```
Inputs:
    y: array of {+1, -1}: class of the i-th instance
    Q: Q[i][j] = y[i]*y[j]*K[i][j]; K: kernel matrix
    len: number of instances

//parameters
eps = 1e-3 // stopping tolerance
tau = 1e-12

//main routine
initialize alpha array A to all zero
initialize gradient array G to all -1

while(1)
{
    (i,j) = selectB()
    if (j == -1)
        break

    //working set is (i,j)
    a = Q[i][i]+Q[j][j]-2*y[i]y[j]*Q[i][j]
    if (a <= 0)
        a = tau
    b = -y[i]*G[i]+y[j]*G[j]

    //update alpha
    oldAi = A[i], oldAj = A[j]
    A[i] += y[i]*b/a
    A[j] -= y[j]*b/a

    //project alpha back to the feasible region
    sum = y[i]*oldAi + y[j]*oldAj
    if A[i] > C
        A[i] = C
    if A[i] < 0
        A[i] = 0
    A[j] = y[j]*(sum - y[i]*A[i])

    if A[j] > C
        A[j] = C
    if A[j] < 0
        A[j] = 0
    A[i] = y[i]*(sum - y[j]*A[j])

    //update gradient
    deltaAi = A[i] - oldAi,   deltaAj = A[j] - oldAj
    for t = 1 to len
        G[t] += Q[t][i]*deltaAi + Q[t][j]*deltaAj
}

procedure selectB
    //select i
    i = -1
    G_max = -inf
    G_min = inf
    for t = 1 to len
        if(y[t]==+1 and A[t] < C) or (y[t]==-1 and A[t] >0)
            {
                if(-y[t]*G[t] >= G_max)
                    {
                        i = t
```

```
                        G_max = -y[t]*G[t]
                    }
            }

    //select j
    j = -1
    obj_min = inf
    for t = 1 to len
    {
        if(y[t]==+1 and A[t] >0)or(y[t]==-1 and A[t] < C)
        {
            b = G_max + y[t]*G[t]
            if (-y[t]*G[t] <= G_min)
                G_min = -y[t]*G[t]
            if (b > 0)
            {
                    a = Q[i][i]+Q[t][t]-2*y[i]*y[t]*Q[i][t]
                    if (a <= 0)
                        a = tau
                    if (-(b*b)/a <= obj_min)
                    {
                        j = t
                        obj_min = -(b*b)/a
                    }
            }
        }
    }

    if (G_max-G_min < eps)
        return (-1,-1)

    return (i,j)
end procedure
```