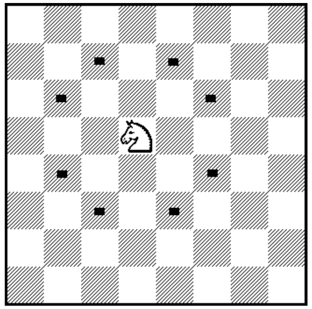
Plan 8

October 28 – November 4

- Read Chapter 9 in the textbook.

- **Exercise 15** Solve the exercise on the next pages.

Exercise 16 The “Knight's Tour” is an ancient puzzle. The object is to move a knight, starting from any square on a chessboard, to every other square, landing on each square only once. A move is made according to the rule of chess, in which a knight can move two squares horizontally and one square vertically, or two squares vertically and one square horizontally (see the figure below).



The C program on the next two pages prints a solution to the puzzle.

Generate x86 assembly code for the recursive function `try` and explain the generated code.

```
#include <stdio.h>
#include <stdlib.h>

int board[8][8];
int delta[8][2] = {{-2, -1}, {-2, 1}, {-1, -2}, {-1, 2},
                   { 1, -2}, { 1, 2}, { 2, -1}, { 2, 1}};

void print_board(){
    int r, c;
    for (r = 7; r >= 0; r--) {
        for (c = 0; c < 8; c++)
            printf("%2d ", board[r][c]);
        printf("\n");
    }
    printf("\n");
    exit(0);
}
```

Cont’d on next page

```
void try(int r, int c, int moves) {
    if (r >= 0 && r <= 7 && c >= 0 && c <= 7 && board[r][c] == 0) {
        board[r][c] = moves + 1;
        if (moves == 63)
            print_board();
        int i;
        for (i = 0; i < 8; i++)
            try(r + delta[i][0], c + delta[i][1], moves + 1);
        board[r][c] = 0;
    }
}

int main() {
    try(0, 0, 0);
}
```