**2022 Spring OOP Assignment Report**

과제 번호 : ASSN #1 Prob3

학번 : 20210479

이름 : 이주현

Povis ID : lanthanum14

**명예서약 (Honor Code)**

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

프로그램을 하다 보면 결정해야 할 세부 사항이 많은데, 이러한 세부 사항을 처리한 방법과 이유를 보고서에 쓰십시오.

독창적인 아이디어와 추가 기능은 보너스 점수를 받을 수 있으므로, 보고서에 명확히 기재하십시오.

문제가 여러 개인 경우, 각 문제별로 정리해서 작성합니다.

각 문항별 설명은 편의를 위한 것으로, 삭제하고 제출한다.

1. **프로그램 개요**
   * Prob3은 “input.txt” 안의 미로를 읽어들여 출발 지점에서 k를 거쳐 e까지 가는 최단 거리를 계산하여 그 거리를 “output.txt”에 출력하는 내용의 프로그램이다.
   * 프로그램을 실행하기 위해서는 가로 10, 세로 10 크기의 지도 (임의의 위치에 k, e가 표시되어 있어야 함) 를 넣은 “input.txt”가 있어야 한다.
   * 프로그램이 실행되면 자동으로 “input.txt”가 읽기 전용으로 열리며, 해당 파일이 존재하지 않거나 프로그램 상에서 정상적으로 열리지 않는다면 오류 문구가 출력되고 프로그램이 종료된다. 정상적으로 열린 경우 이어서 “output.txt”가 쓰기 전용으로 열리며, 해당 파일이 존재하지 않으면 생성되어 열린다. 정상적으로 열리지 않은 경우에는 마찬가지로 오류 문구가 출력되고 프로그램이 종료된다. 두 파일 모두 정상적으로 열린다면 “input.txt”의 내용을 읽어들이는 작업으로 넘어간다.
2. **프로그램의 구조 및 알고리즘**

* 지도상의 좌표를 유용하게 다루기 위해 struct position 이라는 구조체를 선언하였다. 구조체 내에는 int 변수 x, y를 선언하였다.
* Linked list로 큐를 구현하기 위해 큐의 한 원소에 해당하는 struct node 라는 구조체를 선언하였다. node 구조체에는 position 변수 p와 다음 원소를 가리키는 node\* next가 포함된다. 또한 큐를 다루기 위한 struct queue 라는 구조체를 선언하였다. 이 구조체 내에는 정보가 나가는 부분인 front와 정보가 들어오는 부분인 rear를 각각 가리킬 node \*front, \*rear와 사이즈를 다룰 int 변수 size를 선언하였다.
  + C++ 헤더 파일 <queue>에 들어있는 pop, push, front, empty를 직접 구현하기 위해 함수로 선언하였고, 이와 더불어 queue를 초기화하기 위한 함수 qInit도 선언하였다.
* 이 프로그램에서는 C++ 파일 입출력(fstream)을 사용하였으며 “input.txt”를 열 변수로 ifstream fi, “output.txt”를 열 변수로 ofstream fo를 선언하였다.
* 지도 저장용 배열로 int형 2차원 배열 map을 선언하였다. 그리고 map 내의 k와 e의 좌표를 따로 저장하기 위한 xk, yk, xe, ye라는 int 변수를 선언하였다. 출발 지점부터 k까지의 최단거리와 k부터 e까지의 최단거리를 따로 계산할 것이므로 각각의 최단거리를 지칭하는 int 변수 kRslt, eRslt를 선언해 주었다.
  + “input.txt”에서 map을 읽어들여 2차원 배열에 저장한다. 이때 map 내에는 숫자와 문자가 섞여 있음을 고려하여 문자를 읽어들일 변수로 char 변수 buffer를 선언하고, 하나씩 읽어들여 k 또는 e일 경우 해당 좌표를 저장한 후 map에는 0과 1이 아닌 다른 숫자로 저장하고, 그냥 숫자일 경우(0 or 1) map에 int 형으로 저장한다.
* 최단 거리를 계산할 함수로 FindOpt를 선언하였다. 이 함수는 시작 좌표(xs, ys)와 목표 좌표(xt, yt), map, 결과를 저장할 int \*rslt를 매개변수로 받는다.
  + 이 함수 내에는 map과 같은 크기의 2차원 배열이 두 개 더 선언되어 있다. 하나는 방문 여부를 기록하는 배열이며, 하나는 출발 지점으로부터 거리를 저장하는 배열이다. 둘 모두 선언된 뒤 모든 원소가 0으로 초기화된다.
  + 이 함수는 출발 좌표부터 그 좌표 상, 하, 좌, 우에 있는 좌표들을 살펴보는 함수이다. 이동 가능하고 방문한 적 없는 좌표들을 모두 큐에 순서대로 기록하여 방문하며 거리를 기록한다. 상, 하, 좌, 우를 고려할 때 map을 벗어나는 좌표(ex. 음수)는 고려하지 않고 패스한다.
  + 그 후 목표 좌표 상, 하, 좌, 우의 거리 배열을 비교하여, 가장 짧은 거리를 \*rslt에 저장한다. 이때 거리 배열에 0으로 저장되어 있는 좌표는 고려하지 않도록 설정한다. (0이면 해당 길로는 도착할 수 없다는 뜻이므로)
* 출발 지점부터 k, k부터 e까지 두 번 FindOpt를 호출해 최단거리를 구해 각각 kRslt, eRslt에 저장한 후에 이를 더하여 rslt에 저장한다. 이후 “output.txt”에 rslt를 출력한다. 이후 완료 문구를 출력하고 프로그램을 종료한다.

1. **토론 및 개선**
   * + 이번 과제를 통하여 linked list를 통해 queue를 구현하는 방법을 복습할 수 있었으며, queue와 함께 쓰는 함수들의 작동 알고리즘을 복습하고 구현해 볼 수 있었다.
     + 2차원 배열의 활용을 복습하고, 함수 호출 인자로 사용될 경우 어떻게 사용해야 하는지 복습할 수 있었다.
     + 어떤 좌표들을 통해 최단거리에 도달하였는지를 “output.txt”에 출력하는 기능을 추가할 수 있을 것 같다.
2. **참고 문헌**

* CSED101 Lab13\_2 ppt
  + linked list를 통해 queue를 구현하고 같이 사용하는 함수들을 구현하기 위한 알고리즘 참고
* <https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=tipsware&logNo=221329432324>
  + 2차원 배열을 함수의 인자로 넘기기 위한 방법 참고