Group Project 1

```
int max=0;
int sum=0;
int length= sizeof(a)/sizeof(a[0])
maxsubarray(a[1,...,n])
    for(int i=0; i< length      ; i++)
        sum=0;
        for(int j=i; j< length      ; j++)
            sum=sum+a[j];
            if(max<sum)
                max=sum;
    return max;
```

$O(n^2)$

	1	2	3	4	5	6
	[3,	-4,	7,	9,	-2,	3]
①	3 max=3					
②	-1 max=3	② -4 max=16				
③	6 max=6	③ 3 max=16	③ 7 max 16			
④	15 max=15	④ 12 max 16	④ 16 max 16	④ 9 max 17		
⑤	13 max 15	⑤ 10 max=16	⑤ 14 max 16	⑤ 7 max 17	-2	
⑥	16 max 16	⑥ 13 max=16	⑥ 17 max 17	⑥ 10 max 17	1	3

The runtime for this algorithm should be $O(n \log(n))$ it first goes through n times then it decreases more and more.