void divide\_impera(int\* a, int\* b, int la, int ha, int lb, int hb)

{

// LUNGIMI DIFERITE

// a -> sirul de lungime mai mica

int mid1 = (la + ha) / 2;

int mid2 = (lb + hb) / 2;

int rest1 = (la + ha) % 2;

int rest2 = (lb + hb) % 2;

printf("%d %d %d %d---> %d %d ", la, ha, lb, hb, mid1, mid2);

int new\_mid1 = mid1;

int new\_mid2 = mid2;

if (a[mid1] == b[mid2])

{

//oprire

printf("mijloc: %d\n", a[mid1]);

return;

}

else if (ha - la == 0)

{

if (a[la] < b[lb])

printf("mijloc: %d", a[la]);

else

{

printf("mijloc: %d", b[lb]);

}

}

else if (a[mid1] < b[mid2])

{

printf("caz mai mic\n");

// la numar impar de elemente voi lua inclusiv mijlocul in calculul subproblemei

// la numar par de elemente voi lua cel mai din dreapta mijloc

if (rest1 != 0) //nr par de elem

{

new\_mid1++;

}

int new\_lb = new\_mid2 - (ha - new\_mid1);

divide\_impera(a, b, new\_mid1, ha, new\_lb, new\_mid2);

}

else if (a[mid1] > b[mid2])

{

printf("caz mai mare\n");

if (rest2 != 0) // nr par de elemente

{

new\_mid2++;

}

int new\_hb = new\_mid2 + (new\_mid1 - la);

divide\_impera(a, b, la, new\_mid1, new\_mid2, new\_hb);

}

}

int divide\_conquer\_siruri\_egale(int\* a, int\* b, int la, int ha, int lb, int hb)

{

// DIMENISUNI EGALE

int mid1 = (la + ha) / 2;

int mid2 = (lb + hb) / 2;

int rest1 = (la + ha) % 2;

int rest2 = (lb + hb) % 2;

printf("%d %d %d %d---> %d %d ", la, ha, lb, hb, mid1, mid2);

int new\_mid1 = mid1, new\_mid2 = mid2;

if (a[mid1] == b[mid2])

{

//oprire

printf("mijloc: %d\n", a[mid1]);

return;

}

else if (ha - la == 1)

{

int\* aux = (int\*)malloc(4 \* sizeof(int));

aux[0] = a[la]; aux[1] = a[ha];

aux[2] = b[lb]; aux[3] = b[hb];

for (int i = 0; i < 3; i++)

{

for (int j = i + 1; j < 4; j++)

{

if (aux[i] > aux[j])

{

int x = aux[i];

aux[i] = aux[j];

aux[j] = x;

}

}

}

// mijloace: aux[1] si aux[2]

printf("mijloc: %d %d\n", aux[1], aux[2]);

return;

}

else if (a[mid1] < b[mid2])

{

printf("caz mai mic\n");

// la numar impar de elemente voi lua inclusiv mijlocul

// la numar par de elemente voi lua cel mai din dreapta mijloc

new\_mid1 = mid1;

new\_mid2 = mid2;

if (rest1 != 0) //nr par de elem

{

new\_mid1++;

}

divide\_conquer\_siruri\_egale(a, b, new\_mid1, ha, lb, new\_mid2);

}

else if (a[mid1] > b[mid2])

{

printf("caz mai mare\n");

if (rest2 != 0) // nr par de elemente

{

new\_mid2++;

}

divide\_conquer\_siruri\_egale(a, b, la, new\_mid1, new\_mid2, hb);

}

}