1. Изпълнете описаните действия с указатели към стекови променливи
   1. Декларирайте 2 инициализирани стекови променливи от тип double – ***x*** и ***y***
   2. Декларирайте указател **p1,** сочещ променливата **x**
   3. Изведете указателя и сочената от него стойност (чрез указателя и операцията \*)
   4. Променете сочената стойност от указателя p1 (чрез указателя!) и отново изпълнете 1.3
   5. Променете стойността на указателя, така че да сочи променливата **y** и отново изпълнете 1.3
   6. Декларирайте **константен указател** **p2**, сочещ ***х***, и изпълнете за него 1.3
   7. Пробвайте т.1.4 и 1.5 за константния указател- коя от операциите е възможна?
   8. Декларирайте **указател към константа** **p3**, сочещ ***х***, и изпълнете за него 1.3
   9. Пробвайте т.1.4 и 1.5 за указателя към константа- коя от операциите е възможна?
   10. Декларирайте **константен указател** **към константа p4**  и изпълнете за него 1.3
2. Деклариран е масивът names

string names[] {"Ана", "Петър", "Димо", "Стоян", "Асен", "Лиляна", "Димитър",

"Мирослав", "Катя", "Денислав"};

Размерът му се определя според дължината инициализиращия списък:

const int COUNT = sizeof names/sizeof \*names;

Целта на задачата е да се изберат за награждаване 3-ма души по случаен принцип. За упражнение на връзката между масиви и указатели нека използваме алгоритъм, подобен на разгледания на лекцията. Идеята е да се направят 3 последователни разбърквания на масива names: в първото участва целия масив names, във второто – масива names без последния елемент, в третото – масива names без последните два елемента. За избрани се смятат последните 3 лица в масива names, т.е. имената, попаднали на последна позиция в поредното разбъркване (в примера от лекцията, вариант 2, взехме първите числа от всяко разбърване)

За тези, които се затрудняват, описвам алгоритъма

* 1. Изведете масива names, като използвате *цикъл по индекс*
  2. Разбъркайте масива names с функцията random\_shuffle()
  3. Изведете масива names, като използвате *цикъл с параметър указател*
  4. Разбъркайте масива частта от масива names без последния елемент
  5. Изведете масива names, като използвате *цикъл за диапазон* *(enhanced for)*
  6. Разбъркайте масива частта от масива names без последните два елемента
  7. Изведете масива names, като използвате *цикъл за диапазон* *(enhanced for)*
  8. Декларирайте масив for\_awarding[], като го инициализирате с последните 3 елемента на масива names (по реда на „изтеглянето“)
  9. Изведете масива for\_awarding като използвате *цикъл за диапазон* *(enhanced for)*

1. Напишете програма, която извлича в масива **top\_n\_marks** първите **n** най-високи оценки и в масива **bottom\_m\_marks** - последните **m** най-ниски оценки от едномерния масив **marks**, съдържащ оценки от конкурсен изпит (от тип double). Размерите на масивите, както и оценките в изходния масив да се въведат от конзолата. И трите масива – изходния **marks** и двата масива – резултати **top\_n\_marks** и **bottom\_m\_marks**, да се запишат в динамичната памет (heap). Oценките в масива **top\_n\_marks** да бъдат записани от най-високата към най-ниската, а в масива **bottom\_m\_marks** - от най-слабата оценка към по-малко слабата.

Отново записвам примерно решение по стъпки (на които им е лесно, да не ги четат ☺

* 1. Въведете размера на изходния масив **marks**
  2. Заделете памет за масива **marks** в хийпа (масивът е от тип double, т.к. оценките от конкурсните изпити са с точност до стотните)
  3. Въведете елементите на масива **marks** от конзолата

За да си спестите многократно въвеждане, подгответе и вариант , в който оценките се задават чрез инициализиращ списък в т.3.2, тогава цикълът по въвеждане 3.3 се коментира)

* 1. Сортирайте масива **marks** по нарастване (това е подразбиращият се начин на сортиране)
  2. Въведете размера **n**  на масива **top\_n\_marks** (колко на брой най-високи оценки са нужни)
  3. Заделете памет за масива **top\_n\_marks** в хийпа
  4. В масива **top\_n\_marks** запишете последните **n** оценки от сортирания масив **marks** (като започнете от последната, най-високата).
  5. Изведете масива **top\_n\_marks**
  6. Въведете размера **m** на масива **bottom\_n\_marks** (колко на брой най-ниски оценки са нужни)
  7. Заделете памет за масива **bottom\_n\_marks** в хийпа
  8. В масива **bottom\_n\_marks** запишете първите  **m** оценки от сортирания масив **marks**
  9. Изведете масива **bottom\_n\_marks**
  10. Освободете паметта, заемана от трите масива с операцията delete[]