

# Programarea Algoritmilor

## – SEMINAR NR. 1 –

### (grupele 132-134, 141-144)

1. Se citește un număr natural  $n$  și un vector de  $n$  elemente. Să se verifice dacă vectorul are toate elementele distincte.

```
# Solutia 1 ---> O(n*n)
n = int(input("n = "))
v = [0] * n
for i in range(n):
    v[i] = int(input(f"nr_{i+1}: "))
ok = True
for i in range(n-1): # O(n*n) pt verificare
    for x in v[i+1:]:
        if v[i] == x:
            ok = False
if ok == True:
    print("Toate elementele sunt distincte.")
else:
    print("Nu toate elementele sunt distincte.")

# Solutia 2 ---> O(n * log(n))
n = int(input("n = "))
v = [0] * n
for i in range(n):
    v[i] = int(input(f"nr_{i+1}: "))
ok = True
v.sort() # sortat crescator, in place -> O(n*log(n))
for i in range(n-1): # O(n) pt verificare
    if v[i] == v[i+1]:
        ok = False
if ok == True:
    print("Toate elementele sunt distincte.")
else:
    print("Nu toate elementele sunt distincte.")

# Solutia 3 ---> O(n)
n = int(input("n = "))
v = [0] * n
for i in range(n):
    v[i] = int(input(f"nr_{i+1}: "))
s = set(v) # transformam lista in set => se elimina duplicatele
if len(s) == n:
    print("Toate elementele sunt distincte.")
else:
    print("Nu toate elementele sunt distincte.")
```

2. Se citește un vector sortat de numere întregi distincte și un număr natural  $p$ . Să se găsească toate perechile de 2 numere care au suma  $p$ .

```
# Solutia 1 ---> O(n*n)
# solutia banala, cu 2 for-uri

# Solutia 2 ---> O(n*log(n))
# Pentru fiecare nr x din vector, se cauta binar nr y = p-x.
```

```

# Solutia 3 ---> O(n) daca vectorul se citește deja sortat
n = int(input("n = "))
v = [0] * n
for i in range(n):
    v[i] = int(input(f"nr_{i+1}: "))
# v.sort() ---> O(n*log(n)) daca vectorul nu se citește sortat
p = int(input("\np = "))
print("\nPerechile de numere care au suma", p)
# pornim cu i si j din capetele opuse ale vectorului
i = 0
j = n-1
while i < j:
    if v[i] + v[j] == p:
        print(v[i], ",", v[j])
        i = i + 1
        j = j - 1
    elif v[i] + v[j] > p:
        j = j - 1
    else:
        i = i + 1

```

3. **TEMĂ:** Se dă un vector nesortat de numere întregi distincte. Să se găsească tripletele de numere care au suma 0.