MUSTAQIL ISHLASH UCHUN MASALALAR YECHIMLARI

1-masala. Yechim (prefix sum + dict)

```
n = int(input())
a = list(map(int, input().split()))

pos = {0: -1}
s = 0
ans = 0

for i, x in enumerate(a):
    s += x
    if s in pos:
        ans = max(ans, i - pos[s])
    else:
        pos[s] = i

print(ans)
```

2-masala. Yechim (sort + two pointers)

```
n, S = map(int, input().split())
a = list(map(int, input().split()))

a.sort()
ok = False
for i in range(n):
    l, r = i + 1, n - 1
    while l < r:
        s = a[i] + a[l] + a[r]
        if s == S:
            ok = True
            break
        if s < S:
            l += 1
        else:
            r -= 1
    if ok:
        break

print("YES" if ok else "NO")
```

3-masala. Yechim (binary search + greedy)

```
def can(mx):
    cur = 0
    cnt = 1
    for x in a:
        if x > mx:
            return False
        if cur + x <= mx:
            cur += x
```

```
        else:
            cnt += 1
            cur = x
    return cnt <= K

n, K = map(int, input().split())
a = list(map(int, input().split()))

lo, hi = max(a), sum(a)
ans = hi

while lo <= hi:
    mid = (lo + hi) // 2
    if can(mid):
        ans = mid
        hi = mid - 1
    else:
        lo = mid + 1

print(ans)
```

4-masala. Yechim (chiziqli tahlil)

```
n = int(input())
a = list(map(int, input().split()))

ans = 1
i = 0
while i < n - 1:
    j = i
    while j + 1 < n and a[j + 1] > a[j]:
        j += 1
    k = j
    while k + 1 < n and a[k + 1] < a[k]:
        k += 1
    if j > i and k > j:
        ans = max(ans, k - i + 1)
    i = max(k, i + 1)

print(ans)
```

5-masala. Yechim (sliding window)

```
n, K = map(int, input().split())
a = list(map(int, input().split()))

l = 0
s = 0
ans = 0

for r in range(n):
    s += a[r]
    while s > K:
```

```
        s -= a[l]
        l += 1
    ans = max(ans, r - l + 1)

print(ans)
```

6-masala. Yechim (BFS × 2)

```
from collections import deque

n, m = map(int, input().split())
g = [[] for _ in range(n + 1)]
for _ in range(m):
    u, v = map(int, input().split())
    g[u].append(v)
    g[v].append(u)

s1, s2 = map(int, input().split())

def bfs(src):
    dist = [-1] * (n + 1)
    dq = deque([src])
    dist[src] = 0
    while dq:
        u = dq.popleft()
        for v in g[u]:
            if dist[v] == -1:
                dist[v] = dist[u] + 1
                dq.append(v)
    return dist

d1 = bfs(s1)
d2 = bfs(s2)

ans = -1
best = 10**18
for i in range(1, n + 1):
    if d1[i] != -1 and d2[i] != -1:
        mx = max(d1[i], d2[i])
        if mx < best:
            best = mx
            ans = mx

print(ans)
```

7-masala. Yechim (prefix mod + dict)

```
n, K = map(int, input().split())
a = list(map(int, input().split()))

first = {0: -1}
s = 0
ans = 0
```

```
for i, x in enumerate(a):
    s = (s + x) % K
    if s in first:
        ans = max(ans, i - first[s])
    else:
        first[s] = i

print(ans)
```

8-masala. Yechim (sliding window + dict)

```
n = int(input())
a = list(map(int, input().split()))

need = len(set(a))
cnt = {}
l = 0
have = 0
ans = n + 1

for r in range(n):
    cnt[a[r]] = cnt.get(a[r], 0) + 1
    if cnt[a[r]] == 1:
        have += 1
    while have == need:
        ans = min(ans, r - l + 1)
        cnt[a[l]] -= 1
        if cnt[a[l]] == 0:
            have -= 1
        l += 1

print(ans if ans <= n else 0)
```

9-masala. Yechim (2 ta heap)

```
import heapq

n, k = map(int, input().split())
a = list(map(int, input().split()))

low = []   # max-heap (manfiy)
high = []  # min-heap

def balance():
    if len(low) > len(high) + 1:
        heapq.heappush(high, -heapq.heappop(low))
    if len(high) > len(low):
        heapq.heappush(low, -heapq.heappop(high))

res = []
for i in range(n):
    if not low or a[i] <= -low[0]:
```

```
            heapq.heappush(low, -a[i])
        else:
            heapq.heappush(high, a[i])
        balance()

        if i >= k:
            if a[i - k] <= -low[0]:
                low.remove(-a[i - k])
                heapq.heapify(low)
            else:
                high.remove(a[i - k])
                heapq.heapify(high)
            balance()

        if i >= k - 1:
            res.append(-low[0])

print(*res)
```

10-masala. Yechim (stack)

```
s = input()

st = [-1]
ans = 0

for i, ch in enumerate(s):
    if ch == '(':
        st.append(i)
    else:
        st.pop()
        if not st:
            st.append(i)
        else:
            ans = max(ans, i - st[-1])

print(ans)
```