```
(* ED Program for the u`Hubbard model on  M-site chain *)
Clear["
Global`*"];
```

```
M = 4 ; (* M dimension of the spatial lattice *)
v = {{1, 1, 0, 0}, {1, 1, 0, 0}};
FullForm[v];
MP1 = Length[Table[Permutations[v[[1]]]]];
MP2 = Length[Table[Permutations[v[[2]]]]];

index = Flatten[Table[{Permutations[v[[1]]][[i]], Permutations[v[[2]]][[j]]},
    {i, MP1}, {j, MP2}], 1];

plus[k_, sigma_][arg_] := ReplacePart[arg, 1, {sigma, k}];
minus[k_, sigma_][arg_] := ReplacePart[arg, 0, {sigma, k}];

signn[1, 1][arg_] := 1
signn[k_, 1][arg_] := (-1)^Sum[arg[[1, i]], {i, 1, k - 1}]
signn[1, 2][arg_] :=  (-1)^Sum[arg[[1, i]], {i, 1, M}]
signn[k_, 2][arg_] :=
 (-1)^(Sum[arg[[2, i]], {i, 1, k - 1}] + Sum[arg[[1, i]], {i, 1, M}])
```

```
index;
FullForm[index];
TreeForm[v];
```

```
c[k_, sigma_][0] := 0;
c[k_, sigma_][factor_. * s[arg_]] :=
 factor * (arg[[sigma, k]]) * signn[k, sigma][arg] * s[minus[k, sigma][arg]]
cdg[k_, sigma_][0] := 0;
cdg[k_, sigma_][factor_. * s[arg_]] :=
  factor * (1 - arg[[sigma, k]]) * signn[k, sigma][arg] * s[plus[k, sigma][arg]];

c[M + 1, sigma_] := c[1, sigma]
cdg[M + 1, sigma_] := cdg[1, sigma]

c[M + 2, sigma_] := c[2, sigma]
cdg[M + 2, sigma_] := cdg[2, sigma]

n[k_, sigma_][0] := 0
n[k_, sigma_][factor_. * s[arg_]] := factor * arg[[sigma, k]] * s[arg]

scalarp[a_, 0] := 0;
scalarp[a_, b_ + c_] := scalarp[a, b] + scalarp[a, c];
scalarp[factor2_. * s[arg1_], factor_. * s[arg2_]] :=
  factor2 * factor * If[arg1 == arg2, 1, 0];
```

```
(* The Hamiltonian *)
H[vector_] := Expand[      -t * Sum[cdg[k, sigma][ c[k + 1, sigma][vector] ] +
       cdg[k + 1, sigma][ c[k, sigma][vector] ],
                      {k, M}, {sigma, 2}] +
                    u * Sum[n[k, 1][n[k, 2][vector]], {k, 1, M}]];
```

```
MaxL = Length[index];
h = (hlist = Table[H[s[index[ [j]] ]], {j, MaxL}];
   Table[scalarp[ s[ index[[i]] ], hlist[[j]] ], {i, MaxL}, {j, MaxL}]);
 g[uu_] := Sort[Thread[ Eigensystem[ N[ h /. {t → 1.0, u → uu } ]]]] [[1, 2]]
gs[uu_] := Sort[Thread[ Eigensystem[ N[ h /. {t → 1.0, u → uu } ]]]] ;
 gt[tt_, uu_] := Sort[Thread[ Eigensystem[ N[ h /. {t → tt, u → uu } ]]]] [[1, 2]];
```

```
g[1.];
```

```
(* double occupancy *)

  dd[u_] :=
   Sum[Sum[Abs[g[u][[j]]]^2 * index[[j, 1, k]] * index[[j, 2, k]], {j, MaxL}],
     {k, M}] / M;
```

```
MatrixForm[h];
MatrixForm[gs[0]];
g[0];
Eigenvalues[h];
```