

자료구조응용

04. sparse matrix & Pattern Matching

1. 크기가 20*20인 이차원 배열로 희소 행렬을 다음과 같이 만들어라. 0이 아닌 배열 원소의 수를 입력 받아, rand()함수를 이용하여 위치와 값을 설정하도록 하시오. (원소의 개수는 입력 받은 수만큼 되어야 함. 매번 수행시마다 서로 다른 희소행렬이 만들어져야 함), 이 행렬을 이용하여, 배열의 공간을 절약하기 위한 형태의 희소행렬을 만들고 출력하시오.

[실행 예]

[illegible]

2. 다음 희소행렬(sparse matrix)에 대한 전치행렬(transpose matrix)을 구하는 함수로 Program 2.9 fastTranspose를 구현하여 실행결과를 보여라. 출력 화면에 희소행렬과, 전치행렬의 모든 원소들의 값을 출력하고, 희소행렬(sparse matrix) 파일 a.txt의 전치행렬을 아래의 출력 파일 b.txt와 같이 파일에 저장하시오.

```
void fastTranspose1( term a[], term b[]);
```

[프로그램 설명]

① 희소행렬(sparse matrix) 파일 a.txt를 사용하여 전치행렬(transpose matrix)을 구함

입력파일(a.txt)		
6	6	8
0	0	15
0	3	22
0	5	-15
1	1	11
1	2	3
2	3	-6
4	0	91
5	2	28

② 출력 파일의 이름은 b.txt이며 입력 파일 a.txt의 전치행렬(transpose matrix)이다.

출력파일(b.txt)		
6	6	8
0	0	15
0	4	91
1	1	11
2	1	3
2	5	28
3	0	22
3	2	-6
5	0	-15

③ 입력과 출력 파일인 희소행렬(sparse matrix)을 화면에 출력한다. 이때 0의 값을 갖는 항목도 함께 출력한다.

[실행 예]

```
C:\WINDOWS\system32\cmd.exe
A
15  0  0 22  0 -15
 0 11  3  0  0  0
 0  0  0 -6  0  0
 0  0  0  0  0  0
91  0  0  0  0  0
 0  0 28  0  0  0
B
15  0  0  0 91  0
 0 11  0  0  0  0
 0  3  0  0  0 28
22  0 -6  0  0  0
 0  0  0  0  0  0
-15 0  0  0  0  0
계속하려면 아무 키나 누르십시오 . . .
```

3. 위 2번 문제에서 fastTranspose 함수의 파라미터를 구조체 포인터로 정의하여 사용할 수 있다. 다음과 같이 수정된 함수 fastTranspose1을 사용하여 전치행렬을 구하는 프로그램을 작성하라.

```
void fastTranspose1( term *a, term *b);
{
    a, b배열 원소를 구조체멤버 참조연산자 (->)를 사용한 표현으로 바꾸기
}
```

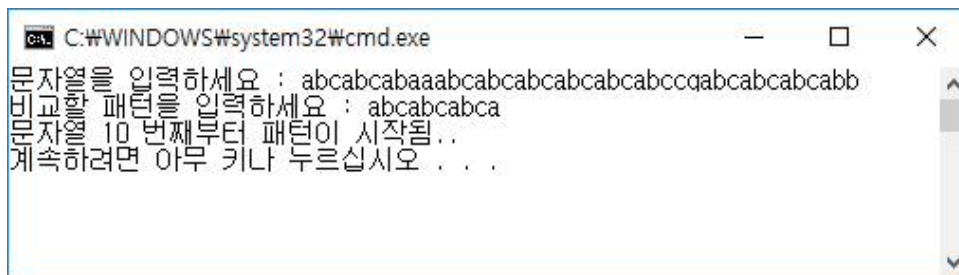
```
int main(void)
{
    ...
    fastTranspose1(a, b);

}
```

```
void fastTranspose1(term *a, term *b)
{
    /* the transpose of a is placed in b */
    int rowTerms[MAX_COL], startPos[MAX_COL];
    int i, j, numCols = a->col, numTerms = a->value;
    ....
}
```

4. 두 개의 스트링 string, pat이 있을 때, pat는 string을 탐색하기 위한 패턴이다. string 안에 pat가 있는지 찾아내기 위해 pat의 마지막 문자를 먼저 검사하는 프로그램을 작성하시오. (program 2.13을 참조)

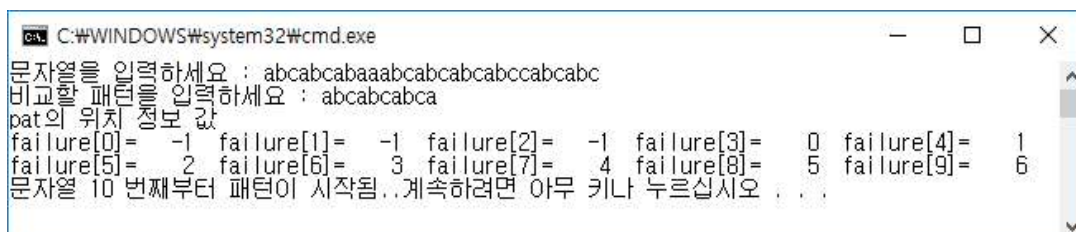
string과 pat 값은 키보드로 입력 받게하고, pat이 string안에 있다면 시작 위치를 출력하시오. (단, 시작위치 0부터 출발)



```
C:\WINDOWS\system32\cmd.exe
문자열을 입력하세요 : abcabcabaaabccqabcbabb
비교할 패턴을 입력하세요 : abcabcabca
문자열 10 번째부터 패턴이 시작됨..
계속하려면 아무 키나 누르십시오...
```

5. 두 개의 스트링 string, pat가 있을 때, pat는 string을 탐색하기 위한 패턴이다. string 안에 pat가 있는지 찾아내기 위해 pat 내의 위치 정보를 이용하여 어디서 탐색을 계속해야 할지 결정 하는 프로그램을 작성하시오. (program 2.14를 참조)

string과 pat 값은 키보드로 입력 받게하고, 패턴 매칭을 위해 작성한 pat의 위치 정보 값 (failure 배열 값)과 pat가 string안에 있다면 시작 위치를 출력하시오. (단, 시작위치 0부터 출발)



```
C:\WINDOWS\system32\cmd.exe
문자열을 입력하세요 : abcabcabaaabccqabcbabb
비교할 패턴을 입력하세요 : abcabcabca
pat의 위치 정보 값
failure[0]= -1 failure[1]= -1 failure[2]= -1 failure[3]= 0 failure[4]= 1
failure[5]= 2 failure[6]= 3 failure[7]= 4 failure[8]= 5 failure[9]= 6
문자열 10 번째부터 패턴이 시작됨..계속하려면 아무 키나 누르십시오...
```

■ 제출 형식

- 솔루션 이름 : DS_04
- 프로젝트 이름 : 1, 2, 3, 4, 5
- 솔루션 폴더를 압축하여 제출할 것.
- 학습관리시스템에 과제를 올릴 때 제목:
 - 1차 제출: 학번_이름_DS_04(1), 2차 제출: 학번_이름_DS_04(2)
 - 제출은 2회걸쳐 가능(수정 시간 기준으로 처리)