* **Nộp bài chấm:**

1. Tên tệp input, output thật chính xác, không để đường dẫn tuyệt đối.
2. Những lệnh in ra để quan sát kết quả khi code phải xoá sạch tuyệt đối.
3. Quy cách ghi dữ liệu ra file output, tuyệt đối giống như yêu cầu của đề bài.

* **Tính trọng lượng của dãy**

Mảng a

Tính F, với Fi = F1+F2+…+Fi

Fi gọi là trọng lượng của dãy con i

Tính trọng lượng của các dãy con trong mảng.

**Bài 1. Dãy con tổng bằng 0 DAY0**

Cho dãy số nguyên a = (a1, a2, …, an),1≤n≤10000,-10000≤ai≤10000.

Hãy tìm một dãy con dài nhất gồm các phần tử liên tiếp trong dãy a có tổng bằng 0.

***Dữ liệu vào:*** đọc từ tệp **DAY0.INP:**

* Dòng đầu tiên chứa số n.
* Dòng thứ hai chứa n số a1, a2, …, an cách nhau ít nhất 1 dấu cách.

Lưu ý: Dữ liệu luôn hợp lí để tồn tại một đoạn các phần tử liên tiếp có tổng bằng 0.

***Dữ liệu ra:*** ghi ra tệp **DAY0.OUT** hai số là số thứ tự của phần tử đầu tiên và phần tử cuối cùng của dãy liên tiếp có tổng bằng 0. Nếu có nhiều đoạn, xuất đoạn đầu tiên tìm được.

***Ví dụ:***

|  |  |
| --- | --- |
| **DAY0.INP** | **DAY0.OUT** |
| 9  2 7 5 -3 -2 4 -9 -2 -1 | 2 8 |

**Hướng dẫn:**

-B1: Đọc dữ liệu từ tệp DAY0.INP lưu vào biến n và mảng a.

Đặt s[i] = a[1] + a[2] + … + a[i] = s[i-1] + a[i]

Theo yêu cầu đề bài, ta có 0 = a[i] + … + a[j] = s[j] – s[i-1] => s[j] = s[i-1]

Bài toán trở thành: tìm 2 phần tử có cùng giá trị mà **xa nhau nhất** trong dãy s[0],…,s[n].

-B2: Tạo dãy s: s[i] = a[1] + a[2] + … + a[i]

-B3: Tạo dãy id: id[i] = i, vị trí ban đầu của các phần tử trong dãy s.

-B4: Sắp xếp dãy s có giá trị tăng dần, kèm vị trí, nếu giá trị s bằng nhau thì sắp tăng dần theo vị trí id.

-B5: Tìm đoạn có giá trị bằng nhau dài nhất s[i], s[i+1], …,s[j], id[i]+1 chính là phần tử bắt đầu, id[j] là phần tử kết thúc của đoạn con cần tìm và ***id[j]-id[i]*** chính là độ dài của dãy có tổng bằng 0.

-B6: Ghi ra tệp DAY0.OUT giá trị id[i]+1và id[j].

**Code:**

-B1: Đọc dữ liệu từ tệp DAY0.INP lưu vào biến n và mảng a.

fin = open("DAY0.INP","r")

fout = open("DAY0.OUT","w")

li = fin.readline()

n = int(li)

a = []

li = fin.readline()

ls = li.split()

for i in range(n):

a.append(int(ls[i]))

-B2: Tạo dãy s: s[i] = a[1] + a[2] + … + a[i]

-B3: Tạo dãy id: id[i] = i, vị trí ban đầu của các phần tử trong dãy s.

s = []

s.append(a[0])

id = []

id.append(0)

for i in range(1,n):

s.append(s[i-1]+a[i])

id.append(i)

-B4: Sắp xếp dãy s có giá trị tăng dần, kèm vị trí, nếu giá trị s bằng nhau thì sắp tăng dần theo vị trí id.

def qs (d,c):

i = d

j = c

idg = id[(i+j)//2]

m = s[(i+j)//2]

while i <= j:

while (s[i] < m) or ((s[i] == m) and (id[i] < idg)):

i = i + 1

while (s[j] > m) or ((s[j] == m) and (id[j] > idg)):

j = j - 1

if i <= j:

s[i],s[j] = s[j],s[i]

id[i],id[j] = id[j],id[i]

i = i + 1

j = j - 1

if d < j:

qs(d,j)

if i < c:

qs(i,c)

qs(0,n-1)

-B5: Tìm đoạn có giá trị bằng nhau dài nhất s[i], s[i+1], …,s[j], id[i]+1 chính là phần tử bắt đầu, id[j] là phần tử kết thúc của đoạn con cần tìm và ***id[j]-id[i]*** chính là độ dài của dãy có tổng bằng 0.

maxd = 0

maxi = -1

maxj = -1

for i in range(1,n):

if (s[i] == s[i-1]) and (id[i]-id[i-1] > maxd):

maxd = id[i]-id[i-1]

maxi = id[i-1] + 2

maxj = id[i] + 1

-B6: Ghi ra tệp DAY0.OUT giá trị id[i]+1và id[j].

print(maxi,maxj,file = fout)

fin.close()

fout.close()

**Bài 2. Dãy con liên tiếp dài nhất có tổng >= p MAXSEQ**

Cho một dãy số nguyên a1, a2,…,an. Dãy số ai, ai+1, …, aj với 1≤i≤j≤n được gọi là dãy con của dãy số đã cho và khi đó, j-i+1 được gọi là độ dài, còn ai + ai+1 + …+ aj được gọi là trọng lượng của dãy con này.

***Yêu cầu:*** cho số nguyên p, trong số các dãy con của dãy số đã cho có trọng lượng không nhỏ hơn p hãy tìm dãy con có độ dài lớn nhất.

***Dữ liệu vào:*** đọc từ tệp **MAXSEQ.INP:**

* Dòng đầu ghi hai số nguyên n và p cách nhau bởi dấu cách.
* Dòng thứ i trong số n dòng tiếp theo chứa số nguyên ai là số hạng thứ i của dãy số đã cho, i=1,2,…,n.

***Dữ liệu ra:*** ghi ra tệp **MAXSEQ.OUT** độ dài của dãy con tìm được (qui ước: nếu không có dãy con nào thoả mãn điều kiện đặt ra thì k=-1).

***Ví dụ:***

|  |  |
| --- | --- |
| **MAXSEQ.INP** | **MAXSEQ.OUT** |
| 5 6  -2  3  2  -2  3 | 4 |
| 4 9  2  3  2  -2 | -1 |

**Hướng dẫn:**

Ta có s[i] là tổng của a[1] + … + a[i].

* Trọng lượng của dãy từ i đến j là s[j] – s[i]
* Tìm **max(j-i)** thoả mãn điều kiện **s[j]-s[i] >= p**

-B1: Đọc dữ liệu từ tệp MAXSEQ.INP

-B2: Tạo dãy s, có s[i] là tổng của a[1] + … + a[i]

-B3: Tìm max(j-i) thoả mãn điều kiện s[j]-s[i] >= p

-B4: Ghi kết quả ra tệp MAXSEQ.OUT

**Code:**

-B1: Đọc dữ liệu từ tệp MAXSEQ.INP

fin = open("MAXSEQ.INP","r")

fout = open("MAXSEQ.OUT","w")

li = fin.readline()

ls = li.split()

n = int(ls[0])

p = int(ls[1])

a = []

for i in range(n):

li = fin.readline()

a.append(int(li))

-B2: Tạo dãy s, có s[i] là tổng của a[1] + … + a[i]

s = []

s.append(a[0])

id = []

id.append(0)

for i in range(1,n):

s.append(s[i-1]+a[i])

id.append(i)

-B3: Tìm max(j-i) thoả mãn điều kiện s[j]-s[i] >= p

kq = -1

for i in range(n):

for j in range(n-1,i+1,-1):

if (s[j] - s[i] >= p) and (j-i > kq):

kq = j-i

-B4: Ghi kết quả ra tệp MAXSEQ.OUT

print(kq,file = fout)

fin.close()

fout.close()