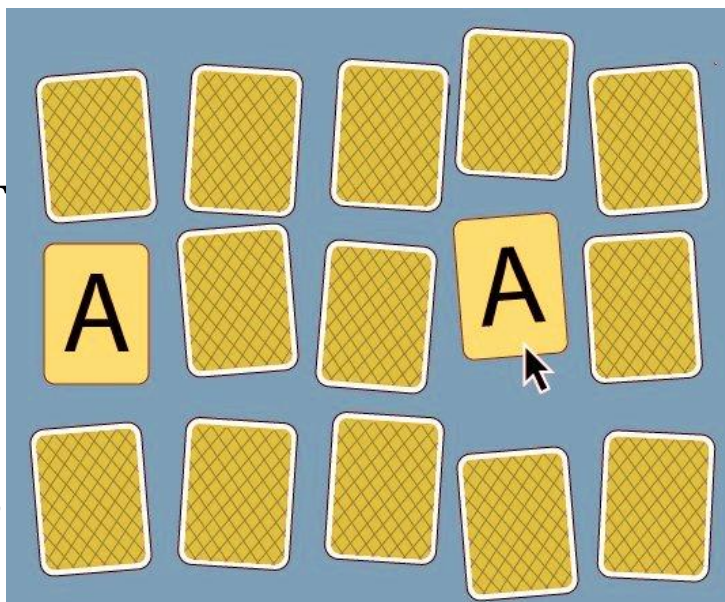




о дня: Память

В игре под названием «Память» используется 50 карт. Каждая из букв от А до Y (ASCII-коды которых соответственно равны от 65 до 89) напечатана на лицевой стороне ровно двух карт. Перед игрой все карты перемешиваются случайным образом и выкладываются на стол лицевой стороной вниз.



Во время игры Джек каждый раз переворачивает две карты лицевой стороной вверх и видит две буквы. Джек получает от мамы конфету за каждую из 25 букв тогда, когда он в первый раз увидит одновременно эту букву на обеих перевернутых картах. Например, когда Джек перевернул две карты и в первый раз увидел на обеих картах букву «М», он получает конфету. После этого, независимо от того, равны буквы на перевернутых картах или нет, карты снова переворачиваются лицевой стороной вниз. Процесс игры повторяется до тех пор, пока Джек не получит все 25 конфет — по одной за каждую букву.

Вам необходимо написать процедуру **play**, которая играет в данную игру. Ваша процедура должна вызывать процедуру **faceup(C)**, которая реализована в системе оценивания, где **C** — целое число от 1 до 50, обозначающее номер конкретной карты, которую необходимо перевернуть лицевой стороной вверх. На момент вызова процедуры **faceup(C)** карта с номером **C** не должна быть перевернута лицевой стороной вверх. Процедура **faceup(C)** возвращает символ, который обозначает букву, изображённую на карте с номером **C**.

После каждого второго вызова процедуры **faceup**, система оценивания автоматически переворачивает карты снова лицевой стороной вниз.

Ваша процедура **play** может завершить свою работу только тогда, когда Джек получил все 25 конфет. Разрешено делать вызовы процедуры **faceup(C)** даже после того, как Джек получит последнюю конфету.

Пример

Ниже приведен пример (вместе с комментариями) одной из возможных последовательностей вызовов, сделанных вашей процедурой **play**.

Вызов	Результат вызова	Комментарии
faceup(1)	'B'	На карте с номером 1 написана буква B.
faceup(7)	'X'	На карте с номером 7 написана буква X. Буквы не совпадают.

<i>Система оценивания автоматически переворачивает карты 1 и 7 лицевой стороной вниз</i>		
faceup(7)	'X'	На карте с номером 7 написана буква X.
faceup(15)	'O'	На карте с номером 15 написана буква O. Буквы не совпадают.
<i>Система оценивания автоматически переворачивает карты 7 и 15 лицевой стороной вниз.</i>		
faceup(50)	'X'	На карте с номером 50 написана буква X.
faceup(7)	'X'	На карте с номером 7 написана буква X. Джек получает первую конфету.
<i>Система оценивания автоматически переворачивает карты 50 и 7 лицевой стороной вниз.</i>		
faceup(7)	'X'	На карте с номером 7 написана буква X.
faceup(50)	'X'	На карте с номером 50 написана буква X. Буквы совпадают, но Джек не получает конфету.
<i>Система оценивания автоматически переворачивает карты 7 и 50 лицевой стороной вниз.</i>		
faceup(2)	'B'	На карте с номером 2 написана буква B.
...		
<i>(Некоторые вызовы процедуры были пропущены)</i>		
...		
faceup(1)	'B'	На карте с номером 1 написана буква B.
faceup(2)	'B'	На карте с номером 2 написана буква B. Джек получает 25-ю конфету.

Подзадача 1 [50 баллов]

Реализуйте любую стратегию, удовлетворяющую правилам игры и завершающуюся в отведённое время.

Например, существует достаточно простая стратегия, которая всегда производит ровно $2 \times (49 + 48 + \dots + 2 + 1) = 2450$ вызовов процедуры **faceup(C)**.

Подзадача 2 [50 баллов]

Реализуйте стратегию, которая решает задачу не более чем за 100 вызовов процедуры **faceup(C)**

Детали реализации

- Используйте [среду программирования и тестирования RunC](#)
- Папка для реализации: /home/ioi2010-contestant/memory/ (Прототип: [memory.zip](#))
- Участник должен реализовать: memory.c или memory.cpp или memory.pas
- Интерфейс участника: memory.h или memory.pas
- Интерфейс системы оценивания: grader.h или graderlib.pas

- Пример системы оценивания: `grader.c` или `grader.cpp` или `grader.pas` и `graderlib.pas`
- Пример ввода для оценивания: `grader.in.1`
Внимание: Входной файл содержит одну строку из 50 символов, описывающих буквы на картах в порядке от 1-ой до 50-ой.
- Ожидаемый вывод на пример ввода: Если ваше решение завершилось корректно и нашло верный ответ, то выходной файл будет содержать строку `OK n`, где n — это число вызовов процедуры **faceup(C)**.
- Компиляция и запуск (из командной строки): `runc grader.c` или `runc grader.cpp` или `runc grader.pas`
- Компиляция и запуск (модуль для gedit): *Control-R*, в момент редактирования файла решения.
- Посылка (из командной строки): `submit grader.c` или `submit grader.cpp` или `submit grader.pas`
- Посылка (модуль для gedit): *Control-J*, в момент редактирования