```python
import torch

NUM_EX = 100
NUM_TRIALS = 10


# noinspection PyTypeChecker
def _main():
    print("Homework #3, Problem 2 -- Effect of Dimension on KNN")
    for d in [1, 2, 3, 4, 5, 10, 20, 50, 100, 200, 500]:  # Dimension of X
        acc = 0
        for _ in range(NUM_TRIALS):  # Experiments
            ds = []  # ds is the training and test set respectively
            for _ in range(2):
                y = torch.cat((torch.zeros((NUM_EX // 2, 1)), torch.ones((NUM_EX // 2, 1)
)), dim=0)
                x = torch.randint(0, 2, (NUM_EX, d - 1), dtype=y.dtype)
                ds.append(torch.cat((y, x), dim=1))
            # Find the closest neighbor
            num_correct = 0
            for i in range(NUM_EX):
                dist_i = ds[0].sub(ds[1][i, :]) ** 2  # L2 distance
                pred_val = int(torch.sum(dist_i, dim=1).argmin())  # Closest neighbor
                mid = NUM_EX // 2
                if pred_val < mid and i < mid or pred_val >= mid and i >= mid:
                    num_correct += 1
            acc += num_correct / NUM_EX
        print("d=%d, Accuracy=%.6f" % (d, acc / NUM_TRIALS))


if __name__ == "__main__":
    _main()

```