Celem zadania jest możliwość „projektowania” **kartki świątecznej** (w okienku konsolowym).
We właściwościach okienka konsolowego ustaw np: czcionki_rastrowe 6x8 oraz rozmiar okienka szerokość 80 i wysokość 70.
Docelowo na tle wielkanocnego jaja będą „rysowane” wzorki w postaci poziomych pasków.
Funkcja **main()** jest przygotowana w całości w pliku **prog.cpp** .

ETAP 0

Ogólnie popatrz na klasy występujące w projekcie, zostały wstępnie zaprojektowane.
Należy doprowadzić do **pomyślnej kompilacji i uruchomienia wstępnego aplikacji** (w pierwszej kolejności wykorzystuj **zapowiedź** odpowiedniej **klasy**; a jeśli to nie jest wystarczające, to włącz jej **pełną definicję** dyrektywą `#include`).
Powinno wyświetlić się **MENU**.

ETAP 1

Zapoznaj się z dwiema klasami: `element` oraz `tab`.

Klasa `element`
Tylko obejrzyj (jest gotowa).
Odpowiada za wypisanie znaku w odpowiednim kolorze w okienku konsolowym.

Klasa `tab` (klasa bazowa dla klasy `card`)
Obejrzyj (jest prawie gotowa) i tylko zaimplementuj `operator<<`, który wypisze obiekt tej klasy.
W dynamicznej tablicy 1-wymiarowej przechowujemy postać kwadratowej kartki wymiaru (m x m) – kolejnymi wierszami. Element tablicy jest typu `element`.

ETAP 2

Należy zaimplementować **klasy**

pattern odpowiada za malowanie poziomego paska o poziomej współrzędnej **sx** (indeks wiersza) **elementem** podanym w konstruktorze.
Metoda **draw** wypełnia wybrany wiersz tablicy obrazka podanym **elementem**, czyli znakiem o podanym kolorze, np

* * * * * * * * *

`egg` odpowiada za malowanie jaja, a precyzyjniej zadaniem metody `draw` jest „zamalowanie” elementów tablicy obrazka **wystających** poza kontur jaja (podanym w konstruktorze **elementem**):

- środek jaja `s=m/2` (gdzie m-wymiar obrazka)
- promień jaja: `r=s-1`;
- zamazujemy podanym **elementem** wszystkie elementy `(i,j)` dla których: $(i-s)^2+(j-s)^2>r^2$

`card` **to publiczna pochodna klasy tab**, do zarządzania zawartością obrazka wykorzystujemy interfejs klasy `tab`.
Pole wskaźnikowe `h` odpowiada za pamiętanie historii obrazka (tj. kolejne jego edytowane wersje). Klasa historia będzie implementowana w **etapie 3**.

Obiekty klasy **`card`** nie mogą być kopiowane, ani nie wolno dla nich stosować operator=. Zdefiniuj te właściwości w headerze tej klasy.

Konstruktor klasy **`card`**, powinien wypełnić tablicę podanym w konstruktorze **elementem**, wywołać metodę **`draw`** dla **`egg`** (domyślnego) oraz zainicjować historię (w przypadku **`nullptr`** działamy, ale bez zapisu historii).

Przeciążona metoda **`draw`** w klasie **`card`**, której parametrem jest obiekt klasy **`egg`**, w tablicy obrazka „maluje" domyślne jajo (wywołaj malowanie jaja).

Przeciążona metoda **`draw`** w klasie **`card`**, której parametrem jest obiekt klasy **`pattern`**, w tablicy obrazka „maluje" szlaczek na tle jaja (wywołaj malowanie szlaczka i malowanie domyślnego jaja).

Metoda **`reset`** umożliwia restart tylko pola obrazek, czyli nowa kartka z jajem i być może innym rozmiarem (ale **`historia`** edycji tworzona w etapie 3 jest kontynuowana).

ETAP 3

Należy zaimplementować klasę **`history`** oraz dokończyć implementację klasy **`card`**.
W klasie **`history`** będą zapisywane kolejne wersje tworzonych obrazków (obiekty klasy **`tab`**, tj. fragment bazowy klasy **`card`**)

history

Kolejne wersje obrazka (obiekty klasy **`tab`**) zapisywane są automatycznie do **history**. Jest to klasa narzędziowa dla klasy **`card`** (wszystkie jej składowe są prywatne).
Klasa **`card`** jest zaprzyjaźniona z klasą **`history`**.

Ponadto (podobnie jak **`card`**) **nie posiada** konstruktora bezparametrowego, ani konstruktora kopiującego, ani operator=.
Zdefiniuj te właściwości w headerze tej klasy.

Klasa **`history`** zajmuje się zapisywaniem kolejnych wersji obrazków do tablicy.
Umożliwia **przechowywanie** maksymalnie N kolejnych obrazków lub wykonanie operacji **undo**.

`Obrazek` jest dokładany zawsze na ostatnią wolną pozycję tablicy, a jeśli jest ich za dużo - to usuwamy pierwszy w historii i robimy miejsce na dopisanie.
W tablicy zawsze pozostaje co najmniej 1 obrazek (wtedy bieżący).

Poniżej przykład „malowania"…


C:\Users\EP\source\repos\proj_lab5\Debug\proj_lab5.exe
3 - UNDO
0 - END
Choose number: 1
MENU:
1 - PATTERN
2 - NEW CARD WITH EGG
3 - UNDO
0 - END
Choose number: