Упражнение №3 по СДП

Стек. Опашка.

Стек.

Задача 1. (Ханойски кули)

В класическата задача за Ханойските кули има три стълба и n наброй диска с различен размер. Дисковете са поставени на първия стълб в намаляващ ред на размера, като най-големият диск е разположен най-отдолу. Като спазвате следните правила:

- може да се мести само един диск – този, който е на върха на стълба;
- един диск може да се сложи върху друг диск, само ако размерът на долния диск е по-голям,

напишете функция, която премества дисковете от първия на последния стълб, използвайте стекове за представянето на стълбовете.

Задача 2.

Дефинирайте функция, която намира стойността на n! като използва стек.

Задача 3.

Дефинирайте прост текстов редактор, който въвежда редица от знакове от клавиатурата, редактира въведения текст и го извежда на екрана. Въвеждането продължава до въвеждане на знака за край на ред. При въвеждане на знака '*', редакторът изтрива знака преди него. При въвеждане на знака '-', редакторът изтрива реда.

Например:

При въвеждане на abcd*efg****hij*klm, в действителност въведеният ред е abhiklm. Ако е въведен редът abcdefgh-, в действителност редът е празен.

Задача 4.

В текстов файл е записан лабиринт в следния формат. На първия ред от файла е записан размерът на лабиринта, цяло число m. Следват m реда с по m елемента, описващи самия лабиринт.

Символът 0 означава клетка, през която може да се премине. Символът 1 означава стена. В лабиринта има точно една мишка означена със символа m и един изход, означен с e.

Да се намери пътя на мишката до изхода, ако такъв съществува и да бъде маркиран със символа '.'.

Примерно съдържание на файла:

6

1 1 1 1 1 1

1 1 1 0 0 1

1 0 0 0 e 1

1 0 0 0 0 1

1 0 0 m 1 1

1 1 1 1 1 1

Опашка.

Задача 0.

Дефинирайте шаблон на клас Опашка, описана с помощта на два стека. Направете оценка на сложността на всяка от операциите. (След като помислите сами, ето едно много добре илюстрирано обяснение тук.)

Задача 1. (Прости задачи с опашка)

a) Дефинирайте шаблон на функция, която проверява дали всички елементи в дадена опашка са различни. Приложете функцията върху опашка от цели числа.
b) Дефинирайте функция, която обръща елементите на опашка от цели числа.
c) Дефинирайте функция, която извежда в нарастващ ред елементите на опашка от цели числа.

Задача 2. (беше в предишното упражнение по погрешка)

Дадена е редица от числа, чиито членове се получават по-следния начин:

- първият елемент е N;
- вторият се получава като съберем N с 1;
- третият – като се умножи първия с 2 и така последователно всеки елемент се събира с 1 и се добавя в края на редица, след което се умножава по 2 и отново се добавя в редицата.

Напишете програма, която за дадено N и p намира p-тия пореден елемент на редицата.

Задача 3.

Нека a и b са дадени цели числа, a < b. Напишете програма, която само с едно преминаване през елементите на масив от числа (без използване на допълнителни масиви) извежда на екрана елементите на масива в следния ред:

- отначало всички числа, които са по-малки от a;
- след това всички числа в интервала [a, b];
- накрая всички останали числа, запазвайки техния първоначален ред.

Задача 4. (Лабиринт)

Даден е лабиринт с размери n x n. Някои от клетките на лабиринта са празни (0), а други са запълнени (x). Можете да се движите от празна клетка до друга празна клетка, ако двете имат обща стена. При дадена начална позиция (*), изчислете и попълнете лабиринта с минималната дължина от началната позиция до всяка друга. Ако някоя клетка не може да бъде достигната, я попълнете с "u”.

Задача 5. (Път в граф, обхождане в ширина)

Неориентиран граф с n върха, номерирани с 1, 2, ..., n е представен с матрица на съседство. Ако има дъга от връх i до връх j, то в матрицата на съседство елементите [i][j] и [j][i] имат стойност 1, в противен случай стойността е 0.

Напишете функция, която проверява дали съществува ацикличен път от връх i до връх j в графа. Да се използва алгоритъма за обхождане в ширина.