 또는 가로로 정의할 수 있다.
메모크기	Vertical 모드이면 Width를 정해줄 수 있다. Horizontal 모드이면 Height를 정해줄 수 있다.
메모 안에 메모 삽입	메모 안에 자식으로 메모를 삽입할 수 있다. 자식도 위의 모든 기능을 사용 할 수 있다.
메모 안에 텍스트 데이터 입력	메모에 텍스트를 입력할 수 있다.
메모 안에 이미지 데이터 입력	메모에 이미지를 삽입할 수 있다.

기능의 제약과 상세명세서이다.

기능	내용
텍스트 데이터 길이	1 <= Length <= 3000
이미지크기	1 <= Width, Height <= 500 직사각형
텍스트 방향과 메모크기	가로모드이면 Width 조절이 가능하다. 텍스트의 수가 채울 수 있는 칸을 넘 어가면 자동으로 개행 된다. 가로 모드일 경우 Height는 Auto이다.(자식의 사이즈 만큼 늘어난다) 세로모드이면 Height 조절이 가능하다. 텍스트의 수가 채울 수 있는 칸을 넘어가면 자동으로 개행 된다. 세로모드일 경우 Width는 Auto이다.(자식의 사이즈 만큼 늘어난다.
메모 안에 텍스트 삽입	1. Vertical 인 경우 데이터는 가로로 들어가고 세로로 쌓인다. 가로로 가득 차는 경우 아래로 개행 된다. 2. Horizontal 인 경우 데이터는 세로로 들어가고 가로로 쌓인다. 세로 로 가득 차는 경우 우측으로 개행 된다. 3. 다른 요소들에 의해 메모의 사이즈가 변경되면 개행도 변경될 수 있다.
메모 안에 이미지 삽입	1. 정의 된 이미지 크기의 변경은 없다. 메모의 크기가 이미지보다 작 은 경우 메모의 사이즈는 변경 된다.
메모 안에 메모 삽입	1. 부모 메모는 자식 메모의 사이즈에 영향을 받는다. 자식 메모는 부 모메모의 사이즈의 영향을 받지 않는다.

	2. 메모안에 메모를 삭제 하는 경우는 없다. 3. 작은 부모메모 안에 큰 자식메모가 들어가거나 커지면 부모의 사이즈는 커진다. 사이즈에 의한 텍스트 개행 또한 새로 계산 된다. 4. 부모메모와 자식메모간에 1칸의 테두리가 존재한다.
--	--

추가설명

- 메모안에 여러가지의 데이터와 프레임을 삽입 할 수 있다.
 - Vertical 경우 상에서 하로 쌓인다
 - Horizontal 경우 좌에서 우로 쌓인다.
 - 한 메모에 텍스트가 연속으로 삽입되더라도 이어 붙지 않고 다음 행에 쌓인다.
- 절대 관계가 사이클이 생기지 않는다. 트리형이다.
- 메모의 테두리는 + 로 표시한다.
- 최종 결과물들은 각 $4 \leq \text{width}, \text{height} \leq 5000$ 이다
- Element id 는 0~99이며 생성되는 element 수는 최대 100개이다.

예제

- 가로모드의 메모에 width를 6로 지정하였다. 이 메모의 **id는 0**이다.
 - 현재 이 메모의 크기는 width 6, height 2이다.

+	+	+	+	+	+
+	+	+	+	+	+

- Id 0** 의 메모에 abcdef 텍스트를 추가하였다. 이 데이터의 **id는 1**이다. Width가 6이기 때문에 내부는 4칸이므로 개행 된다.

+	+	+	+	+	+
+	a	b	c	d	+
+	e	f			+
+	+	+	+	+	+

3. **Id 0** 의 메모에 세로모드의 메모를 추가하였다. Height 는 4이다. 이 메모의 **id**는 2이다.

+	+	+	+	+	+
+	a	b	c	d	+
+	e	f			+
+	+	+			+
+	+	+			+
+	+	+			+
+	+	+			+
+	+	+	+	+	+

4. **Id 2**의 메모에 opqrstuvwxy 텍스트를 추가하였다. 이 데이터의 **id**는 3이다.

A. 이때 부모의 width이 수정되면서 ID 1의 text의 개행이 수정된다.

+	+	+	+	+	+	+	+	+	+
+	a	b	c	d	e	f			+
+	+	+	+	+	+	+	+	+	+
+	+	o	q	s	u	w	y	+	+
+	+	p	r	t	v	x		+	+
+	+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+	+

5. **Id 0** 의 메모에 가로모드의 메모를 추가하였다. Width는 10이다. 이 메모의 **id**는 4이다.

+	+	+	+	+	+	+	+	+	+	+
+	a	b	c	d	e	f				+
+	+	+	+	+	+	+	+	+		+
+	+	o	q	s	u	w	y	+		+
+	+	p	r	t	v	x		+		+
+	+	+	+	+	+	+	+	+		+
+	+	+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+	+	+

6. **Id 4** 의 메모에 텍스트 uvuvuv를 추가 하였다. 이 데이터의 **id**는 7이다.

[illegible]

7. **id 2**의 메모에 이미지 4*4를 추가하였다. 이미지데이터는 * 와 #으로 표시되어있다. 이 데이터의 **id**는 **8**이다.

[illegible]

8. **Id 4**의 메모에 텍스트 helloworl 을 추가 하였다. 이 데이터의 **id**는 **12**이다.

[illegible]

아래와 같이 설계하였고 함수를 구현 하라.

함수명	구현내용
Init()	초기화한다.
Void create(int id, Element* e)	<p>메모 or 데이터가 생성된다. Id가 주어진다. 구조체의 정의는 아래와 같다.</p> <pre> struct Element { int type; int id; union { struct { int width; }VerticalFrame; struct { int height; }HorizontalFrame; struct { char* strPtr; }Text; struct { int width, height; char* imgPtr; }Image; }; }; </pre> <p>Type 분류는 아래와 같다 0: memo, vertical 1: memo, horizontal 2: text 3: image</p> <p>Element의 데이터는 주어지고 사라지기 때문에 깊은 복사 필수.</p>
void add(int parentId, int childId)	Parent에 child를 넣는다. 이 때 parentId는 절대 memo가 아닌 data(text,image)가 들어오지 않는다.
void show (int id, char(*refView)[MAXSIZE], int* refWidth, int* refHeight)	<p>메모 Id의 상태를 view 배열에 출력한다. 여기서 id는 메모의 id임을 보장한다. 데이터(이미지,텍스트)의 id는 들어오지 않는다. 최상부모의 id임은 보장하지 않는다. 자식 메모의 id일수도있다. 그리고 출력될 메모의 테두리를 +로, 빈칸은 ' '(space)로 채워야 합니다. 출력된 width, height의 값을 채워야 합니다.</p>

Base code

```
#include <stdio.h>

#define VERTICAL 0
#define HORIZONTAL 1
#define TEXT 2
#define IMG 3
#define MAXSIZE 5000

char view[MAXSIZE][MAXSIZE];

struct Element {
    int type;
    int id;
    union {
        struct {
            int width;
        }VerticalFrame;
        struct {
            int height;
        }HorizontalFrame;
        struct {
            char* strPtr;
        }Text;
        struct {
            int width, height;
            char* imgPtr;
        }Image;
    };
};

void add(int parentId, int childId) {

void create(Element* e) {
```

```
}
```

```
void show(int id, char(*view)[MAXSIZE], int* width, int* height) {
```

```
}
```

```
void init() {
```

```
}
```

```
int main()
```

```
{
```

```
    Element data;
```

```
    int tc;
```

```
    int N;
```

```
    int cmd, id;
```

```
    int parent, child;
```

```
    char text[3500];
```

```
    char imgData[5000];
```

```
    freopen("memoInput.txt", "r", stdin);
```

```
    freopen("memoOutput.txt", "w", stdout);
```

```
    scanf("%d", &tc);
```

```
    while (tc-->0) {
```

```
        scanf("%d", &N);
```

```
        init();
```

```
        for (int i = 0; i < N; i++) {
```

```
            scanf("%d", &cmd);
```

```
            if (cmd == 0) {
```

```
                data.type = cmd;
```

```
                scanf("%d", &data.id);
```

```
                scanf("%d", &data.VerticalFrame.width);
```

```
                create(&data);
```

```
            }
```

```
            else if (cmd == 1) {
```

```
                data.type = cmd;
```

```
                scanf("%d", &data.id);
```

```
                scanf("%d", &data.HorizontalFrame.height);
```

```
                create(&data);
```

```

    }
    else if (cmd == 2) {
        data.type = cmd;
        scanf("%d", &data.id);
        scanf("%s", text);
        data.Text.strPtr = text;
        create(&data);
    }
    else if (cmd == 3) {
        data.type = cmd;
        scanf("%d", &data.id);
        scanf("%d", &data.Image.width);
        scanf("%d", &data.Image.height);
        scanf("%s", imgData);
        data.Image.imgPtr = imgData;
        create(&data);
    }
    else if (cmd == 4) {
        scanf("%d %d", &parent, &child);
        add(parent, child);
    }
    else if (cmd == 5) {
        scanf("%d", &id);
        int width, height;
        show(id, view, &width, &height);
        for (int j = 0; j < height; j++) {
            for (int k = 0; k < width; k++) {
                printf("%c", view[j][k]);
            }
            printf("\n");
        }
        printf("\n");
    }
}

}

return 0;
}

```


input	output
1	++++++
19	+abcd+
0 0 6	+ef +
2 1 abcdef	++++++
4 0 1	+++++++
5 0	+oqsuwy+
1 2 4	+prtvx +
4 0 2	+++++++
2 3 opqrstuvwxyz	+++++++
4 2 3	+osw#***+
0 4 10	+ptx*##*+
5 2	+quy*#*#+
4 0 4	+rv **##+
2 7 uvuvuv	+++++++
4 4 7	+++++++
3 8 4 4 #***#*#*#*#*#	+abcdef +
4 2 8	+++++++ +
5 2	++osw#***+ +
2 12 helloworld	++ptx*##*+ +
4 4 12	++quy*#*#+ +
5 0	++rv **##+ +
	+++++++ +
	+++++++
	++uvuvuv ++
	++helloworld++
	++ ++
	+++++++
	+++++++