

```

; Задача о "m" ферзях. Программа на языке Lisp.
; Расставить на шахматной доске с числом клеток "m*m" "m" ферзей так,
; чтобы ни один из них не находился под боем другого.
; Каждая вертикальная линия ("i"=const ) должна содержать ровно одного ферзя.
; Каждая горизонтальная линия ("j"=const ) должна содержать только одного ферзя.
; Каждая диагональ, параллельная главной диагонали ("i"- "j" = const),
; должна содержать не более одного ферзя.
; Каждая диагональ, параллельная побочной диагонали ("i"+"j" = const),
; должна содержать не более одного ферзя.
; (Фомин Владимир Леонидович, ИВГУ, математический факультет)

```

```

(defun nt(n lst o) ; Выводит элемент с номером "n"
  (cond ; из списка "lst" с первым номером "o"
    ((eq n o) (car lst))
    (t (nt (- n 1) (cdr lst) o) )
  )
)

```

```

(defun dl(n lst o) ; Удаляет элемент с номером "n"
  (cond ; из списка "lst" с первым номером "o"
    ((eq n o) (cdr lst))
    (t (cons (car lst) (dl (- n 1) (cdr lst) o)) )
  )
)

```

```

(defun ins(x n lst o) ; Вставляет элемент "x" на место с номером "n"
  (cond ; в список "lst" с первым номером "o"
    ((null lst) (cons x lst))
    ((eq n o) (cons x lst))
    (t (cons (car lst) (ins x (- n 1) (cdr lst) o)) )
  )
)

```

```

(defun len (lst) ; Длина списка "lst"
  (cond ((eq lst nil) 0)
    (t (+ 1 (len (cdr lst)))))
)

```

```

(defun z0(m zn) ; Пустая доска длины "m" и ширины "m"
  (setq z zn)
  (setq z (cons '0 z))
  (if (< (len z) m) (z0 m z) )
  z
)

```

```

(defun a0(m an) ; Все горизонтальные линии свободны
  (setq a an)
  (setq a (cons 't a))
  (if (< (len a) m) (a0 m a))
  a
)

```

```

(defun b0(m bn) ; Все побочные диагонали свободны
  (setq b bn)
  (setq b (cons 't b))
  (if (< (len b) (- (* m 2) 1)) (b0 m b))
  b
)

```

```

(defun c0(m cn) ; Все главные диагонали свободны
(setq c cn)
(setq c (cons 't c))
(if (< (len c) (- (* m 2) 1)) (c0 m c))
c
)

(defun v0(m vn coln) ; Выведены "m" латинских букв для обозначения вертикалей
(setq u '(a b c d e f g h i j k l m n o p q r s t u v w x y z))
(setq col coln)
(setq v vn)
(setq v (cons (nt (+ m col) u 1) v))
(if (< (len v) m) (v0 m v col))
)

(defun prints(z) ; Печать решения и номера решения
(setq count (+ count 1))
(print z)
(princ count)
)

(defun field(i j m a b c) ; Если "t", то поле ("i" ,"j") свободно
(cond ; Если "nil", то поле ("i" ,"j") находится под боем
( (eq (nt j a 1) nil) nil)
( (eq (nt (- i j) b (- 1 m)) nil) nil)
( (eq (nt (+ i j) c 2) nil) nil)
( t t )
)
)

(defun fight(i j m) ; Сделать занятыми ("nil") все три линии,
(setq a (dl j a 1)) ; проходящие через поле ("i" ,"j")
(setq a (ins nil j a 1))
(setq b (dl (- i j) b (- 1 m)))
(setq b (ins nil (- i j) b (- 1 m)))
(setq c (dl (+ i j) c 2))
(setq c (ins nil (+ i j) c 2))
nil
)

(defun nofight(i j m) ; Сделать свободными ("t") все три линии,
(setq a (dl j a 1)) ; проходящие через поле ("i" ,"j")
(setq a (ins t j a 1))
(setq b (dl (- i j) b (- 1 m)))
(setq b (ins t (- i j) b (- 1 m)))
(setq c (dl (+ i j) c 2))
(setq c (ins t (+ i j) c 2))
t
)

```

```

(defun stayferz(i j m a b c) ; Попытка поставить ферзя на поле ("i" ,"j")
(if (field i j m a b c) (setq tryd t) (setq tryd nil) )
(if tryd (setq z (dl i z 1)) )
(if tryd (setq z (ins j i z 1)) )
(if tryd (fight i j m) )
(if tryd (setq step (+ step 1)))
(setq checkuer (+ checkuer 1))
tryd
)
; Если поле ("i" ,"j") свободно, то "tryd" = "t",
; если "tryd" = "t", то ферзь ставится на свободное поле ("i" ,"j"), сделан шаг,
; если "tryd" = "t" , то становятся занятыми все 3 линии,
; проходящие через поле ("i" ,"j"),
; сделана попытка поставить ферзя на поле ("i" ,"j").

```

```

(defun ferz(i j m) ; Решение задачи о ферзях при помощи рекурсии
(stayferz i j m a b c)
(if tryd (if (< i m) (ferz (+ i 1) 1 m) (prints z) ) )
(if tryd (nofight i j m) )
(if (< j m) (ferz i (+ j 1) m) (if (> i 1) (nofight (- i 1) (nt (- i 1) z 1) m)))
)
; 1) Сделана попытка поставить ферзя
; на свободное поле вертикальной линии с номером "i":
; сначала номер горизонтали "j" = 1, если поле ("i" ,"j") занято,
; то при "j"<"m" делается попытка поставить ферзя на поле ("i" ,"j+1")
; 2) Если ферзь успешно поставлен
; на свободное поле вертикальной линии с номером "i",
; то при "i"<"m" делается попытка поставить ферзя
; на свободное поле вертикальной линии с номером "i+1":
; 3) Если свободных полей на вертикальной линии с номером "i" нет
; ("j"="m" && "tryd"="nil"),
; то освобождаются три линии , проходящие через ферзя,
; стоящего на линии с номером "i-1"
; 4) Если "i"="m", то производится распечатка решения задачи,
; после распечатки освобождаются три линии,
; проходящие через поле ("i" ,"j"), где "i"="m"

```

```

(defun f(m) ; Расстановка m ферзей на шахматной доске
(z0 m nil)
(a0 m nil)
(b0 m nil)
(c0 m nil)
(v0 m nil 1)
(print v)
(setq count 0)
(setq checkuer 0)
(setq step 0)
(ferz 1 1 m)
(print "Количество расставляемых ферзей: ")
(princ m)
(print "Всего решений задачи: ")
(princ count)
(print "Сделано всего попыток: ")
(princ checkuer)
(print "Сделано всего шагов: ")
(princ step)
t
)

```