/*

1. Generarea combinarilor pentru multimi standard */ #include using namespace std;

void afisare(int c[], int n) { for(int i = 1; i <= n; i++) //printf("%d \n",p[i]); cout << c[i] << ' '; cout << endl; return; }

void combinari(int m,int n) { int c[100]; int i, j, k; for(i = 1; i <= n; i++) c[i] = i; afisare(c,n); do { k = n; while(c[k] == m-n+k && k > 0) k--; if(k > 0) { c[k] = c[k] + 1; for(i = k+1; i <= n; i++) c[i] = c[i-1] + 1; afisare(c,n); } } while(k > 0); } int main() { int m, n; cin >> m >> n; combinari(m,n); return 0; } /* 2. Generarea combinarilor pentru multimi standard */

#include

using namespace std;

void afisare(int c[], int a[], int n) { for(int i = 1; i <= n; i++) cout << a[c[i]] << ' '; cout << '\n'; return; }

void combinari(int a[], int m, int n) { int c[100], i, j, k; for(i = 1; i <= n; i++) c[i] = i; afisare(c, a, n); do { k = n; while(c[k] == m-n+k && k > 0) k--; if(k > 0) { c[k]++; for(i = k + 1; i <= n; i++) c[i] = c[i-1] + 1; afisare(c, a, n); } } while(k > 0); }

int main() { int a[101], m, n; cin >> m >> n; for(int i = 1; i <= m; i++) cin >> a[i]; //multimea combinari(a, m, n); return 0; } /* 3. Generarea combinarilor cu repetite pentru multimi standard */

#include

using namespace std;

void afisare(int c[], int n) { for(int i = 1; i <= n; i++) cout << c[i] << ' '; cout << '\n'; return; }

void combinari_cu_repetitie(int m, int n) { int c[100], i, j, k; for(i = 1; i <= n; i++) c[i] = 1; afisare(c, n); do { k = n; while(c[k] == m && k > 0) k--; if(k > 0) { c[k]++; for(i = k + 1; i <= n; i++) c[i] = c[k]; afisare(c, n); } } while(k > 0); }

int main() { int a[101], m, n; cin >> m >> n; combinari_cu_repetitie(m, n); return 0; } /* 4. Generarea combinarilor cu repetite pentru multimi oarecare */

#include

using namespace std;

void afisare(int c[], int a[], int n) { for(int i = 1; i <= n; i++) cout << a[c[i]] << ' '; cout << '\n'; }

void combinari_cu_repetitie(int a[], int m, int n) { int c[100], i, j, k; for(i = 1; i <= n; i++) c[i] = 1; afisare(c, a, n); do { k = n; while(c[k] == m && k > 0) k--; if(k > 0) { c[k]++; for(i = k + 1; i <= n; i++) c[i] = c[k]; afisare(c, a, n); } } while(k > 0); }

int main() { int a[101], m, n; cin >> m >> n; for(int i = 1; i <= m; i++) cin >> a[i]; //multimea combinari_cu_repetitie(a, m, n); return 0; } /* 5. Generarea permutarilor pentru multimi standard */

#include

```
using namespace std;

void afisare(int p[], int n) { for(int i = 1; i <= n; i++) cout << p[i] << ' '; cout << '\n'; }

void permutari(int n) { int p[101], i, j, k; for(i = 1; i <= n; i++) p[i] = i; afisare(p, n); do { k = n - 1; while(p[k] >= p[k+1] && k > 0) k--; if(k > 0) { j = n; while(p[j] <= p[k]) j--; swap(p[k],p[j]); for(i = 1; i <= (n-k)/2; i++) swap(p[k+i],p[n-i+1]); afisare(p, n); } } while(k > 0); }

int main() { int n; cin >> n; permutari(n); return 0; } /* 6. Generarea permutarilor pentru multimi oarecare */

#include

using namespace std;

void afisare(int p[], int a[], int n) { for(int i = 1; i <= n; i++) cout << a[p[i]] << ' '; cout << '\n'; }

void permutari(int a[], int n) { int p[101], i, j, k; for(i = 1; i <= n; i++) p[i] = i; afisare(p, a, n); do { k = n - 1; while(p[k] >= p[k+1] && k > 0) k--; if(k > 0) { j = n; while(p[j] <= p[k]) j--; swap(p[k], p[j]); for(i = 1; i <= (n-k)/2; i++) swap(p[k+i], p[n-i+1]); afisare(p, a, n); } } while(k > 0); }

int main() { int a[101], n; cin >> n; for(int i = 1; i <= n; i++) cin >> a[i]; permutari(a, n); return 0; } /* 7. Generarea aranjamentelor pentru multimi standard */

#include

using namespace std;

void afisare(int c[], int p[], int n) { for(int i = 1; i <= n; i++) cout << c[p[i]] << ' '; cout << '\n'; }

void permutari(int c[], int n) { int p[101], i, j, k; for(i = 1; i <= n; i++) p[i] = i; afisare(c, p, n); do { k = n - 1; while(p[k] >= p[k+1] && k > 0) k--; if(k > 0) { j = n; while(p[j] <= p[k]) j--; swap(p[k], p[j]); for(i = 1; i <= (n-k)/2; i++) swap(p[k+i], p[n-i+1]); afisare(c, p, n); } } while(k > 0); }

void aranjamente(int m, int n) { int a[101], i, j, k; for(i = 1; i <= n; i++) a[i] = i; permutari(a, n); do { k = n; while(a[k] == m-n+k && k > 0) k--; if(k > 0) { a[k]++; for(i = k + 1; i <= n; i++) a[i] = a[i-1] + 1; permutari(a, n); } } while(k > 0); }

int main() { int m, n; cin >> m >> n; aranjamente(m, n); return 0; } /* 7. Generarea aranjamentelor pentru multimi oarecare */

#include

using namespace std;

void afisare(int c[], int p[], int n) { for(int i = 1; i <= n; i++) cout << c[p[i]] << ' '; cout << '\n'; }

void permutari(int a[], int n) { int p[101], i, j, k; for(i = 1; i <= n; i++) p[i] = i; afisare(a, p, n); do { k = n - 1; while(p[k] >= p[k+1] && k > 0) k--; if(k > 0) { j = n; while(p[j] <= p[k]) j--; swap(p[k], p[j]); for(i = 1; i <= (n-k)/2; i++) swap(p[k+i], p[n-i+1]); afisare(a, p, n); } } while(k > 0); }

void aranjamente(int c[], int m, int n) { int a[101], i, j, k; for(i = 1; i <= n; i++) a[i] = i; do { int b[101]; for (i = 1; i <= n; i++) b[i] = c[a[i]]; permutari(b, n); k = n; while(a[k] == m-n+k && k > 0) k--; if(k > 0) { a[k]++; for(i = k + 1; i <= n; i++) a[i] = a[i-1] + 1; } } while(k > 0); }
```

```
int main() { int a[101], m, n; cin >> m >> n; for(int i = 1; i <= m; i++) cin >> a[i]; aranjamente(a, m, n); return 0; } /* 5. Generarea permutarilor cu repetitie pentru multimi standard */

#include

using namespace std;

void afisare(int p[], int n) { for(int i = 1; i <= n; i++) cout << p[i] << ' '; cout << '\n'; }

void permutari_cu_repetitie(int n, int t[], int m) { int p[101], i, j, k; n = 0; for(i = 1; i <= m; i++) n += t[i]; i = 0; for(j = 1; j <= m; j++){ for(k = 1; k <= t[j]; k++) { i++; p[i] = j; } } afisare(p, n); do { k = n - 1; while(p[k] >= p[k+1] && k > 0) k--; if(k > 0) { j = n; while(p[j] <= p[k]) j--; swap(p[k], p[j]); for(i = 1; i <= (n-k)/2; i++) swap(p[k+i], p[n-i+1]); afisare(p, n); } } while(k > 0); }

int main() { int t[101], n, m; cin >> n >> m; for(int i = 1; i <= m; i++) cin >> t[i]; permutari_cu_repetitie(n, t, m); return 0; } /* 5. Generarea permutarilor cu repetitie pentru multimi oarecare */

#include

using namespace std;

void afisare(int p[],int a[], int n) { for(int i = 1; i <= n; i++) cout << a[p[i]] << ' '; cout << '\n'; }

void permutari_cu_repetitie(int a[], int n, int t[], int m) { int p[101], i, j, k; n = 0; for(i = 1; i <= m; i++) n += t[i]; i = 0; for(j = 1; j <= m; j++){ for(k = 1; k <= t[j]; k++) { i++; p[i] = j; } } afisare(p, a, n); do { k = n - 1; while(p[k] >= p[k+1] && k > 0) k--; if(k > 0) { j = n; while(p[j] <= p[k]) j--; swap(p[k], p[j]); for(i = 1; i <= (n-k)/2; i++) swap(p[k+i], p[n-i+1]); afisare(p, a, n); } } while(k > 0); }

int main() { int a[101], t[101], n, m; cin >> n; // Nr de elem for(int i = 1; i <= n; i++) cin >> a[i]; // Elementele cin >> m; // Nr de repetitii for(int i = 1; i <= m; i++) cin >> t[i]; // Repetitiile permutari_cu_repetitie(a, n, t, m); return 0; }
```