```
Build := {2 Range@Length@# - 1, 2 #} &;
Compactify@k_ := Sign@#[[1]] # / 2 &[
  Sort[{1, Sign[Times @@ #]} Mod[Abs@If[OddQ@#[[1]], #, #[[{2, 1}]]], 2 Length[k^T], 1] & /@ (k^T)]^T[[2]]];
Data := ToString@# <> ".m" &;
KnotGraph[o_, e_] := Graph[
  Join @@ Table[If[e == 3, Property[#, EdgeWeight → -1], #] & /@ ({v, # - 1} ⟷ {v, Mod[#, 4]} & /@ Range@e) ∪
    {{v, 0} ⟷ {o[[v]], 1}, {v, 2} ⟷ {o[[Mod[v - 1, Length@o, 1]]], 3}}, {v, Length@o}]];
Organize@l_ := If[Length@l ≤ 1, l, Block[{a, e = {}, g = {}, p = RandomChoice@l, s = {}}, a = Abs@p;
   If[# == p, AppendTo[e, #], If[If[Length@# == Length@p,
       If[# == Abs@#, p ≠ a || Order[#, p] == 1, p ≠ a && If[Abs@# == a, Order[#, p] == 1, Order[Abs@#, a] == 1]],
       Length@# < Length@p], AppendTo[s, #], AppendTo[g, #]]] & /@ l;
   Join[Organize@s, e, Organize@g]]];
PermutationConjugate := PermutationProduct[#2, #1, InversePermutation@#2] &;
SetAttributes[Strand, Orderless];
SetDirectory@NotebookDirectory[];
Strand /: Strand[i_, j_] Strand[j_, k_] := Strand[i, k];
Strand /: Strand[i_, i_] := -Sqrt[q] - 1/Sqrt[q];
Strand /: Strand[__]^2 := -Sqrt[q] - 1/Sqrt[q];
Writhe@k_ := Total[If[#[[3]] == Mod[#[[5]] + 1, 2 Length@k, 1], 1, -1] & /@ List @@ k];
Convert := Minimal@Compactify@# &;
DrawGraph@n_ := Graph @@ If[n == 0, {{{}}, {}}, {#[[1]] ⟷ #[[2]] & /@ << Data@n}];
IsPlanar := PlanarGraphQ@KnotGraph[Ordering@Abs@#, 4] &;
IsSorted := # == Organize@# &;
JonesPolynomial@k_ := Sort[{#, # /. q → 1/q} &@@Apart[(-Sqrt[q])^(3 Writhe@#) / Strand[0, 0] Expand[
    # /. X_{a_,b_,c_,d_} :→ Strand[a, b] Strand[c, d] 1/Sqrt[q] + Strand[a, d] Strand[b, c] Sqrt[q]] &@@XNotation@k]][[1]];
Minimal@k_ := Organize[Join @@ Table[Compactify /@ ((s + Abs@Build@k # + 2 Length@k) Sign@Build@k & /@
      {1, -1}), {s, 2 Length@k}]][[1]];
PassReducible@k_ := Block[{l, n = Length@k, p = (#^T ∪ (Abs@Reverse@# Sign@Build@-k)^T)^T[[2]] &@@Build@k,
   v = True}, Do[If[(Sign@p[[Mod[i + j (Range@o - 1), 2 n, 1]]] ∪ {})^2 == {1},
    If[o == 3, Goto@l, Do[If[Total@Mod[e, 2] == 2, If[# == Abs@p[[#]] ∪ {} &[
         Union @@ Mod[(Range @@@ Partition[Mod[e, 2 n, e[[1]], 2]])[[;; , 2 ;; -2]], 2 n, 1]], Goto@l]],
      {e, Select[SortBy[{#, i} ∪ Abs@p[[Mod[{i, i + j}, 2 n, 1]]], Mod[#, 2 n, i] &] & /@ Range[2 n],
        Length@# == 4 &]^T[[If[j == 1, {2, 3, 4, 1}, ;;]]]^T}]]], {o, {3, 2}}, {i, 2 n}, {j, {1, -1}}];
  v = False;
  Label@l;
  v];
XNotation@k_ :=
 Block[{a = Abs@k, n = Length@k, o = Ordering@Abs@k, r, s, t, v}, t = FindSpanningTree@KnotGraph[o, 3];
  v = RowReduce[Join[Abs@#^T, {Sign /@ Abs /@ Total /@ #}]^T, Modulus → 2][[;; n, -1]] &@@
    Table[Total[If[MemberQ[p[[2]]^T[[1]], #[[1]]], {r, s} = {#[[2 ;;]], p[[2, Position[p[[2]]^T[[1]], #[[1]]][[1, 1]], 2 ;;]]};
      If[Length[r ∩ s] == 1, Order @@ Mod[(Complement @@@ {{r, s}, {s, r}}^T - (r ∩ s))[[1]], 4]
        Table[If[j == #[[1]], 1, 0], {j, n}], 0 & /@ k], 0 & /@ k] & /@ p[[1]],
     {p, Map[Join[#[[1]], {#[[-1, 2]]}] &, Map[SplitBy[FindPath[t, {Mod[(# + Mod[#, 2, 1])/2, n, 1],
          2 - 2 Mod[#, 2]}, {o[[(# + Mod[#, 2])/2]], 3 - 2 Mod[#, 2]}][[1]], First] &, Subsets[
       Select[Range[2 n], VertexDegree[t, If[OddQ@#, {(# + 1)/2, 0}, {o[[#/2]], 3}]] == 1 &], {2}], {2}], {3}]}];
  Times @@ (X_## &@@@ ({2 # - 1, 2 a[[#]], 2 #, Mod[2 a[[#]] + 1, 2 n, 1]}[[If[Sign@k[[#]] == 1, ;; , {2, 3, 4, 1}]]][[
```

```
            If[v[[#]] == 0, ;; , {1, 4, 3, 2}]] & /@ Range@n)) ];
AllFlypes[k_, l_] := Block[{a, c, e, n = Length@k, p = (#ᵀ ⋃ Reverse@#ᵀ)ᵀ[[2]] &@@Build@k, y = {}},
  Do[c = Mod[2 i - 1 + s[[1]] Range@o, 2 n, 1];
   For[e = Max@Mod[Complement[p[[c]], c] - s[[2]] 2 k[[i]], 2 n, 1],
    e < Mod[s[[2]] (2 i - 1 - 2 k[[i]]), 2 n, 1], e++, c = c ⋃ Mod[2 k[[i]] + s[[2]] Range@e, 2 n, 1];
    If[Sort@p[[c]] == c, y = y ⋃ {l, Convert /@ Map[Mod[a = Abs@#;
              a + If[Mod[s[[1]] (a - 2 i + 1), 2 n, 1] ≤ o, -s[[1]], If[Mod[s[[2]] (a - 2 k[[i]]), 2 n, 1] ≤ e, -s[[2]],
                 If[a == 2 i - 1, s[[1]] o, If[a == 2 k[[i]], s[[2]] e, 0]]]], 2 n, 1] Sign@# &, Build /@ l, {3}]}ᵀ]],
   {i, n}, {s, {{1, 1}, {1, -1}, {-1, 1}}}, {o, 2, Mod[s[[1]] (2 k[[i]] - 2 i + 1), 2 n, 1] - 1}];
  y];
CandidateKnots@n_ := If[List @@ SavedCandidateKnots@n === {n},
  SavedCandidateKnots@n = If[n == 0, {{}}, Block[{k, l, p, y = {}}, For[p = 0, p < n!, p++, k = {};
      Do[AppendTo[k, Delete[Range@n, {k}ᵀ][[Floor@ Mod[p, (n - i + 1)!]/(n - i)! + 1]]];
       If[2 k[[1]] - 1 > Min[#, 2 n - #] &@@Abs[2 i - 1 - 2 k[[i]]], p += (n - i)! - 1;
        Goto@l];
       If[k[[-1]] ≤ i, Do[If[# == Range[j, i] || # == Range[j, i] - 1, p += (n - i)! - 1;
            Goto@l] &[k[[j ;;]] ⋃ {}], {j, If[i == n && n > 1, 2, 1], i}]], {i, n}];
      If[IsPlanar@k && k == Minimal@k, AppendTo[y, k]];
      Label@l];
     y]], SavedCandidateKnots@n];
Homomorphisms[k_, o_] := If[k == {},
  Length /@ Values@GroupBy[Permutations@Range@o, Sort[Length /@ (List @@ PermutationCycles@#)[[1]]] &],
  Block[{e, s, v, w = List @@@ List @@ XNotation@k}, s = (w //. (Max@# → Min@# & /@ w[[ ;; , {3, 5}]]))ᵀ[[2 ;; 4]]ᵀ;
   e = Complement @@@ ({#, {{}} ⋃ #[[ ;; -2]]}ᵀ &@@FindShortestPath[#, Sort[ConnectedComponents[#,
              {Union @@ s}][[1]]][[1]], Union @@ s] &@@Graph@Flatten@Table[{Union[#, i[[ ;; 2]]] ⟷ Union[#, i],
             Union[#, i[[2 ;;]]] ⟷ Union[#, i]} & /@ Subsets[Complement[Union @@ s, i]], {i, s}]);
   s = SortBy[If[Order[Position[Join @@ e, #[[1]]], Position[Join @@ e, #[[3]]]] == 1, #, Reverse@#] & /@ s,
     Max@Table[Position[Join @@ e, #[[j]]], {j, 2}] &];
   Total /@ Table[v = 0 & /@ Range[2 Length@k];
     v[[e[[1]]]] = g;
     If[And @@ Table[(If[v[[c[[3]]]] === 0, v[[c[[3]]]] = #];
           v[[c[[3]]]] == #) &&PermutationConjugate[v[[c[[1]]]], If[#[[3]] == Mod[#[[5]] + 1, 2 Max@e, 1], v[[c[[2]]]],
            InversePermutation@v[[c[[2]]]]] &&SortBy[w, Length[c ⋂ #] &][[-1]], {c, s}], 1, 0],
     {p, Values@GroupBy[Permutations@Range@o, Sort[Length /@ (List @@ PermutationCycles@#)[[1]]] &]},
     {g, Tuples[p, Length@e[[1]]]}]]];
IsReducible@k_ := Or @@ (Mod[# - Abs@k[[#]], Length@k] ≤ 1 & /@ Range@Length@k) || PassReducible@k;
P2@k_ := Block[{a, c, n = Length@k, p = (#ᵀ ⋃ (Abs@Reverse@# Sign@#)ᵀ)ᵀ[[2]] &@@Build@k, v, y = {}},
  Do[v = Abs@p[[Mod[{i, i + 1}, 2 n, 1]]];
   If[Sort@Sign[p[[Mod[{i, i + 1}, 2 n, 1]]]] == {-1, 1},
    Do[If[Total@Mod[l, 2] == 2, c = Range @@@ Partition[l + {1, -1, 1, -1}, 2];
      If[¬ MemberQ[Join @@ c, i], l = RotateLeft@l;
       c = Mod[Range @@@ Partition[l + {1, -1, 1, 2 n - 1}, 2], 2 n, 1]];
      If[Length[Join @@ c] < 2 n - 4 && v ⋃ Join @@ c == Abs@p[[Join @@ c]] ⋃ Mod[{i, i + 1}, 2 n, 1],
       AppendTo[y, Convert[Build@k /. x_Integer :> If[a = Abs@x;
            Length[v ⋂ l[[ ;; 2]]] == 1, Mod[If[MemberQ[v ⋃ Join @@ c, a], If[MemberQ[c[[1]], a],
              a + If[Mod[Abs@p[[l[[1]]]] - i, 2 n] > 1, 1, -1], If[MemberQ[c[[2]], a], a + If[Mod[Abs@p[[l[[3]]]] - i,
                 2 n] > 1, 1, -1], (l[[{2, 1, 4, 3}]] + {-1, 1, -1, 1})[[Position[l, If[OddQ[l[[1]] + l[[2]]],
                 Total@v - a, a]][[1, 1]]]]], a], 2 n, 1] If[MemberQ[v ⋃ Join @@ c, a] || EvenQ@a,
            If[Mod[a - i, 2 n] ≤ 1 || OddQ@a && ¬ MemberQ[l, a], -Sign@p[[a]], 1], Sign@p[[a]]],
           Mod[If[MemberQ[v, a], SortBy[Delete[l, FirstPosition[l, #] & /@ v], Mod[#, 2] &][[Mod[x, 2] + 1]],
              a + If[MemberQ[Join @@ c, a], 0, If[MemberQ[v, l[[Ordering[Mod[a - l, 2 n, 1]][[1]]]]], -1, 1]]],
             2 n, 1] If[OddQ@x, Sign@p[[a]] If[MemberQ[v ⋃ Join @@ c, a], 1, -1], 1]]]]],
      {l, Sort@Join[#, v] & /@ Subsets[Delete[Range[2 n], Mod[{{i, i + 1}}ᵀ, 2 n, 1]], {2}]}]], {i, 2 n}];
```

```
  Organize[Minimal /@ y ⋃ {}]];
R1@k_ := Organize[Table[
    Convert[(Map[# + If[Abs@# ≥ 2 i, 2 Sign@#, 0] &, Build@k^T, {2}] ⋃ {2 i + {0, 1}})^T], {i, Length@k}] ⋃ {}];
R3@k_ := Block[{b, f, n = Length@k, p = (#^T ⋃ (Abs@Reverse@# Sign@#)^T)^T[[2]] &@Build@k, v, y = {}},
  b = #[[Mod[#[[1]] + {1, -1}, 2 n, 1]]] &@Abs@p;
  Do[f = If[OddQ@i, Abs@p[[#]], #] &@Mod[Abs@p[[i]] + {1, -1}, 2 n, 1];
   Do[If[Total[Sign@p[[#]]]^2 == 1 && MemberQ[(Abs@p[[#]] ⋃ #)[[2 ;; 3]], i] &@{c, i - 1, i},
     v = 1/2 p[[{Abs@p[[c]], Abs@p[[i - Mod[i, 2]]], i - Mod[i + 1, 2]}]];
     If[DuplicateFreeQ@v, AppendTo[y, k /. (v[[#1]] → -Abs@v[[#2]] Sign@v[[#3]] &@@@
         {{1, 2, 3}, {2, 3, 1}, {3, 1, 2}})]]], {c, b ⋂ f}];
   b = f, {i, 2, 2 n}];
  Organize[Minimal /@ y ⋃ {}]];
Flype := Organize[AllFlypes[Abs@#, {#}]^T[[2]] &;
AlternatingKnots@n_ := If[List @@ SavedAlternatingKnots@n === {n},
  SavedAlternatingKnots@n = If[n == 0, {{}}, Organize[First /@ Organize /@ ConnectedComponents@
      Graph[Join @@ Table[Sort[k ⟷ #] & /@ Flype@k, {k, CandidateKnots@n}]]]], SavedAlternatingKnots@n];
ValidKnots@n_ := If[List @@ SavedValidKnots@n === {n}, SavedValidKnots@n = If[n == 0, {{}},
   Organize@Select[Join @@ Table[c # & /@ AlternatingKnots@n, {c, Tuples[{1, -1}, n][[ ;; 2^(n-1)]]}],
     ¬ PassReducible@# && Minimal@# == # &]], SavedValidKnots@n];
CreateGraph@n_ := (If[n == 0, {}, Block[{r, y = {}}, y = Join[Reverse@Organize@#[[ ;; 2]], {#[[3]]}] & /@
      (Join[#, {"Flype"}] & /@ Union @@ AllFlypes /@ CandidateKnots@n ⋃ Flatten[Table[
          {{k, #, "Reidemeister 3"} & /@ R3@k, {k, #, "2-Pass"} & /@ P2@k}, {k, ValidKnots@n}], 3]) ⋃ {};
   r = Join @@ Select[ConnectedComponents@Graph[#[[1]] ⟷ #[[2]] & /@ y], Or @@ PassReducible /@ # &];
   Sort[Select[y, ¬ MemberQ[r, #[[1]]] &], If[#1[[1]] == #2[[1]], If[#1[[2]] == #2[[2]],
     Order[#1[[3]], #2[[3]]] ≥ 0, IsSorted@{#1[[2]], #2[[2]]}], IsSorted@{#1[[1]], #2[[1]]}] &]] >> Data@n);
KnotAssociation@n_ := If[List @@ SavedAssociations@n === {n}, SavedAssociations@n =
   Association@Table[k → Select[ValidKnots@n, Abs@# == k &], {k, CandidateKnots@n}], SavedAssociations@n];
RolfsenTable[n_, o_] := If[n == 0, {{}}, Block[{a = ValidKnots@n, e, l, r, t = {}, v},
   For[l = 0, a ≠ {}, l++, e = Join @@ Table[Sort[k ⟷ #] & /@ If[l == 0, R3@k ⋃ P2@k, R1@k], {k, a}] ⋃ If[l == 0,
     Sort[#[[1]] ⟷ #[[2]] & /@ Union @@ (AllFlypes[#, KnotAssociation[n]@#] & /@ AlternatingKnots@n), {}];
    r = Join @@ Select[ConnectedComponents@Graph@e, Length /@ # ⋃ {} == {n} && Or @@ PassReducible /@ # &];
    e = Select[e, r ⋂ List @@ # == {} &];
    While[v = Join @@ List @@@ e;
     a = Complement[Select[v, Length@# ≠ n &], a];
     a ≠ {}, e = e ⋃ Join @@ Table[Sort[k ⟷ #] & /@ R3@k, {k, a}];
     a = v];
    a = First /@
      Organize /@ Select[ConnectedComponents@Graph@e, Length /@ # ⋃ {} ≠ {n} || Nor @@ PassReducible /@ # &];
    Do[If[(i = i /@ a) ≠ {}, t = Select[{a, i}^T, Count[i, #[[2]]] == 1 &]^T[[1]] ⋃ t;
      a = Complement[a, t]], {i, {If[# == Abs@#, #, 0] &, JonesPolynomial, Homomorphisms[#, o] &}}]];
   Organize@t]];
Timing[{Length@#, #} &[Join @@ (RolfsenTable[#, 5] & /@ Range[0, 10])]]
```

```
{1400.437500,
 {250, {{}, {2, 3, 1}, {2, 3, 4, 1}, {2, 4, 5, 1, 3}, {3, 4, 5, 1, 2}, {2, 4, 5, 1, 6, 3}, {2, 4, 5, 6, 1, 3},
   {2, 4, 6, 5, 1, 3}, {2, 4, 5, 6, 1, 7, 3}, {2, 4, 6, 1, 7, 3, 5}, {2, 5, 6, 7, 1, 4, 3}, {2, 5, 7, 6, 1, 4, 3},
   {3, 5, 6, 7, 1, 2, 4}, {3, 5, 6, 7, 2, 1, 4}, {4, 5, 6, 7, 1, 2, 3}, {2, 4, 5, 7, 1, 8, 3, 6},
   {2, 4, 6, 1, 7, 3, 8, 5}, {2, 4, 6, 1, 7, 8, 3, 5}, {2, 4, 6, 1, 8, 7, 3, 5}, {2, 4, 7, 5, 1, 8, 3, 6},
   {2, 5, 6, 7, 1, 8, 3, 4}, {2, 5, 6, 7, 1, 8, 4, 3}, {2, 5, 6, 7, 8, 1, 3, 4}, {2, 5, 6, 7, 8, 1, 4, 3},
   {2, 5, 7, 8, 6, 1, 4, 3}, {2, 5, 8, 7, 6, 1, 4, 3}, {3, 4, 5, 6, 7, 8, 1, 2}, {3, 4, 6, 1, 7, 8, 2, 5},
   {3, 4, 6, 7, 2, 8, 1, 5}, {3, 4, 7, 6, 2, 8, 1, 5}, {3, 5, 6, 7, 8, 2, 1, 4}, {3, 5, 6, 8, 7, 2, 1, 4},
   {3, 6, 5, 8, 7, 2, 1, 4}, {2, 4, -6, 1, -7, -3, -8, -5}, {2, 4, -6, 1, 7, -3, 8, 5},
   {2, 4, -6, 1, -7, -8, -3, -5}, {2, 4, 5, 7, 1, 8, 3, 9, 6}, {2, 4, 5, 7, 1, 8, 9, 3, 6},
   {2, 4, 5, 7, 1, 9, 8, 3, 6}, {2, 4, 6, 1, 8, 3, 9, 5, 7}, {2, 4, 6, 1, 8, 7, 3, 9, 5}, {2, 4, 6, 7, 1, 8, 9, 5, 3},
   {2, 4, 7, 1, 8, 9, 3, 6, 5}, {2, 4, 7, 1, 9, 8, 3, 6, 5}, {2, 4, 7, 5, 1, 8, 9, 3, 6}, {2, 4, 7, 5, 1, 9, 8, 3, 6},
   {2, 4, 7, 6, 1, 8, 9, 5, 3}, {2, 5, 6, 7, 1, 9, 8, 3, 4}, {2, 5, 6, 7, 1, 9, 8, 4, 3}, {2, 5, 6, 7, 8, 1, 3, 9, 4},
```

{2, 5, 6, 7, 8, 1, 9, 4, 3}, {2, 5, 6, 8, 1, 4, 9, 3, 7}, {2, 5, 6, 8, 7, 1, 9, 4, 3}, {2, 5, 7, 6, 8, 1, 3, 9, 4},
{2, 5, 7, 8, 1, 9, 4, 3, 6}, {2, 5, 7, 8, 6, 1, 9, 3, 4}, {2, 5, 7, 8, 6, 1, 9, 4, 3}, {2, 5, 8, 7, 1, 9, 4, 3, 6},
{2, 6, 7, 8, 9, 1, 5, 3, 4}, {2, 6, 7, 8, 9, 1, 5, 4, 3}, {2, 6, 8, 9, 7, 1, 4, 5, 3}, {2, 6, 8, 9, 7, 1, 5, 4, 3},
{2, 6, 9, 8, 7, 1, 5, 4, 3}, {3, 4, 5, 8, 7, 9, 2, 1, 6}, {3, 5, 7, 6, 8, 1, 9, 2, 4}, {3, 5, 7, 9, 2, 8, 1, 4, 6},
{3, 5, 7, 9, 2, 8, 4, 1, 6}, {3, 5, 7, 9, 8, 1, 4, 2, 6}, {3, 6, 7, 8, 9, 1, 2, 5, 4}, {3, 6, 7, 8, 9, 2, 1, 5, 4},
{3, 6, 7, 9, 8, 1, 2, 5, 4}, {3, 6, 7, 9, 8, 2, 1, 5, 4}, {3, 8, 7, 6, 2, 1, 9, 5, 4}, {4, 6, 7, 8, 9, 1, 2, 3, 5},
{4, 6, 7, 8, 9, 1, 3, 2, 5}, {4, 6, 8, 7, 9, 2, 1, 3, 5}, {5, 6, 7, 8, 9, 1, 2, 3, 4},
{2, 4, 5, −7, 1, −8, −3, −9, −6}, {2, 4, 5, −7, 1, 8, −3, 9, 6}, {2, 4, 5, 7, 1, −8, 3, −9, −6},
{2, 4, 5, −7, 1, −8, −9, −3, −6}, {2, 5, −7, −6, −8, 1, −3, −9, −4}, {2, 5, −7, −6, 8, 1, −3, 9, 4},
{3, 4, 5, 8, 7, −9, 2, 1, −6}, {3, −5, −7, 6, −8, −1, 9, −2, −4}, {2, 4, 5, 7, 1, 8, 9, 3, 10, 6},
{2, 4, 5, 7, 1, 9, 3, 10, 6, 8}, {2, 4, 5, 7, 1, 9, 8, 3, 10, 6}, {2, 4, 5, 8, 1, 9, 10, 3, 7, 6},
{2, 4, 5, 8, 1, 10, 9, 3, 7, 6}, {2, 4, 6, 1, 7, 9, 3, 10, 5, 8}, {2, 4, 6, 1, 8, 3, 9, 5, 10, 7},
{2, 4, 6, 1, 8, 3, 9, 10, 5, 7}, {2, 4, 6, 1, 8, 3, 10, 9, 5, 7}, {2, 4, 6, 1, 9, 7, 3, 10, 5, 8},
{2, 4, 6, 7, 1, 8, 10, 9, 5, 3}, {2, 4, 6, 9, 1, 8, 10, 3, 5, 7}, {2, 4, 7, 1, 8, 9, 3, 10, 5, 6},
{2, 4, 7, 1, 8, 9, 3, 10, 6, 5}, {2, 4, 7, 1, 8, 9, 10, 3, 5, 6}, {2, 4, 7, 1, 8, 9, 10, 3, 6, 5},
{2, 4, 7, 1, 9, 8, 3, 6, 10, 5}, {2, 4, 7, 1, 9, 10, 8, 3, 6, 5}, {2, 4, 7, 1, 10, 9, 8, 3, 6, 5},
{2, 4, 7, 5, 1, 9, 3, 10, 6, 8}, {2, 4, 7, 6, 1, 8, 10, 9, 5, 3}, {2, 4, 8, 5, 1, 9, 10, 3, 7, 6},
{2, 4, 8, 5, 1, 10, 9, 3, 7, 6}, {2, 4, 9, 6, 1, 8, 10, 3, 5, 7}, {2, 5, 6, 7, 8, 1, 10, 9, 4, 3},
{2, 5, 6, 7, 9, 1, 3, 10, 4, 8}, {2, 5, 6, 7, 9, 1, 8, 3, 10, 4}, {2, 5, 6, 8, 1, 4, 9, 10, 3, 7},
{2, 5, 6, 8, 1, 4, 10, 9, 3, 7}, {2, 5, 6, 8, 1, 10, 3, 9, 4, 7}, {2, 5, 6, 8, 1, 10, 4, 9, 3, 7},
{2, 5, 6, 8, 7, 1, 10, 9, 4, 3}, {2, 5, 6, 8, 9, 1, 10, 3, 4, 7}, {2, 5, 6, 8, 9, 1, 10, 4, 3, 7},
{2, 5, 6, 8, 10, 1, 4, 9, 3, 7}, {2, 5, 7, 6, 1, 8, 9, 10, 4, 3}, {2, 5, 7, 6, 9, 1, 3, 10, 4, 8},
{2, 5, 7, 6, 9, 1, 8, 3, 10, 4}, {2, 5, 7, 8, 1, 4, 9, 10, 6, 3}, {2, 5, 7, 8, 1, 9, 4, 3, 10, 6},
{2, 5, 7, 8, 1, 9, 10, 3, 4, 6}, {2, 5, 7, 8, 1, 9, 10, 4, 3, 6}, {2, 5, 7, 8, 1, 10, 9, 3, 4, 6},
{2, 5, 7, 8, 1, 10, 9, 4, 3, 6}, {2, 5, 7, 9, 1, 8, 3, 10, 4, 6}, {2, 5, 7, 9, 1, 8, 4, 10, 6, 3},
{2, 5, 7, 9, 1, 8, 10, 4, 6, 3}, {2, 5, 8, 6, 1, 4, 9, 10, 3, 7}, {2, 5, 8, 6, 1, 4, 10, 9, 3, 7},
{2, 5, 8, 7, 1, 4, 9, 10, 6, 3}, {2, 5, 8, 7, 1, 9, 4, 3, 10, 6}, {2, 5, 8, 7, 1, 10, 4, 9, 3, 6},
{2, 5, 8, 9, 6, 1, 10, 4, 3, 7}, {2, 5, 9, 8, 6, 1, 10, 4, 3, 7}, {2, 6, 7, 8, 9, 1, 5, 10, 4, 3},
{2, 6, 7, 8, 9, 1, 10, 3, 4, 5}, {2, 6, 7, 8, 9, 1, 10, 3, 5, 4}, {2, 6, 7, 8, 9, 1, 10, 5, 4, 3},
{2, 6, 7, 8, 9, 10, 1, 3, 4, 5}, {2, 6, 7, 8, 9, 10, 1, 3, 5, 4}, {2, 6, 7, 8, 9, 10, 1, 5, 4, 3},
{2, 6, 7, 8, 10, 9, 1, 4, 3, 5}, {2, 6, 7, 9, 8, 1, 5, 10, 4, 3}, {2, 6, 7, 9, 8, 1, 10, 5, 4, 3},
{2, 6, 7, 9, 8, 10, 1, 5, 4, 3}, {2, 6, 8, 7, 9, 1, 4, 10, 5, 3}, {2, 6, 8, 7, 9, 1, 10, 3, 5, 4},
{2, 6, 8, 9, 7, 1, 5, 10, 3, 4}, {2, 6, 8, 9, 7, 1, 5, 10, 4, 3}, {2, 6, 8, 9, 10, 7, 1, 5, 3, 4},
{2, 6, 8, 9, 10, 7, 1, 5, 4, 3}, {2, 6, 8, 10, 9, 1, 4, 3, 5, 7}, {2, 6, 9, 10, 7, 8, 1, 5, 4, 3},
{2, 6, 9, 10, 8, 7, 1, 5, 4, 3}, {2, 6, 10, 9, 8, 7, 1, 5, 4, 3}, {3, 4, 5, 7, 8, 9, 10, 1, 2, 6},
{3, 4, 5, 7, 8, 10, 9, 1, 2, 6}, {3, 4, 6, 1, 8, 2, 9, 10, 5, 7}, {3, 4, 7, 1, 8, 9, 2, 10, 5, 6},
{3, 4, 7, 1, 8, 9, 2, 10, 6, 5}, {3, 4, 7, 1, 8, 9, 10, 2, 5, 6}, {3, 4, 7, 1, 8, 9, 10, 2, 6, 5},
{3, 4, 7, 8, 2, 9, 10, 1, 5, 6}, {3, 4, 7, 8, 2, 9, 10, 1, 6, 5}, {3, 4, 7, 9, 8, 2, 10, 5, 1, 6},
{3, 4, 8, 7, 2, 9, 10, 1, 5, 6}, {3, 4, 8, 7, 2, 9, 10, 1, 6, 5}, {3, 4, 9, 7, 8, 2, 10, 1, 5, 6},
{3, 5, 6, 7, 9, 8, 10, 1, 2, 4}, {3, 5, 6, 10, 9, 8, 4, 1, 2, 7}, {3, 5, 7, 1, 8, 9, 10, 4, 2, 6},
{3, 5, 7, 1, 8, 10, 9, 4, 2, 6}, {3, 5, 7, 8, 1, 9, 2, 10, 4, 6}, {3, 5, 7, 8, 2, 9, 1, 10, 6, 4},
{3, 5, 7, 8, 9, 2, 10, 1, 4, 6}, {3, 5, 7, 9, 1, 8, 10, 2, 4, 6}, {3, 5, 7, 9, 8, 2, 10, 1, 4, 6},
{3, 5, 8, 7, 1, 9, 4, 10, 2, 6}, {3, 5, 8, 7, 9, 2, 10, 1, 6, 4}, {3, 5, 8, 10, 7, 1, 9, 2, 4, 6},
{3, 5, 8, 10, 7, 2, 9, 1, 6, 4}, {3, 5, 9, 6, 2, 8, 10, 4, 1, 7}, {3, 5, 9, 7, 1, 8, 10, 4, 2, 6},
{3, 5, 9, 7, 8, 2, 10, 4, 1, 6}, {3, 5, 9, 8, 7, 2, 10, 4, 1, 6}, {3, 5, 10, 7, 8, 9, 2, 4, 1, 6},
{3, 6, 7, 8, 9, 1, 2, 10, 4, 5}, {3, 6, 7, 8, 9, 1, 2, 10, 5, 4}, {3, 6, 7, 8, 9, 2, 1, 10, 5, 4},
{3, 6, 7, 8, 9, 10, 2, 1, 4, 5}, {3, 6, 7, 8, 9, 10, 2, 1, 5, 4}, {3, 6, 7, 9, 10, 8, 2, 1, 5, 4},
{3, 6, 7, 10, 9, 8, 2, 1, 5, 4}, {3, 7, 6, 8, 9, 10, 2, 1, 4, 5}, {3, 7, 6, 8, 9, 10, 2, 1, 5, 4},
{3, 7, 6, 9, 10, 8, 2, 1, 5, 4}, {3, 7, 6, 10, 9, 8, 2, 1, 5, 4}, {3, 8, 6, 7, 9, 2, 10, 1, 4, 5},
{3, 8, 6, 7, 9, 2, 10, 1, 5, 4}, {3, 8, 9, 7, 1, 2, 10, 4, 5, 6}, {4, 5, 6, 7, 8, 9, 10, 1, 2, 3},
{4, 5, 7, 8, 1, 9, 10, 3, 2, 6}, {4, 5, 8, 7, 1, 9, 10, 3, 2, 6}, {2, 4, 5, −7, 1, −8, −9, −3, −10, −6},
{2, 4, 5, −7, 1, 8, 9, −3, 10, 6}, {2, 4, 5, −7, 1, −9, −3, −10, −6, −8},
{2, 4, −6, 1, −7, −9, −3, −10, −5, −8}, {2, 4, −6, 1, 7, 9, −3, 10, 5, 8},
{2, 4, 6, 1, −7, −9, 3, −10, −5, −8}, {2, 4, −6, 1, −8, −3, −9, −5, −10, −7},
{2, 4, −6, 1, 8, −3, 9, 5, 10, 7}, {2, 4, 6, 1, −8, 3, −9, −5, −10, −7},
{2, 4, −6, 1, −8, −3, −9, −10, −5, −7}, {2, 4, −6, 1, 8, −3, 9, 10, 5, 7},
{2, 4, 6, 1, −8, 3, −9, −10, −5, −7}, {2, 4, −6, 1, −8, −3, −10, −9, −5, −7},
{2, 4, −6, −9, 1, −8, −10, −3, −5, −7}, {2, 4, −7, 1, −8, −9, −3, −10, −5, −6},
{2, 4, −7, 1, 8, 9, −3, 10, 5, 6}, {2, 4, 7, 1, −8, −9, 3, −10, −5, −6},

```
{2, 4, -7, 1, -8, -9, -3, -10, -6, -5}, {2, 4, -7, 1, 8, 9, -3, 10, 6, 5},
{2, 4, 7, 1, -8, -9, 3, -10, -6, -5}, {2, 4, -7, 1, -8, -9, -10, -3, -5, -6},
{2, 4, -7, 1, -8, -9, -10, -3, -6, -5}, {2, 4, -9, -6, 1, -8, -10, -3, -5, -7},
{2, 5, -7, 6, 1, 8, 9, -10, 4, -3}, {2, 5, -7, -8, 1, -9, -4, -3, -10, -6},
{2, 5, -7, -8, 1, 9, -4, -3, 10, 6}, {2, 5, -7, -8, 1, -9, -10, -3, -4, -6},
{2, 5, -7, -8, 1, -9, -10, -4, -3, -6}, {2, 5, -7, -8, 1, 9, 10, -4, -3, 6},
{2, 5, 7, 8, 1, -9, -10, 4, 3, -6}, {2, 6, -8, -7, -9, 1, -4, -10, -5, -3},
{2, 6, -8, 7, -9, 1, 4, -10, -5, -3}, {2, 6, 8, -7, 9, 1, -4, 10, 5, 3}, {3, 4, 5, 7, 8, -9, -10, 1, 2, -6},
{3, 4, 5, 7, 8, -10, -9, 1, 2, -6}, {3, 4, 6, 1, -8, 2, -9, -10, -5, -7}, {3, 4, 7, 9, 8, 2, -10, 5, 1, -6},
{3, -5, -6, 7, -9, -8, 10, -1, -2, -4}, {3, 5, 7, 8, 9, 2, -10, 1, 4, -6}, {3, 5, 7, 9, 8, 2, -10, 1, 4, -6},
{3, -5, -8, 7, -1, -9, 4, 10, -2, -6}, {3, -5, -9, 7, -1, -8, 10, 4, -2, 6}}}}
```