```
Max Sum Subarray
Algorithm
                            E+0 : 142 Krub
   cwir_sum = 0
                      mux of tox cours sum
  max_80-far=a[0]
  Start = 0
  end=0
  temp=0
                   of constant in the stand
  for i in range (0, size):
       Curr_sum = curr_sum + arr [i]
       if (max_so_for < com. sum): we ....
             Start = temp ( De - 100 > 101-02 xin
        end si i true E grow true
             max_so_far=curr_sum
                                               FOR
        if (cwisum colinated pison muz remo
             curr sun =0
             temp=i+1 [wall &- 1 me mue mis
Example: 0
  ample: 0 1 2 3 4 5 6 7 8 9 10 11 12

avr = 4 -3 -2 2 3 1 -2 -3 6 -6 -4 2 1
   Stort = 0 end = 0 :temp=0 i=0
 Step 1:
     curi-sum = 0+4=4 [60th if conditions fail]
Step 2:
   stort=0 end=0 cour_sum=4-3=1 (BICF) (i=1)
   1=2 Stort=0 end=0 cur-sum=1-2=-1=) cur-sum=0
Step3:
                                       temp=3
```

```
Mark Sim Silponen
 Step 4:
                                        Algorithm
       curr_sum = 0+2=2
         max_of_far < curr_sum => 422 => False
                              [0]10 = rut-
        curr_sum 20 => Folse
                                        stort c
 Steps:
                                         0-1000
        cur sum = 2+3=5
    1=4
                                        o dung
        max_so-forcown_sum =>465 => True
              Listart= temp=3 end=i=3 max-90-for=5
              CUM - SUM - CULT- LUN + ONN (i)
 Step 6:
         curr_8007 =5+1=6 1000 > 107-02-1000) 11
    1=5
         max-so-for cour-som = 5 5 < 5 > True
            L> start= temp=3 end=i=5 max_so-far=6
                MIN-80-107- CULY-SUM
 Step 7:
        curr_sum=6-2=4 [BICF] > (1)2_1500) 7
Step8:
                          O= CANS - LUND
    1=7 cwor_sum=4-3=1 (BICF) 1+1 - qrist
Step9:
    max_90_far < curr_sum => 6<7 => True
           L) start = temp = 3 end = 8 = 1 max = 80 - for = 7
step10:
    1=9 cur sum = 7-6=1 [BICF]
Step11:
    1710. jan - 9um 211-4= -3 mus mus o bris o brots
             Ls curr_sum=0 & temp=10+1=11
             man_90_for=7 stort=3 end=8
```

Step 12:

1=11 Cwy_Sum = 0+2=2 (BIC)

Step 13:

1=12 cever_sum = 2+1 [BICF]

Result:

maximum sur of subavay = 7 Subavay indexes [3,8]