```
1  #ifndef _PILE_H_
2  #define _PILE_H_
3
4  #include <string>
5  #include <time.h>
6  #include <iostream>
7  using namespace std;
8
9  class Card;
10
11 class Pile
12 {
13
14 protected:
15     //1)Number of organs in array
16     unsigned int m_size;
17     //2)Size of array
18     const int m_capacity = 54;
19     //3)Array of pointers with m_capacity length
20     Card** m_queue;
21     //4)Index of the first organ in array m_queue
22     int m_left;
23     //Extras:
24     //Index of the last organ in array m_queue
25     int m_right;
26
27 public:
28     //1)Ctor
29     //Initializes the members m_size ,m_left and m_right to 0
30     //creating array m_queue of pointer (length of the array is m_capacity
         (54)) , and initialize every pointer to point to NULL
31     Pile();
32     //2)Dtor
33     //Delete all created dynamic allocations (the pointer that are in the
         array m_queue and the m_queue itself
34     ~Pile();
35     //3)
36     //push to the back of the array m_queue if it is'nt full ,and updating
         m_size
37     void push_back(Card* card);
38     //4)
39     //pop the last organ in array m_queue if it is'nt empty ,and updating
         m_size
40     Card* pop_back();
41     //5)
42     //push to the front of the array m_queue if it is'nt full ,and updating
          m_size
43     void push_front(Card* card);
44     //6)
45     //pop the first organ in array m_queue if it is'nt empty ,and updating
         m_size
46     Card* pop_front();
47     //7)
```

```
48      //Returns the first organ of m_queue
49      const Card& front() const;
50      //8)
51      //Returns the last organ of m_queue
52      const Card& back() const;
53      //9)
54      //Returns the number of organs in m_queue
55      unsigned int size() const;
56      //10)
57      //Returns 1 if m_queue is empty and 0 if not
58      bool is_empty() const;
59      //Extras:
60      //Returns 1 if m_queue is full and 0 if not
61      bool is_full() const;
62      //Receives a push instruction ("push back" or "push front") and
          updating the appropriate index according to the instruction .
          (Auxiliary function for the push functions). It refers to the fact
          that the array is a cyclical array
63      void for_push_next_index(const string& push_back_or_push_front);
64      ////Receives a pop instruction ("pop back" or "pop front") and updating
           the appropriate index according to the instruction . (Auxiliary
          function for the pop functions) .It refers to the fact that the array
           is a cyclical array
65      void for_pop_prev_index(const string& pop_back_or_pop_front);
66      //creating new pile - 24 cat cards , 21 rat cards , 3 peek cards, 3
          draw 2 cards,3 swap cards. randomly!! . //!!!! can do with template
          function!!!!//
67      void new_pile();
68      //Mixes the queue randomly
69      void shuffle();
70      //Gets two indexes and changes them to be random (in range 0-53)
71      void random_2_indexes_0_to_m_capacity_minus1(unsigned int& index_a,
          unsigned int& index_b);
72      //Switches between two pointers
73      void swap_cards(Card*& card_a_pointer, Card*& card_b_pointer);
74  };
75
76  #endif      // _PILE_H_
77
78  //?many questions about const - where to write const? where to write
      static?
```