Assignment No 1.

```
FIND_MAX_CROSSING_SUBARRAY(A; low; mid; high)
1.      left_sum = -inf
2.      sum = 0
3.      for i = mid downto low
4.      sum = sum + A[i]
5.      if sum > left_sum
6.              left_sum = sum
7.              max_left = i
8.      right_sum = -inf
9.      sum = 0
10.     for j = mid + 1 to high
11.     sum = sum + A[j]
12.     if sum > right_sum
13.             right_sum = sum
14.             max_right = j
15. return (max_left; max_right; left_sum + right_sum)
```

```
FIND_MAXIMUM_SUBARRAY(A; low; high)
1.      if high == low
2.              return (low; high;A[low])
3.      else
4.              mid = (low + high) = 2
5.      (left_low; left_high; left+sum) =    FIND-MAXIMUM-SUBARRAY(A;
low; mid)
6.      (right_low; right_high; right_sum) = FIND-MAXIMUM-SUBARRAY(A;
mid + 1; high)
7.      (cross_low; cross_high; cross_sum) = FIND-
MAX_CROSSING_SUBARRAY(A; low; mid; high)
8.      if left_sum >= right_sum and left_sum >= cross_sum
9.              return (left_low; left_high; left_sum)
10. else if (right_sum >= left_sum and right_sum >= cross_sum)
11.             return (right_low; right_high; right_sum)
12. else
13.             return (cross_low; cross_high; cross_sum)
```

```
FIND_MAXIMUM_SUBARRAY(A)
1.      max_sum = 0;
2.      max_left = -1
3.      max_right = -1

4.      For i = 1 to n
5.              For j = i to n
6.                      sum = 0;
7.                      For k = i to j
8.                              sum += A[k];
9.                              if (sum > max_sum)
10.                                     max_sum = sum;
11.                                     max_left = i
12.                                     max_right = j

13. return (max_left; max_right; max_sum)
```

```
FIND_MAXIMUM_SUBARRAY(A)
1.      max_sum = 0;
2.      max_left = -1
3.      max_right = -1

4.      For i = 1 to n
5.              sum = 0
5.              For j = i to n
8.                      sum += A[j];
9.                      if (sum > max_sum)
10.                             max_sum = sum;
11.                             max_left = i
12.                             max_right = j

13. return (max_left; max_right; max_sum)
```

```
FIND_MAXIMUM_SUBARRAY(A)
1.      max_sum = 0
2.      max_left = -1
3.      max_right = -1

4.      maximum = 0;
5.      maximum_left = -1

6.      For i = 0 to n
7.              if( maximum + A[i] >  0)
8.                      maximum = maximum + A[i]
9.              else
10.                     maximum_left = i+1

11.             if( max_sum     < maximum)
12.                     max_sum = maximum
13.                     max_left = maximum_left
14.                     max_right = i
15.     return (max_left; max_right; max_sum)
```