

## Electronic Insert I.1 – DCT-II / DCT-III code generator

```
#####
# DCT-II / DCT-III generator
#
# Based on:
# "A low multiplicative complexity fast recursive DCT-2 algorithm"
# by Maxim Vashkevich and Alexander Petrovsky / arXiv / 20 Jul 2012
#####

import math
import sys
N = 8

#####
# Base transforms / generators
#####

CNTR = 0
def makeTmp():
    global CNTR
    result = "t{:02d}".format(CNTR)
    CNTR = CNTR + 1
    return result

def makeVar(i):
    return "i{:02d}".format(i)

def add(x, y):
    tmp = makeTmp()
    print(tmp + " = " + x + " + " + y + ";")
    return tmp

def sub(x, y):
    tmp = makeTmp()
    print(tmp + " = " + x + " - " + y + ";")
    return tmp

def mul(x, c):
    tmp = makeTmp()
    print(tmp + " = " + x + " * " + c + ";")
    return tmp

# 2.0 * math.cos((a + 0.0) / (b + 0.0) * math.pi)
```

```

def C2(a, b):
    return "c_c2_" + str(a) + "_" + str(b)

# 1.0 / C2(a, b)
def iC2(a, b):
    return "c_ic2_" + str(a) + "_" + str(b)

#####
# Utilities
#####

# Generate identity matrix. Usually this matrix is passed to
# DCT algorithm to generate "basis" vectors of the transform.
def makeVars():
    return [makeVar(i) for i in range(N)]

# Split list of variables info halves.
def split(x):
    m = len(x)
    m2 = m // 2
    return (x[0 : m2], x[m2 : m])

# Make a list of variables in a reverse order.
def reverse(varz):
    m = len(varz)
    result = [0] * m
    for i in range(m):
        result[i] = varz[m - 1 - i]
    return result

# Apply permutation
def permute(x, p):
    return [x[p[i]] for i in range(len(p))]

def transposePermutation(p):
    n = len(p)
    result = [0] * n
    for i in range(n):
        result[p[i]] = i
    return result

# See paper. Split even-odd elements.
def P(n):
    if n == 1:
        return [0]

```

```

n2 = n // 2
return [2 * i for i in range(n2)] + [2 * i + 1 for i in range(n2)]

# See paper. Interleave first and second half.
def Pt(n):
    return transposePermutation(P(n))

#####
# Scheme
#####

def B2(x):
    n = len(x)
    n2 = n // 2
    if n == 1:
        raise "oops"
    (top, bottom) = split(x)
    bottom = reverse(bottom)
    t = [add(top[i], bottom[i]) for i in range(n2)]
    b = [sub(top[i], bottom[i]) for i in range(n2)]
    return t + b

def iB2(x):
    n = len(x)
    n2 = n // 2
    if n == 1:
        raise "oops"
    (top, bottom) = split(x)
    t = [add(top[i], bottom[i]) for i in range(n2)]
    b = [sub(top[i], bottom[i]) for i in range(n2)]
    return t + reverse(b)

def B4(x, rn):
    n = len(x)
    n2 = n // 2
    if n == 1:
        raise "oops"
    (top, bottom) = split(x)
    rbottom = reverse(bottom)
    t = [sub(top[i], rbottom[i]) for i in range(n2)]
    b = [mul(bottom[i], C2(rn, 2 * N)) for i in range(n2)]
    top = [add(t[i], b[i]) for i in range(n2)]
    bottom = [sub(t[i], b[i]) for i in range(n2)]
    return top + bottom

```

```

def iB4(x, rn):
    n = len(x)
    n2 = n // 2
    if n == 1:
        raise "oops"
    (top, bottom) = split(x)
    t = [add(top[i], bottom[i]) for i in range(n2)]
    b = [sub(top[i], bottom[i]) for i in range(n2)]
    bottom = [mul(b[i], iC2(rn, 2 * N)) for i in range(n2)]
    rbottom = reverse(bottom)
    top = [add(t[i], rbottom[i]) for i in range(n2)]
    return top + bottom

```

```

def P4(n):
    if n == 1:
        return [0]
    if n == 2:
        return [0, 1]
    n2 = n // 2
    result = [0] * n
    tc = 0
    bc = 0
    i = 0
    result[i] = tc; tc = tc + 1; i = i + 1
    turn = True
    while i < n - 1:
        if turn:
            result[i] = n2 + bc; bc = bc + 1; i = i + 1
            result[i] = n2 + bc; bc = bc + 1; i = i + 1
        else:
            result[i] = tc; tc = tc + 1; i = i + 1
            result[i] = tc; tc = tc + 1; i = i + 1
        turn = not turn
    result[i] = tc; tc = tc + 1; i = i + 1
    return result

```

```

def iP4(n):
    return transposePermutation(P4(n))

```

```

def d2n(x):
    n = len(x)
    if n == 1:
        return x
    y = B2(x)
    (top, bottom) = split(y)

```

```

    return permute(d2n(top) + d4n(bottom, N // 2), Pt(n))

def id2n(x):
    n = len(x)
    if n == 1:
        return x
    (top, bottom) = split(permute(x, P(n)))
    return iB2(id2n(top) + id4n(bottom, N // 2))

def d4n(x, rn):
    n = len(x)
    if n == 1:
        return x
    y = B4(x, rn)
    (top, bottom) = split(y)
    rn2 = rn // 2
    return permute(d4n(top, rn2) + d4n(bottom, N - rn2), P4(n))

def id4n(x, rn):
    n = len(x)
    if n == 1:
        return x
    (top, bottom) = split(permute(x, iP4(n)))
    rn2 = rn // 2
    y = id4n(top, rn2) + id4n(bottom, N - rn2)
    return iB4(y, rn)

#####
# Main.
#####

def help():
    print("Usage: %s [N [T]]" % sys.argv[0])
    print("  N should be the power of 2, default is 8")
    print("  T is one of {2, 3}, default is 2")
    sys.exit()

def parseInt(s):
    try:
        return int(s)
    except ValueError:
        help()

if __name__ == "__main__":
    if len(sys.argv) < 1 or len(sys.argv) > 3: help()

```













```

2, 199, 1, 222, 93, 94, 1, 232, 2, 65, 74, 139, 201, 48,
2, 254, 169, 127, 52, 243, 251, 249, 102, 86, 202, 153, 65, 65,
146, 69, 8, 238, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128,
128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128,
128, 128, 128, 128, 128, 128, 128, 128

```

**Table M.3 – Protocol Buffer descriptor of top-level structure of losslessly compressed JPEG stream**

```

message Header {
  optional uint64 width = 1;
  optional uint64 height = 2;
  required uint64 version_and_component_count_code = 3;
  optional uint64 subsampling_code = 4;
}

message Jpeg {
  required bytes signature = 1;
  required Header header = 2;
  optional bytes meta_data = 3;
  optional bytes jpeg1_internals = 4;
  optional bytes quant_data = 5;
  optional bytes histogram_data = 6;
  optional bytes dc_data = 7;
  optional bytes ac_data = 8;
  optional bytes original_jpg = 9;
}

```

**Table M.4 – APP0 template**

```

0xE0, 0x00, 0x10, 0x4A, 0x46, 0x49, 0x46, 0x00, 0x01, 0x01, 0x00, 0x00,
0x01, 0x00, 0x01, 0x00, 0x00

```

**Table M.6 – common ICC profile template**

```

0xE2, 0x0C, 0x58, 0x49, 0x43, 0x43, 0x5F, 0x50, 0x52, 0x4F, 0x46, 0x49,
0x4C, 0x45, 0x00, 0x01, 0x01, 0x00, 0x00, 0x00, 0x0C, 0x48, 0x4C, 0x69, 0x6E,
0x6F, 0x02, 0x10, 0x00, 0x00, 0x6D, 0x6E, 0x74, 0x72, 0x52, 0x47, 0x42,
0x20, 0x58, 0x59, 0x5A, 0x20, 0x07, 0xCE, 0x00, 0x02, 0x00, 0x09, 0x00,
0x06, 0x00, 0x31, 0x00, 0x00, 0x61, 0x63, 0x73, 0x70, 0x4D, 0x53, 0x46,
0x54, 0x00, 0x00, 0x00, 0x00, 0x49, 0x45, 0x43, 0x20, 0x73, 0x52, 0x47,
0x42, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0x00, 0x00, 0xF6, 0xD6, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0xD3,

```

0x2D, 0x48, 0x50, 0x20, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x11, 0x63, 0x70, 0x72, 0x74, 0x00, 0x00, 0x01,  
0x50, 0x00, 0x00, 0x00, 0x33, 0x64, 0x65, 0x73, 0x63, 0x00, 0x00, 0x01,  
0x84, 0x00, 0x00, 0x00, 0x6C, 0x77, 0x74, 0x70, 0x74, 0x00, 0x00, 0x01,  
0xF0, 0x00, 0x00, 0x00, 0x14, 0x62, 0x6B, 0x70, 0x74, 0x00, 0x00, 0x02,  
0x04, 0x00, 0x00, 0x00, 0x14, 0x72, 0x58, 0x59, 0x5A, 0x00, 0x00, 0x02,  
0x18, 0x00, 0x00, 0x00, 0x14, 0x67, 0x58, 0x59, 0x5A, 0x00, 0x00, 0x02,  
0x2C, 0x00, 0x00, 0x00, 0x14, 0x62, 0x58, 0x59, 0x5A, 0x00, 0x00, 0x02,  
0x40, 0x00, 0x00, 0x00, 0x14, 0x64, 0x6D, 0x6E, 0x64, 0x00, 0x00, 0x02,  
0x54, 0x00, 0x00, 0x00, 0x70, 0x64, 0x6D, 0x64, 0x64, 0x00, 0x00, 0x02,  
0xC4, 0x00, 0x00, 0x00, 0x88, 0x76, 0x75, 0x65, 0x64, 0x00, 0x00, 0x03,  
0x4C, 0x00, 0x00, 0x00, 0x86, 0x76, 0x69, 0x65, 0x77, 0x00, 0x00, 0x03,  
0xD4, 0x00, 0x00, 0x00, 0x24, 0x6C, 0x75, 0x6D, 0x69, 0x00, 0x00, 0x03,  
0xF8, 0x00, 0x00, 0x00, 0x14, 0x6D, 0x65, 0x61, 0x73, 0x00, 0x00, 0x04,  
0x0C, 0x00, 0x00, 0x00, 0x24, 0x74, 0x65, 0x63, 0x68, 0x00, 0x00, 0x04,  
0x30, 0x00, 0x00, 0x00, 0x0C, 0x72, 0x54, 0x52, 0x43, 0x00, 0x00, 0x04,  
0x3C, 0x00, 0x00, 0x08, 0x0C, 0x67, 0x54, 0x52, 0x43, 0x00, 0x00, 0x04,  
0x3C, 0x00, 0x00, 0x08, 0x0C, 0x62, 0x54, 0x52, 0x43, 0x00, 0x00, 0x04,  
0x3C, 0x00, 0x00, 0x08, 0x0C, 0x74, 0x65, 0x78, 0x74, 0x00, 0x00, 0x00,  
0x00, 0x43, 0x6F, 0x70, 0x79, 0x72, 0x69, 0x67, 0x68, 0x74, 0x20, 0x28,  
0x63, 0x29, 0x20, 0x31, 0x39, 0x39, 0x38, 0x20, 0x48, 0x65, 0x77, 0x6C,  
0x65, 0x74, 0x74, 0x2D, 0x50, 0x61, 0x63, 0x6B, 0x61, 0x72, 0x64, 0x20,  
0x43, 0x6F, 0x6D, 0x70, 0x61, 0x6E, 0x79, 0x00, 0x00, 0x64, 0x65, 0x73,  
0x63, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x12, 0x73, 0x52, 0x47,  
0x42, 0x20, 0x49, 0x45, 0x43, 0x36, 0x31, 0x39, 0x36, 0x36, 0x2D, 0x32,  
0x2E, 0x31, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x12, 0x73, 0x52, 0x47, 0x42, 0x20, 0x49, 0x45, 0x43, 0x36, 0x31,  
0x39, 0x36, 0x36, 0x2D, 0x32, 0x2E, 0x31, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x58, 0x59, 0x5A,  
0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF3, 0x51, 0x00, 0x01, 0x00,  
0x00, 0x00, 0x01, 0x16, 0xCC, 0x58, 0x59, 0x5A, 0x20, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x58, 0x59, 0x5A, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x6F,  
0xA2, 0x00, 0x00, 0x38, 0xF5, 0x00, 0x00, 0x03, 0x90, 0x58, 0x59, 0x5A,  
0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x62, 0x99, 0x00, 0x00, 0xB7,  
0x85, 0x00, 0x00, 0x18, 0xDA, 0x58, 0x59, 0x5A, 0x20, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x24, 0xA0, 0x00, 0x00, 0x0F, 0x84, 0x00, 0x00, 0xB6,  
0xCF, 0x64, 0x65, 0x73, 0x63, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x16, 0x49, 0x45, 0x43, 0x20, 0x68, 0x74, 0x74, 0x70, 0x3A, 0x2F, 0x2F,

0x77, 0x77, 0x77, 0x2E, 0x69, 0x65, 0x63, 0x2E, 0x63, 0x68, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x16, 0x49, 0x45,  
0x43, 0x20, 0x68, 0x74, 0x74, 0x70, 0x3A, 0x2F, 0x2F, 0x77, 0x77, 0x77,  
0x2E, 0x69, 0x65, 0x63, 0x2E, 0x63, 0x68, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x64, 0x65, 0x73, 0x63, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x2E, 0x49, 0x45, 0x43, 0x20, 0x36, 0x31, 0x39,  
0x36, 0x36, 0x2D, 0x32, 0x2E, 0x31, 0x20, 0x44, 0x65, 0x66, 0x61, 0x75,  
0x6C, 0x74, 0x20, 0x52, 0x47, 0x42, 0x20, 0x63, 0x6F, 0x6C, 0x6F, 0x75,  
0x72, 0x20, 0x73, 0x70, 0x61, 0x63, 0x65, 0x20, 0x2D, 0x20, 0x73, 0x52,  
0x47, 0x42, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x2E, 0x49, 0x45, 0x43, 0x20, 0x36, 0x31, 0x39, 0x36, 0x36, 0x2D,  
0x32, 0x2E, 0x31, 0x20, 0x44, 0x65, 0x66, 0x61, 0x75, 0x6C, 0x74, 0x20,  
0x52, 0x47, 0x42, 0x20, 0x63, 0x6F, 0x6C, 0x6F, 0x75, 0x72, 0x20, 0x73,  
0x70, 0x61, 0x63, 0x65, 0x20, 0x2D, 0x20, 0x73, 0x52, 0x47, 0x42, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x64, 0x65, 0x73,  
0x63, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x2C, 0x52, 0x65, 0x66,  
0x65, 0x72, 0x65, 0x6E, 0x63, 0x65, 0x20, 0x56, 0x69, 0x65, 0x77, 0x69,  
0x6E, 0x67, 0x20, 0x43, 0x6F, 0x6E, 0x64, 0x69, 0x74, 0x69, 0x6F, 0x6E,  
0x20, 0x69, 0x6E, 0x20, 0x49, 0x45, 0x43, 0x36, 0x31, 0x39, 0x36, 0x36,  
0x2D, 0x32, 0x2E, 0x31, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x2C, 0x52, 0x65, 0x66, 0x65, 0x72, 0x65, 0x6E, 0x63,  
0x65, 0x20, 0x56, 0x69, 0x65, 0x77, 0x69, 0x6E, 0x67, 0x20, 0x43, 0x6F,  
0x6E, 0x64, 0x69, 0x74, 0x69, 0x6F, 0x6E, 0x20, 0x69, 0x6E, 0x20, 0x49,  
0x45, 0x43, 0x36, 0x31, 0x39, 0x36, 0x36, 0x2D, 0x32, 0x2E, 0x31, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x76, 0x69, 0x65, 0x77, 0x00, 0x00, 0x00, 0x00, 0x00, 0x13, 0xA4,  
0xFE, 0x00, 0x14, 0x5F, 0x2E, 0x00, 0x10, 0xCF, 0x14, 0x00, 0x03, 0xED,  
0xCC, 0x00, 0x04, 0x13, 0x0B, 0x00, 0x03, 0x5C, 0x9E, 0x00, 0x00, 0x00,  
0x01, 0x58, 0x59, 0x5A, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x4C, 0x09,  
0x56, 0x00, 0x50, 0x00, 0x00, 0x00, 0x57, 0x1F, 0xE7, 0x6D, 0x65, 0x61,  
0x73, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x02, 0x8F, 0x00, 0x00, 0x00, 0x02, 0x73, 0x69, 0x67,  
0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x43, 0x52, 0x54, 0x20, 0x63, 0x75, 0x72,  
0x76, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x00,  
0x05, 0x00, 0x0A, 0x00, 0x0F, 0x00, 0x14, 0x00, 0x19, 0x00, 0x1E, 0x00,  
0x23, 0x00, 0x28, 0x00, 0x2D, 0x00, 0x32, 0x00, 0x37, 0x00, 0x3B, 0x00,  
0x40, 0x00, 0x45, 0x00, 0x4A, 0x00, 0x4F, 0x00, 0x54, 0x00, 0x59, 0x00,  
0x5E, 0x00, 0x63, 0x00, 0x68, 0x00, 0x6D, 0x00, 0x72, 0x00, 0x77, 0x00,  
0x7C, 0x00, 0x81, 0x00, 0x86, 0x00, 0x8B, 0x00, 0x90, 0x00, 0x95, 0x00,

0x9A, 0x00, 0x9F, 0x00, 0xA4, 0x00, 0xA9, 0x00, 0xAE, 0x00, 0xB2, 0x00,  
0xB7, 0x00, 0xBC, 0x00, 0xC1, 0x00, 0xC6, 0x00, 0xCB, 0x00, 0xD0, 0x00,  
0xD5, 0x00, 0xDB, 0x00, 0xE0, 0x00, 0xE5, 0x00, 0xEB, 0x00, 0xF0, 0x00,  
0xF6, 0x00, 0xFB, 0x01, 0x01, 0x01, 0x07, 0x01, 0x0D, 0x01, 0x13, 0x01,  
0x19, 0x01, 0x1F, 0x01, 0x25, 0x01, 0x2B, 0x01, 0x32, 0x01, 0x38, 0x01,  
0x3E, 0x01, 0x45, 0x01, 0x4C, 0x01, 0x52, 0x01, 0x59, 0x01, 0x60, 0x01,  
0x67, 0x01, 0x6E, 0x01, 0x75, 0x01, 0x7C, 0x01, 0x83, 0x01, 0x8B, 0x01,  
0x92, 0x01, 0x9A, 0x01, 0xA1, 0x01, 0xA9, 0x01, 0xB1, 0x01, 0xB9, 0x01,  
0xC1, 0x01, 0xC9, 0x01, 0xD1, 0x01, 0xD9, 0x01, 0xE1, 0x01, 0xE9, 0x01,  
0xF2, 0x01, 0xFA, 0x02, 0x03, 0x02, 0x0C, 0x02, 0x14, 0x02, 0x1D, 0x02,  
0x26, 0x02, 0x2F, 0x02, 0x38, 0x02, 0x41, 0x02, 0x4B, 0x02, 0x54, 0x02,  
0x5D, 0x02, 0x67, 0x02, 0x71, 0x02, 0x7A, 0x02, 0x84, 0x02, 0x8E, 0x02,  
0x98, 0x02, 0xA2, 0x02, 0xAC, 0x02, 0xB6, 0x02, 0xC1, 0x02, 0xCB, 0x02,  
0xD5, 0x02, 0xE0, 0x02, 0xEB, 0x02, 0xF5, 0x03, 0x00, 0x03, 0x0B, 0x03,  
0x16, 0x03, 0x21, 0x03, 0x2D, 0x03, 0x38, 0x03, 0x43, 0x03, 0x4F, 0x03,  
0x5A, 0x03, 0x66, 0x03, 0x72, 0x03, 0x7E, 0x03, 0x8A, 0x03, 0x96, 0x03,  
0xA2, 0x03, 0xAE, 0x03, 0xBA, 0x03, 0xC7, 0x03, 0xD3, 0x03, 0xE0, 0x03,  
0xEC, 0x03, 0xF9, 0x04, 0x06, 0x04, 0x13, 0x04, 0x20, 0x04, 0x2D, 0x04,  
0x3B, 0x04, 0x48, 0x04, 0x55, 0x04, 0x63, 0x04, 0x71, 0x04, 0x7E, 0x04,  
0x8C, 0x04, 0x9A, 0x04, 0xA8, 0x04, 0xB6, 0x04, 0xC4, 0x04, 0xD3, 0x04,  
0xE1, 0x04, 0xF0, 0x04, 0xFE, 0x05, 0x0D, 0x05, 0x1C, 0x05, 0x2B, 0x05,  
0x3A, 0x05, 0x49, 0x05, 0x58, 0x05, 0x67, 0x05, 0x77, 0x05, 0x86, 0x05,  
0x96, 0x05, 0xA6, 0x05, 0xB5, 0x05, 0xC5, 0x05, 0xD5, 0x05, 0xE5, 0x05,  
0xF6, 0x06, 0x06, 0x06, 0x16, 0x06, 0x27, 0x06, 0x37, 0x06, 0x48, 0x06,  
0x59, 0x06, 0x6A, 0x06, 0x7B, 0x06, 0x8C, 0x06, 0x9D, 0x06, 0xAF, 0x06,  
0xC0, 0x06, 0xD1, 0x06, 0xE3, 0x06, 0xF5, 0x07, 0x07, 0x07, 0x19, 0x07,  
0x2B, 0x07, 0x3D, 0x07, 0x4F, 0x07, 0x61, 0x07, 0x74, 0x07, 0x86, 0x07,  
0x99, 0x07, 0xAC, 0x07, 0xBF, 0x07, 0xD2, 0x07, 0xE5, 0x07, 0xF8, 0x08,  
0x0B, 0x08, 0x1F, 0x08, 0x32, 0x08, 0x46, 0x08, 0x5A, 0x08, 0x6E, 0x08,  
0x82, 0x08, 0x96, 0x08, 0xAA, 0x08, 0xBE, 0x08, 0xD2, 0x08, 0xE7, 0x08,  
0xFB, 0x09, 0x10, 0x09, 0x25, 0x09, 0x3A, 0x09, 0x4F, 0x09, 0x64, 0x09,  
0x79, 0x09, 0x8F, 0x09, 0xA4, 0x09, 0xBA, 0x09, 0xCF, 0x09, 0xE5, 0x09,  
0xFB, 0x0A, 0x11, 0x0A, 0x27, 0x0A, 0x3D, 0x0A, 0x54, 0x0A, 0x6A, 0x0A,  
0x81, 0x0A, 0x98, 0x0A, 0xAE, 0x0A, 0xC5, 0x0A, 0xDC, 0x0A, 0xF3, 0x0B,  
0x0B, 0x0B, 0x22, 0x0B, 0x39, 0x0B, 0x51, 0x0B, 0x69, 0x0B, 0x80, 0x0B,  
0x98, 0x0B, 0xB0, 0x0B, 0xC8, 0x0B, 0xE1, 0x0B, 0xF9, 0x0C, 0x12, 0x0C,  
0x2A, 0x0C, 0x43, 0x0C, 0x5C, 0x0C, 0x75, 0x0C, 0x8E, 0x0C, 0xA7, 0x0C,  
0xC0, 0x0C, 0xD9, 0x0C, 0xF3, 0x0D, 0x0D, 0x0D, 0x26, 0x0D, 0x40, 0x0D,  
0x5A, 0x0D, 0x74, 0x0D, 0x8E, 0x0D, 0xA9, 0x0D, 0xC3, 0x0D, 0xDE, 0x0D,  
0xF8, 0x0E, 0x13, 0x0E, 0x2E, 0x0E, 0x49, 0x0E, 0x64, 0x0E, 0x7F, 0x0E,  
0x9B, 0x0E, 0xB6, 0x0E, 0xD2, 0x0E, 0xEE, 0x0F, 0x09, 0x0F, 0x25, 0x0F,  
0x41, 0x0F, 0x5E, 0x0F, 0x7A, 0x0F, 0x96, 0x0F, 0xB3, 0x0F, 0xCF, 0x0F,  
0xEC, 0x10, 0x09, 0x10, 0x26, 0x10, 0x43, 0x10, 0x61, 0x10, 0x7E, 0x10,  
0x9B, 0x10, 0xB9, 0x10, 0xD7, 0x10, 0xF5, 0x11, 0x13, 0x11, 0x31, 0x11,  
0x4F, 0x11, 0x6D, 0x11, 0x8C, 0x11, 0xAA, 0x11, 0xC9, 0x11, 0xE8, 0x12,

0x07, 0x12, 0x26, 0x12, 0x45, 0x12, 0x64, 0x12, 0x84, 0x12, 0xA3, 0x12,  
0xC3, 0x12, 0xE3, 0x13, 0x03, 0x13, 0x23, 0x13, 0x43, 0x13, 0x63, 0x13,  
0x83, 0x13, 0xA4, 0x13, 0xC5, 0x13, 0xE5, 0x14, 0x06, 0x14, 0x27, 0x14,  
0x49, 0x14, 0x6A, 0x14, 0x8B, 0x14, 0xAD, 0x14, 0xCE, 0x14, 0xF0, 0x15,  
0x12, 0x15, 0x34, 0x15, 0x56, 0x15, 0x78, 0x15, 0x9B, 0x15, 0xBD, 0x15,  
0xE0, 0x16, 0x03, 0x16, 0x26, 0x16, 0x49, 0x16, 0x6C, 0x16, 0x8F, 0x16,  
0xB2, 0x16, 0xD6, 0x16, 0xFA, 0x17, 0x1D, 0x17, 0x41, 0x17, 0x65, 0x17,  
0x89, 0x17, 0xAE, 0x17, 0xD2, 0x17, 0xF7, 0x18, 0x1B, 0x18, 0x40, 0x18,  
0x65, 0x18, 0x8A, 0x18, 0xAF, 0x18, 0xD5, 0x18, 0xFA, 0x19, 0x20, 0x19,  
0x45, 0x19, 0x6B, 0x19, 0x91, 0x19, 0xB7, 0x19, 0xDD, 0x1A, 0x04, 0x1A,  
0x2A, 0x1A, 0x51, 0x1A, 0x77, 0x1A, 0x9E, 0x1A, 0xC5, 0x1A, 0xEC, 0x1B,  
0x14, 0x1B, 0x3B, 0x1B, 0x63, 0x1B, 0x8A, 0x1B, 0xB2, 0x1B, 0xDA, 0x1C,  
0x02, 0x1C, 0x2A, 0x1C, 0x52, 0x1C, 0x7B, 0x1C, 0xA3, 0x1C, 0xCC, 0x1C,  
0xF5, 0x1D, 0x1E, 0x1D, 0x47, 0x1D, 0x70, 0x1D, 0x99, 0x1D, 0xC3, 0x1D,  
0xEC, 0x1E, 0x16, 0x1E, 0x40, 0x1E, 0x6A, 0x1E, 0x94, 0x1E, 0xBE, 0x1E,  
0xE9, 0x1F, 0x13, 0x1F, 0x3E, 0x1F, 0x69, 0x1F, 0x94, 0x1F, 0xBF, 0x1F,  
0xEA, 0x20, 0x15, 0x20, 0x41, 0x20, 0x6C, 0x20, 0x98, 0x20, 0xC4, 0x20,  
0xF0, 0x21, 0x1C, 0x21, 0x48, 0x21, 0x75, 0x21, 0xA1, 0x21, 0xCE, 0x21,  
0xFB, 0x22, 0x27, 0x22, 0x55, 0x22, 0x82, 0x22, 0xAF, 0x22, 0xDD, 0x23,  
0x0A, 0x23, 0x38, 0x23, 0x66, 0x23, 0x94, 0x23, 0xC2, 0x23, 0xF0, 0x24,  
0x1F, 0x24, 0x4D, 0x24, 0x7C, 0x24, 0xAB, 0x24, 0xDA, 0x25, 0x09, 0x25,  
0x38, 0x25, 0x68, 0x25, 0x97, 0x25, 0xC7, 0x25, 0xF7, 0x26, 0x27, 0x26,  
0x57, 0x26, 0x87, 0x26, 0xB7, 0x26, 0xE8, 0x27, 0x18, 0x27, 0x49, 0x27,  
0x7A, 0x27, 0xAB, 0x27, 0xDC, 0x28, 0x0D, 0x28, 0x3F, 0x28, 0x71, 0x28,  
0xA2, 0x28, 0xD4, 0x29, 0x06, 0x29, 0x38, 0x29, 0x6B, 0x29, 0x9D, 0x29,  
0xD0, 0x2A, 0x02, 0x2A, 0x35, 0x2A, 0x68, 0x2A, 0x9B, 0x2A, 0xCF, 0x2B,  
0x02, 0x2B, 0x36, 0x2B, 0x69, 0x2B, 0x9D, 0x2B, 0xD1, 0x2C, 0x05, 0x2C,  
0x39, 0x2C, 0x6E, 0x2C, 0xA2, 0x2C, 0xD7, 0x2D, 0x0C, 0x2D, 0x41, 0x2D,  
0x76, 0x2D, 0xAB, 0x2D, 0xE1, 0x2E, 0x16, 0x2E, 0x4C, 0x2E, 0x82, 0x2E,  
0xB7, 0x2E, 0xEE, 0x2F, 0x24, 0x2F, 0x5A, 0x2F, 0x91, 0x2F, 0xC7, 0x2F,  
0xFE, 0x30, 0x35, 0x30, 0x6C, 0x30, 0xA4, 0x30, 0xDB, 0x31, 0x12, 0x31,  
0x4A, 0x31, 0x82, 0x31, 0xBA, 0x31, 0xF2, 0x32, 0x2A, 0x32, 0x63, 0x32,  
0x9B, 0x32, 0xD4, 0x33, 0x0D, 0x33, 0x46, 0x33, 0x7F, 0x33, 0xB8, 0x33,  
0xF1, 0x34, 0x2B, 0x34, 0x65, 0x34, 0x9E, 0x34, 0xD8, 0x35, 0x13, 0x35,  
0x4D, 0x35, 0x87, 0x35, 0xC2, 0x35, 0xFD, 0x36, 0x37, 0x36, 0x72, 0x36,  
0xAE, 0x36, 0xE9, 0x37, 0x24, 0x37, 0x60, 0x37, 0x9C, 0x37, 0xD7, 0x38,  
0x14, 0x38, 0x50, 0x38, 0x8C, 0x38, 0xC8, 0x39, 0x05, 0x39, 0x42, 0x39,  
0x7F, 0x39, 0xBC, 0x39, 0xF9, 0x3A, 0x36, 0x3A, 0x74, 0x3A, 0xB2, 0x3A,  
0xEF, 0x3B, 0x2D, 0x3B, 0x6B, 0x3B, 0xAA, 0x3B, 0xE8, 0x3C, 0x27, 0x3C,  
0x65, 0x3C, 0xA4, 0x3C, 0xE3, 0x3D, 0x22, 0x3D, 0x61, 0x3D, 0xA1, 0x3D,  
0xE0, 0x3E, 0x20, 0x3E, 0x60, 0x3E, 0xA0, 0x3E, 0xE0, 0x3F, 0x21, 0x3F,  
0x61, 0x3F, 0xA2, 0x3F, 0xE2, 0x40, 0x23, 0x40, 0x64, 0x40, 0xA6, 0x40,  
0xE7, 0x41, 0x29, 0x41, 0x6A, 0x41, 0xAC, 0x41, 0xEE, 0x42, 0x30, 0x42,  
0x72, 0x42, 0xB5, 0x42, 0xF7, 0x43, 0x3A, 0x43, 0x7D, 0x43, 0xC0, 0x44,  
0x03, 0x44, 0x47, 0x44, 0x8A, 0x44, 0xCE, 0x45, 0x12, 0x45, 0x55, 0x45,

0x9A, 0x45, 0xDE, 0x46, 0x22, 0x46, 0x67, 0x46, 0xAB, 0x46, 0xF0, 0x47,  
0x35, 0x47, 0x7B, 0x47, 0xC0, 0x48, 0x05, 0x48, 0x4B, 0x48, 0x91, 0x48,  
0xD7, 0x49, 0x1D, 0x49, 0x63, 0x49, 0xA9, 0x49, 0xF0, 0x4A, 0x37, 0x4A,  
0x7D, 0x4A, 0xC4, 0x4B, 0x0C, 0x4B, 0x53, 0x4B, 0x9A, 0x4B, 0xE2, 0x4C,  
0x2A, 0x4C, 0x72, 0x4C, 0xBA, 0x4D, 0x02, 0x4D, 0x4A, 0x4D, 0x93, 0x4D,  
0xDC, 0x4E, 0x25, 0x4E, 0x6E, 0x4E, 0xB7, 0x4F, 0x00, 0x4F, 0x49, 0x4F,  
0x93, 0x4F, 0xDD, 0x50, 0x27, 0x50, 0x71, 0x50, 0xBB, 0x51, 0x06, 0x51,  
0x50, 0x51, 0x9B, 0x51, 0xE6, 0x52, 0x31, 0x52, 0x7C, 0x52, 0xC7, 0x53,  
0x13, 0x53, 0x5F, 0x53, 0xAA, 0x53, 0xF6, 0x54, 0x42, 0x54, 0x8F, 0x54,  
0xDB, 0x55, 0x28, 0x55, 0x75, 0x55, 0xC2, 0x56, 0x0F, 0x56, 0x5C, 0x56,  
0xA9, 0x56, 0xF7, 0x57, 0x44, 0x57, 0x92, 0x57, 0xE0, 0x58, 0x2F, 0x58,  
0x7D, 0x58, 0xCB, 0x59, 0x1A, 0x59, 0x69, 0x59, 0xB8, 0x5A, 0x07, 0x5A,  
0x56, 0x5A, 0xA6, 0x5A, 0xF5, 0x5B, 0x45, 0x5B, 0x95, 0x5B, 0xE5, 0x5C,  
0x35, 0x5C, 0x86, 0x5C, 0xD6, 0x5D, 0x27, 0x5D, 0x78, 0x5D, 0xC9, 0x5E,  
0x1A, 0x5E, 0x6C, 0x5E, 0xBD, 0x5F, 0x0F, 0x5F, 0x61, 0x5F, 0xB3, 0x60,  
0x05, 0x60, 0x57, 0x60, 0xAA, 0x60, 0xFC, 0x61, 0x4F, 0x61, 0xA2, 0x61,  
0xF5, 0x62, 0x49, 0x62, 0x9C, 0x62, 0xF0, 0x63, 0x43, 0x63, 0x97, 0x63,  
0xEB, 0x64, 0x40, 0x64, 0x94, 0x64, 0xE9, 0x65, 0x3D, 0x65, 0x92, 0x65,  
0xE7, 0x66, 0x3D, 0x66, 0x92, 0x66, 0xE8, 0x67, 0x3D, 0x67, 0x93, 0x67,  
0xE9, 0x68, 0x3F, 0x68, 0x96, 0x68, 0xEC, 0x69, 0x43, 0x69, 0x9A, 0x69,  
0xF1, 0x6A, 0x48, 0x6A, 0x9F, 0x6A, 0xF7, 0x6B, 0x4F, 0x6B, 0xA7, 0x6B,  
0xFF, 0x6C, 0x57, 0x6C, 0xAF, 0x6D, 0x08, 0x6D, 0x60, 0x6D, 0xB9, 0x6E,  
0x12, 0x6E, 0x6B, 0x6E, 0xC4, 0x6F, 0x1E, 0x6F, 0x78, 0x6F, 0xD1, 0x70,  
0x2B, 0x70, 0x86, 0x70, 0xE0, 0x71, 0x3A, 0x71, 0x95, 0x71, 0xF0, 0x72,  
0x4B, 0x72, 0xA6, 0x73, 0x01, 0x73, 0x5D, 0x73, 0xB8, 0x74, 0x14, 0x74,  
0x70, 0x74, 0xCC, 0x75, 0x28, 0x75, 0x85, 0x75, 0xE1, 0x76, 0x3E, 0x76,  
0x9B, 0x76, 0xF8, 0x77, 0x56, 0x77, 0xB3, 0x78, 0x11, 0x78, 0x6E, 0x78,  
0xCC, 0x79, 0x2A, 0x79, 0x89, 0x79, 0xE7, 0x7A, 0x46, 0x7A, 0xA5, 0x7B,  
0x04, 0x7B, 0x63, 0x7B, 0xC2, 0x7C, 0x21, 0x7C, 0x81, 0x7C, 0xE1, 0x7D,  
0x41, 0x7D, 0xA1, 0x7E, 0x01, 0x7E, 0x62, 0x7E, 0xC2, 0x7F, 0x23, 0x7F,  
0x84, 0x7F, 0xE5, 0x80, 0x47, 0x80, 0xA8, 0x81, 0x0A, 0x81, 0x6B, 0x81,  
0xCD, 0x82, 0x30, 0x82, 0x92, 0x82, 0xF4, 0x83, 0x57, 0x83, 0xBA, 0x84,  
0x1D, 0x84, 0x80, 0x84, 0xE3, 0x85, 0x47, 0x85, 0xAB, 0x86, 0x0E, 0x86,  
0x72, 0x86, 0xD7, 0x87, 0x3B, 0x87, 0x9F, 0x88, 0x04, 0x88, 0x69, 0x88,  
0xCE, 0x89, 0x33, 0x89, 0x99, 0x89, 0xFE, 0x8A, 0x64, 0x8A, 0xCA, 0x8B,  
0x30, 0x8B, 0x96, 0x8B, 0xFC, 0x8C, 0x63, 0x8C, 0xCA, 0x8D, 0x31, 0x8D,  
0x98, 0x8D, 0xFF, 0x8E, 0x66, 0x8E, 0xCE, 0x8F, 0x36, 0x8F, 0x9E, 0x90,  
0x06, 0x90, 0x6E, 0x90, 0xD6, 0x91, 0x3F, 0x91, 0xA8, 0x92, 0x11, 0x92,  
0x7A, 0x92, 0xE3, 0x93, 0x4D, 0x93, 0xB6, 0x94, 0x20, 0x94, 0x8A, 0x94,  
0xF4, 0x95, 0x5F, 0x95, 0xC9, 0x96, 0x34, 0x96, 0x9F, 0x97, 0x0A, 0x97,  
0x75, 0x97, 0xE0, 0x98, 0x4C, 0x98, 0xB8, 0x99, 0x24, 0x99, 0x90, 0x99,  
0xFC, 0x9A, 0x68, 0x9A, 0xD5, 0x9B, 0x42, 0x9B, 0xAF, 0x9C, 0x1C, 0x9C,  
0x89, 0x9C, 0xF7, 0x9D, 0x64, 0x9D, 0xD2, 0x9E, 0x40, 0x9E, 0xAE, 0x9F,  
0x1D, 0x9F, 0x8B, 0x9F, 0xFA, 0xA0, 0x69, 0xA0, 0xD8, 0xA1, 0x47, 0xA1,  
0xB6, 0xA2, 0x26, 0xA2, 0x96, 0xA3, 0x06, 0xA3, 0x76, 0xA3, 0xE6, 0xA4,



```

0x56, 0xA4, 0xC7, 0xA5, 0x38, 0xA5, 0xA9, 0xA6, 0x1A, 0xA6, 0x8B, 0xA6,
0xFD, 0xA7, 0x6E, 0xA7, 0xE0, 0xA8, 0x52, 0xA8, 0xC4, 0xA9, 0x37, 0xA9,
0xA9, 0xAA, 0x1C, 0xAA, 0x8F, 0xAB, 0x02, 0xAB, 0x75, 0xAB, 0xE9, 0xAC,
0x5C, 0xAC, 0xD0, 0xAD, 0x44, 0xAD, 0xB8, 0xAE, 0x2D, 0xAE, 0xA1, 0xAF,
0x16, 0xAF, 0x8B, 0xB0, 0x00, 0xB0, 0x75, 0xB0, 0xEA, 0xB1, 0x60, 0xB1,
0xD6, 0xB2, 0x4B, 0xB2, 0xC2, 0xB3, 0x38, 0xB3, 0xAE, 0xB4, 0x25, 0xB4,
0x9C, 0xB5, 0x13, 0xB5, 0x8A, 0xB6, 0x01, 0xB6, 0x79, 0xB6, 0xF0, 0xB7,
0x68, 0xB7, 0xE0, 0xB8, 0x59, 0xB8, 0xD1, 0xB9, 0x4A, 0xB9, 0xC2, 0xBA,
0x3B, 0xBA, 0xB5, 0xBB, 0x2E, 0xBB, 0xA7, 0xBC, 0x21, 0xBC, 0x9B, 0xBD,
0x15, 0xBD, 0x8F, 0xBE, 0x0A, 0xBE, 0x84, 0xBE, 0xFF, 0xBF, 0x7A, 0xBF,
0xF5, 0xC0, 0x70, 0xC0, 0xEC, 0xC1, 0x67, 0xC1, 0xE3, 0xC2, 0x5F, 0xC2,
0xDB, 0xC3, 0x58, 0xC3, 0xD4, 0xC4, 0x51, 0xC4, 0xCE, 0xC5, 0x4B, 0xC5,
0xC8, 0xC6, 0x46, 0xC6, 0xC3, 0xC7, 0x41, 0xC7, 0xBF, 0xC8, 0x3D, 0xC8,
0xBC, 0xC9, 0x3A, 0xC9, 0xB9, 0xCA, 0x38, 0xCA, 0xB7, 0xCB, 0x36, 0xCB,
0xB6, 0xCC, 0x35, 0xCC, 0xB5, 0xCD, 0x35, 0xCD, 0xB5, 0xCE, 0x36, 0xCE,
0xB6, 0xCF, 0x37, 0xCF, 0xB8, 0xD0, 0x39, 0xD0, 0xBA, 0xD1, 0x3C, 0xD1,
0xBE, 0xD2, 0x3F, 0xD2, 0xC1, 0xD3, 0x44, 0xD3, 0xC6, 0xD4, 0x49, 0xD4,
0xCB, 0xD5, 0x4E, 0xD5, 0xD1, 0xD6, 0x55, 0xD6, 0xD8, 0xD7, 0x5C, 0xD7,
0xE0, 0xD8, 0x64, 0xD8, 0xE8, 0xD9, 0x6C, 0xD9, 0xF1, 0xDA, 0x76, 0xDA,
0xFB, 0xDB, 0x80, 0xDC, 0x05, 0xDC, 0x8A, 0xDD, 0x10, 0xDD, 0x96, 0xDE,
0x1C, 0xDE, 0xA2, 0xDF, 0x29, 0xDF, 0xAF, 0xE0, 0x36, 0xE0, 0xBD, 0xE1,
0x44, 0xE1, 0xCC, 0xE2, 0x53, 0xE2, 0xDB, 0xE3, 0x63, 0xE3, 0xEB, 0xE4,
0x73, 0xE4, 0xFC, 0xE5, 0x84, 0xE6, 0x0D, 0xE6, 0x96, 0xE7, 0x1F, 0xE7,
0xA9, 0xE8, 0x32, 0xE8, 0xBC, 0xE9, 0x46, 0xE9, 0xD0, 0xEA, 0x5B, 0xEA,
0xE5, 0xEB, 0x70, 0xEB, 0xFB, 0xEC, 0x86, 0xED, 0x11, 0xED, 0x9C, 0xEE,
0x28, 0xEE, 0xB4, 0xEF, 0x40, 0xEF, 0xCC, 0xF0, 0x58, 0xF0, 0xE5, 0xF1,
0x72, 0xF1, 0xFF, 0xF2, 0x8C, 0xF3, 0x19, 0xF3, 0xA7, 0xF4, 0x34, 0xF4,
0xC2, 0xF5, 0x50, 0xF5, 0xDE, 0xF6, 0x6D, 0xF6, 0xFB, 0xF7, 0x8A, 0xF8,
0x19, 0xF8, 0xA8, 0xF9, 0x38, 0xF9, 0xC7, 0xFA, 0x57, 0xFA, 0xE7, 0xFB,
0x77, 0xFC, 0x07, 0xFC, 0x98, 0xFD, 0x29, 0xFD, 0xBA, 0xFE, 0x4B, 0xFE,
0xDC, 0xFF, 0x6D, 0xFF, 0xFF

```

**Table M.7 – "Ducky" marker template**

```

0xEC, 0x00, 0x11, 0x44, 0x75, 0x63, 0x6B, 0x79, 0x00, 0x01, 0x00, 0x04,
0x00, 0x00, 0x00, 0x64, 0x00, 0x00

```

**Table M.8 – "Adobe" marker template**

```

0xEE, 0x00, 0x0E, 0x41, 0x64, 0x6F, 0x62, 0x65, 0x00, 0x64, 0x00, 0x00,
0x00, 0x00, 0x01

```

**Table M.9 – stock counts arrays**

is\_ac == 0,stock\_index == 0:

```
0, 0, 3, 1, 1, 1, 1, 1, 1, 1, 1, 2, 0, 0, 0, 0, 0
```

is\_ac == 0,stock\_index == 1:

```
0, 0, 1, 5, 1, 1, 1, 1, 1, 2, 0, 0, 0, 0, 0, 0, 0
```

is\_ac == 1,stock\_index == 0:

```
0, 0, 2, 1, 3, 3, 2, 4, 3, 5, 5, 4, 4, 0, 0, 1, 126
```

is\_ac == 1,stock\_index == 1:

```
0, 0, 2, 1, 2, 4, 4, 3, 4, 7, 5, 4, 4, 0, 1, 2, 120
```

**Table M.10 – stock values arrays**

is\_ac == 0,stock\_index == 0:

```
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 256
```

is\_ac == 0,stock\_index == 1:

```
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 256
```

is\_ac == 1,stock\_index == 0:

```
1, 2, 3, 0, 4, 17, 5, 18, 33, 49, 65, 6, 19, 81,
97, 7, 34, 113, 20, 50, 129, 145, 161, 8, 35, 66, 177, 193,
21, 82, 209, 240, 36, 51, 98, 114, 130, 9, 10, 22, 23, 24,
25, 26, 37, 38, 39, 40, 41, 42, 52, 53, 54, 55, 56, 57,
58, 67, 68, 69, 70, 71, 72, 73, 74, 83, 84, 85, 86, 87,
88, 89, 90, 99, 100, 101, 102, 103, 104, 105, 106, 115, 116, 117,
118, 119, 120, 121, 122, 131, 132, 133, 134, 135, 136, 137, 138, 146,
147, 148, 149, 150, 151, 152, 153, 154, 162, 163, 164, 165, 166, 167,
168, 169, 170, 178, 179, 180, 181, 182, 183, 184, 185, 186, 194, 195,
196, 197, 198, 199, 200, 201, 202, 210, 211, 212, 213, 214, 215, 216,
217, 218, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 241, 242,
243, 244, 245, 246, 247, 248, 249, 250, 256
```

is\_ac == 1,stock\_index == 1:

```

0, 1, 2, 3, 17, 4, 5, 33, 49, 6, 18, 65, 81, 7,
97, 113, 19, 34, 50, 129, 8, 20, 66, 145, 161, 177, 193, 9,
35, 51, 82, 240, 21, 98, 114, 209, 10, 22, 36, 52, 225, 37,
241, 23, 24, 25, 26, 38, 39, 40, 41, 42, 53, 54, 55, 56,
57, 58, 67, 68, 69, 70, 71, 72, 73, 74, 83, 84, 85, 86,
87, 88, 89, 90, 99, 100, 101, 102, 103, 104, 105, 106, 115, 116,
117, 118, 119, 120, 121, 122, 130, 131, 132, 133, 134, 135, 136, 137,
138, 146, 147, 148, 149, 150, 151, 152, 153, 154, 162, 163, 164, 165,
166, 167, 168, 169, 170, 178, 179, 180, 181, 182, 183, 184, 185, 186,
194, 195, 196, 197, 198, 199, 200, 201, 202, 210, 211, 212, 213, 214,
215, 216, 217, 218, 226, 227, 228, 229, 230, 231, 232, 233, 234, 242,
243, 244, 245, 246, 247, 248, 249, 250, 256

```

**Table M.11 – predefined symbol order**

is\_ac == 0:

```

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

```

is\_ac == 1:

```

1, 0, 2, 3, 17, 4, 5, 33, 18, 49, 65, 6, 81, 19,
97, 7, 34, 113, 50, 129, 20, 145, 161, 8, 35, 66, 177, 193,
21, 82, 209, 240, 36, 51, 98, 114, 9, 130, 10, 22, 52, 225,
23, 37, 241, 24, 25, 26, 38, 39, 40, 41, 42, 53, 54, 55,
56, 57, 58, 67, 68, 69, 70, 71, 72, 73, 74, 83, 84, 85,
86, 87, 88, 89, 90, 99, 100, 101, 102, 103, 104, 105, 106, 115,
116, 117, 118, 119, 120, 121, 122, 131, 132, 133, 134, 135, 136, 137,
138, 146, 147, 148, 149, 150, 151, 152, 153, 154, 162, 163, 164, 165,
166, 167, 168, 169, 170, 178, 179, 180, 181, 182, 183, 184, 185, 186,
194, 195, 196, 197, 198, 199, 200, 201, 202, 210, 211, 212, 213, 214,
215, 216, 217, 218, 226, 227, 228, 229, 230, 231, 232, 233, 234, 242,
243, 244, 245, 246, 247, 248, 249, 250, 16, 32, 48, 64, 80, 96,
112, 128, 144, 160, 176, 192, 208, 11, 12, 13, 14, 15, 27, 28,
29, 30, 31, 43, 44, 45, 46, 47, 59, 60, 61, 62, 63, 75,
76, 77, 78, 79, 91, 92, 93, 94, 95, 107, 108, 109, 110, 111,
123, 124, 125, 126, 127, 139, 140, 141, 142, 143, 155, 156, 157, 158,
159, 171, 172, 173, 174, 175, 187, 188, 189, 190, 191, 203, 204, 205,
206, 207, 219, 220, 221, 222, 223, 224, 235, 236, 237, 238, 239, 251,
252, 253, 254, 255

```

**Table M.12 – stock quant tables**

is\_luma == true,stock\_index == 0:

```
3, 2, 2, 3, 5, 8, 10, 12, 2, 2, 3, 4, 5, 12, 12, 11, 3, 3,
3, 5, 8, 11, 14, 11, 3, 3, 4, 6, 10, 17, 16, 12, 4, 4, 7, 11,
14, 22, 21, 15, 5, 7, 11, 13, 16, 21, 23, 18, 10, 13, 16, 17, 21, 24,
24, 20, 14, 18, 19, 20, 22, 20, 21, 20
```

is\_luma == true,stock\_index == 1:

```
8, 6, 5, 8, 12, 20, 26, 31, 6, 6, 7, 10, 13, 29, 30, 28, 7, 7,
8, 12, 20, 29, 35, 28, 7, 9, 11, 15, 26, 44, 40, 31, 9, 11, 19, 28,
34, 55, 52, 39, 12, 18, 28, 32, 41, 52, 57, 46, 25, 32, 39, 44, 52, 61,
60, 51, 36, 46, 48, 49, 56, 50, 52, 50
```

is\_luma == true,stock\_index == 2:

```
6, 4, 4, 6, 10, 16, 20, 24, 5, 5, 6, 8, 10, 23, 24, 22, 6, 5,
6, 10, 16, 23, 28, 22, 6, 7, 9, 12, 20, 35, 32, 25, 7, 9, 15, 22,
27, 44, 41, 31, 10, 14, 22, 26, 32, 42, 45, 37, 20, 26, 31, 35, 41, 48,
48, 40, 29, 37, 38, 39, 45, 40, 41, 40
```

is\_luma == true,stock\_index == 3:

```
5, 3, 3, 5, 7, 12, 15, 18, 4, 4, 4, 6, 8, 17, 18, 17, 4, 4,
5, 7, 12, 17, 21, 17, 4, 5, 7, 9, 15, 26, 24, 19, 5, 7, 11, 17,
20, 33, 31, 23, 7, 11, 17, 19, 24, 31, 34, 28, 15, 19, 23, 26, 31, 36,
36, 30, 22, 28, 29, 29, 34, 30, 31, 30
```

is\_luma == true,stock\_index == 4:

```
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
```

is\_luma == true,stock\_index == 5:

```
2, 1, 1, 2, 2, 4, 5, 6, 1, 1, 1, 2, 3, 6, 6, 6, 1, 1,
2, 2, 4, 6, 7, 6, 1, 2, 2, 3, 5, 9, 8, 6, 2, 2, 4, 6,
7, 11, 10, 8, 2, 4, 6, 6, 8, 10, 11, 9, 5, 6, 8, 9, 10, 12,
12, 10, 7, 9, 10, 10, 11, 10, 10, 10
```

is\_luma == true,stock\_index == 6:

```
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1,
1, 2, 2, 3, 1, 1, 1, 1, 2, 2, 3, 3, 1, 1, 1, 2, 2, 3,
3, 3, 1, 1, 2, 2, 3, 3, 3, 3
```

is\_luma == true,stock\_index == 7:

```
10, 7, 6, 10, 14, 24, 31, 37, 7, 7, 8, 11, 16, 35, 36, 33, 8, 8,
10, 14, 24, 34, 41, 34, 8, 10, 13, 17, 31, 52, 48, 37, 11, 13, 22, 34,
41, 65, 62, 46, 14, 21, 33, 38, 49, 62, 68, 55, 29, 38, 47, 52, 62, 73,
72, 61, 43, 55, 57, 59, 67, 60, 62, 59
```

is\_luma == false,stock\_index == 0:

```
9, 9, 9, 12, 11, 12, 24, 13, 13, 24, 50, 33, 28, 33, 50, 50, 50, 50,
50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50,
50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50,
50, 50, 50, 50, 50, 50, 50, 50, 50, 50
```

is\_luma == false,stock\_index == 1:

```
3, 4, 5, 9, 20, 20, 20, 20, 4, 4, 5, 13, 20, 20, 20, 20, 5, 5,
11, 20, 20, 20, 20, 20, 9, 13, 20, 20, 20, 20, 20, 20, 20, 20, 20,
20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20,
20, 20, 20, 20, 20, 20, 20, 20, 20, 20
```

is\_luma == false,stock\_index == 2:

```
9, 9, 12, 24, 50, 50, 50, 50, 9, 11, 13, 33, 50, 50, 50, 50, 12, 13,
28, 50, 50, 50, 50, 50, 24, 33, 50, 50, 50, 50, 50, 50, 50, 50, 50,
50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50,
50, 50, 50, 50, 50, 50, 50, 50, 50, 50
```

is\_luma == false,stock\_index == 3:

```
5, 5, 7, 14, 30, 30, 30, 30, 5, 6, 8, 20, 30, 30, 30, 30, 7, 8,
17, 30, 30, 30, 30, 30, 14, 20, 30, 30, 30, 30, 30, 30, 30, 30, 30,
30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30,
30, 30, 30, 30, 30, 30, 30, 30, 30, 30
```

is\_luma == false,stock\_index == 4:

```
7, 7, 10, 19, 40, 40, 40, 40, 7, 8, 10, 26, 40, 40, 40, 40, 10, 10,
22, 40, 40, 40, 40, 40, 19, 26, 40, 40, 40, 40, 40, 40, 40, 40, 40,
40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,
40, 40, 40, 40, 40, 40, 40, 40, 40, 40
```

is\_luma == false, stock\_index == 5:

```
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
```

is\_luma == false, stock\_index == 6:

```
2, 2, 2, 5, 10, 10, 10, 10, 2, 2, 3, 7, 10, 10, 10, 10, 2, 3,
6, 10, 10, 10, 10, 10, 5, 7, 10, 10, 10, 10, 10, 10, 10, 10,
10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,
10, 10, 10, 10, 10, 10, 10, 10, 10, 10
```

is\_luma == false, stock\_index == 7:

```
10, 11, 14, 28, 59, 59, 59, 59, 11, 13, 16, 40, 59, 59, 59, 59, 14, 16,
34, 59, 59, 59, 59, 59, 28, 40, 59, 59, 59, 59, 59, 59, 59, 59,
59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59, 59,
59, 59, 59, 59, 59, 59, 59, 59, 59, 59
```

**Table M.13 – template quant tables**

is\_luma == true:

```
16, 11, 10, 16, 24, 40, 51, 61, 12, 12, 14, 19, 26, 58, 60,
55, 14, 13, 16, 24, 40, 57, 69, 56, 14, 17, 22, 29, 51, 87,
80, 62, 18, 22, 37, 56, 68, 109, 103, 77, 24, 35, 55, 64, 81,
104, 113, 92, 49, 64, 78, 87, 103, 121, 120, 101, 72, 92, 95, 98,
112, 100, 103, 99
```

is\_luma == false:

```
17, 18, 24, 47, 99, 99, 99, 99, 18, 21, 26, 66, 99, 99, 99, 99, 24, 26,
56, 99, 99, 99, 99, 99, 47, 66, 99, 99, 99, 99, 99, 99, 99, 99,
99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99,
99, 99, 99, 99, 99, 99, 99, 99, 99, 99
```

**Table M.15 – freq\_context**

scheme == 0:

```
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

scheme == 1:

```
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0
```

scheme == 2:

```
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1
```

scheme == 3:

```
0, 1, 1, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7,
7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 2, 2, 2
```

scheme == 4:

```
0, 1, 2, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 8, 8, 9, 9,
9, 9, 10, 10, 10, 10, 11, 11, 11, 11, 12, 12, 12, 12, 13, 13, 13, 13,
13, 13, 13, 13, 14, 14, 14, 14, 14, 14, 14, 14, 15, 15, 15, 15, 15, 15,
15, 15, 15, 15, 15, 15, 15, 15, 15, 15
```

scheme == 5:

```
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 16,
17, 17, 18, 18, 19, 19, 20, 20, 21, 21, 22, 22, 23, 23, 24, 24, 24, 24,
25, 25, 25, 25, 26, 26, 26, 26, 27, 27, 27, 27, 28, 28, 28, 28, 29, 29,
29, 29, 30, 30, 30, 30, 31, 31, 31, 31
```

scheme == 6:

```
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
54, 55, 56, 57, 58, 59, 60, 61, 62, 63
```

**Table M.16 – num\_nonzero\_context**

scheme == 0:

```
0, 1, 1, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7,
7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7
```

scheme == 1:

```
0, 2, 2, 4, 4, 4, 6, 6, 6, 6, 8, 8, 8, 8, 8, 8, 8, 10, 10,
10, 10, 10, 10, 10, 10, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12,
12, 12, 12, 12, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14,
14, 14, 14, 14, 14, 14, 14, 14, 14, 14
```

scheme == 2:

```
0, 4, 4, 8, 8, 8, 12, 12, 12, 12, 16, 16, 16, 16, 16, 16, 20, 20,
20, 20, 20, 20, 20, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24,
24, 24, 24, 24, 28, 28, 28, 28, 28, 28, 28, 28, 28, 28, 28, 28, 28, 28,
28, 28, 28, 28, 28, 28, 28, 28, 28, 28
```

scheme == 3:

```
0, 8, 8, 16, 16, 16, 24, 24, 24, 24, 32, 32, 32, 32, 32, 32, 40, 40,
40, 40, 40, 40, 40, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48, 48,
48, 48, 48, 48, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55,
55, 55, 55, 55, 55, 55, 55, 55, 55, 55
```

scheme == 4:

```
0, 16, 16, 32, 32, 32, 48, 48, 48, 48, 64, 64, 64, 64,
64, 64, 80, 80, 80, 80, 80, 80, 80, 80, 80, 80, 95, 95, 95, 95,
95, 95, 95, 95, 95, 95, 95, 95, 95, 95, 95, 95, 109, 109,
109, 109, 109, 109, 109, 109, 109, 109, 109, 109, 109, 109, 109, 109,
109, 109, 109, 109, 109, 109, 109, 109
```

scheme == 5:



```
0, 32, 32, 64, 64, 64, 96, 96, 96, 96, 127, 127, 127, 127,
127, 127, 157, 157, 157, 157, 157, 157, 157, 157, 157, 157, 185, 185, 185, 185,
185, 185, 185, 185, 185, 185, 185, 185, 185, 185, 185, 185, 185, 185, 211, 211,
211, 211, 211, 211, 211, 211, 211, 211, 211, 211, 211, 211, 211, 211, 211,
211, 211, 211, 211, 211, 211, 211, 211
```

scheme == 6:

```
0, 64, 64, 127, 127, 127, 188, 188, 188, 188, 246, 246, 246, 246,
246, 246, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 348, 348, 348, 348,
348, 348, 348, 348, 348, 348, 348, 348, 348, 348, 348, 348, 348, 348, 388, 388,
388, 388, 388, 388, 388, 388, 388, 388, 388, 388, 388, 388, 388, 388, 388,
388, 388, 388, 388, 388, 388, 388, 388
```

**Table M.17 – nonzero\_buckets**

```
0, 1, 2, 3, 4, 4, 5, 5, 5, 6, 6, 6, 6, 7, 7, 7, 7,
7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 9, 9,
9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 10, 10, 10, 10, 10, 10,
10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10
```

**Table M.29 – context\_modes table**

```
0, 1, 1, 1, 1, 1, 1, 1, 2, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0,
0, 2, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0,
0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0
```

```
0, 1, 1, 1, 1, 0, 0, 0, 2, 3, 1, 1, 1, 0, 0, 0, 2, 2, 0, 0, 0, 0, 0,
0, 2, 2, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```