Quiz 7

We have a 500 x 500 matrix. Find the smallest number of nonzero elements such that the memory space of 2-D array representation becomes less than that of the sparse matrix representation.

Lets assume the answer is x

2-D array representation

When it comes to 2-D array representation, the required memory space is always the same.

$500 \times 500 \times 4\text{byte} = 1{,}000{,}000\text{ byte}$

(Int)

Sparse matrix representation

$3 \times x \times 4\text{byte} = 12x\text{ byte}$.

the answer x is the smallest integer that conforms following inequality

$12x > 1{,}000{,}000$

$x > 83333.\dot{3}$

So the answer is **83334**

(그러나 sparse matrix에서 맨 앞에 행렬의 meta data가 있다면 정답은 **83333**이다.

Quiz 8

Rewrite the following codes without rowTerms[] to save memory.
Hint: Reuse startingPos[].

```
for(i=1; i<= numTerms; i++){rowTerms[a[i].col]++;}
startingPos[0] = 1;
for(i = 1; i < numCols; i++){
        startingPos[i]=startingPos[i-1]+rowTerms[i-1];}
```

quiz8.c

```c
startingPos[0] = 1;

for (i = 1; i < numCols; i++)
{
    for (i = 1; i < numCols; i++)
    {
        int sum = 0;
        if (a[i - 1].col == i)
        {
            sum++;
        }
        startingPos[i] = startingPos[i - 1] + sum;
    }
}
```

이런식으로

row Terms를 없애면

시간복잡도는 $O(n^2)$으로 늘어나지만

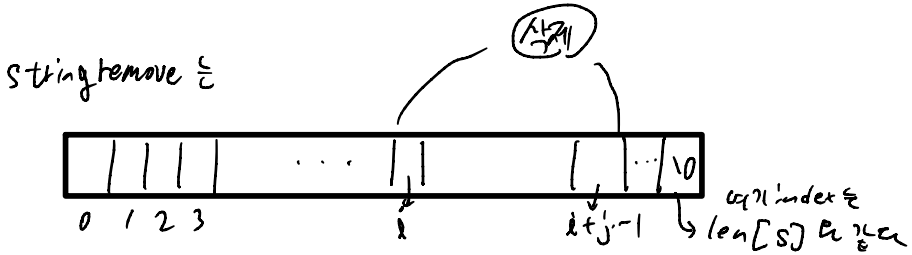
메모리는 아낄 수 있다.

Quiz 9

Write the string remove function to remove j characters beginning from i in string s.

```
void stringremove (char *s, int i, int j) {
```

char s[] = "hello world"; 라면



만약 ① $i+j-1 \le len[s]-1$ 이면 괜찮지만

② $i+j-1 \ge len[s]$ 문제가 생긴다.

② 이 경우에는 $s[i] = '\0'$ 완성되고 $s[i+1] \sim len[s]$까지 free 하면 된다.

① 의 경우에는 ~~s[i] 에서 s[i+j-1] 만큼 뒤 '\0' 을 복사하고~~

$s[i] = s[i+j]$

⋮

$s[\ ] = s[len[s]]$

$len[s]-j$

하면 된다.

그 후 $s[len[s]-j+1]$ 부터

$s[len[s]]$ 까지 free.

의의 선명하게 구현했더니

잘 동작한다.

C quiz9.c > stringremove(char *, int, int)

```c
#include <stdio.h>
#include <stdlib.h>

void stringremove(char *, int, int);

int main(void)
{
    char string[] = "Hello, world!";
    stringremove(string, 3, 33);
}

void stringremove(char *s, int i, int j)
{
    // lens means a length of a string s
    int lens = 0;

    while (s[lens] != '\0')
    {
        lens++;
    }

    if (i + j - 1 <= lens - 1)
    {
        for (int k = i; k <= lens - i - j + 1; k++)
        {
            s[k] = s[k + j];
        }
        for (int k = lens - i - j + 1; k <= lens; k++)
        {
            s[k] = '\0';
        }
    }
    else
    {
        s[i] = '\0';
        for (int k = i + 1; k <= lens; k++)
        {
            s[k] = '\0';
        }
    }

    printf("\nThe result : %s\n", s);
}
```

```
PROBLEMS 3  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

● → 2022-2-data-structure git:(main) ✗ gcc -o quiz8 quiz8.c
● → 2022-2-data-structure git:(main) ✗ ./quiz8

The result : Hel, worl
● → 2022-2-data-structure git:(main) ✗ gcc -o quiz8 quiz8.c
● → 2022-2-data-structure git:(main) ✗ ./quiz8

The result : Hel
○ → 2022-2-data-structure git:(main) ✗
```