برای پیاده‌سازی الگوریتم ساده‌اندیشانه حاصل ضرب دو عدد صحیح A و B، طبق تعریف باید هر رقم B را در هر رقم A ضرب کنیم و سپس طبق اندازه‌شان یا به تعبیر دیگر جایگاهشان با هم جمع می‌کنیم تا جواب حاصلضرب را پیدا کنیم. پیاده‌سازی این الگوریتم به شکل زیر خواهد بود :

```
def brute_force_multiply(x, y):
    strx = list(reversed(str(x)))
    stry = list(reversed(str(y)))
    y_length = len(stry)
    value = 0
    for n in range(len(strx)):
        xn = int(strx[n])
        for m in range(y_length):
            ym = int(stry[m])
            power = 10 ** (n + m)
            value += xn * ym * power
    return value
```

در اینجا ابتدا اعداد را به رشته تبدیل می‌کنیم تا استفاده از ارقام آن راحت‌تر باشد. برعکس کردن رشته‌ها به این دلیل است که اندیس هر رقم و ارزش جایگاهش یکسان باشند تا از انجام محاسبات اضافی جلوگیری شود.
برای پیاده سازی الگوریتم کاراتسوبا نیز کافی‌است مقادیر c_0، c_1 و c_2 را محاسبه کرده و آنهارا طبق ارزششان باهم جمع می‌کنیم

```
def karatsuba(a, b):
    if a < 10 and b < 10:
        return a * b
    else:
        a_str = str(a)
        b_str = str(b)
        n = max(len(a_str), len(b_str))
        a_str = "0" * (n - len(a_str)) + a_str
        b_str = "0" * (n - len(b_str)) + b_str
        m = n // 2
        a0 = int(a_str[m:])
        a1 = int(a_str[:m])

        b0 = int(b_str[m:])
        b1 = int(b_str[:m])

        c2 = karatsuba(a1, b1)
        c0 = karatsuba(a0, b0)
        c1 = karatsuba(a1 + a0, b1 + b0) - c2 - c0

        return c2 * 10 ** (2 * (n - m)) + c1 * 10 ** (n - m) + c0
```

در اینجا نیز مانند قسمت قبل ابتدا اعداد را به رشته تبدیل می‌کنیم تا کار با آنها راحت‌تر باشد سپس آنها را از نظر تعداد ارقام یکسان می‌کنیم تا هنکام تقسیم کردن آنها به دوقسمت، اعداد را با ترتیب درست در هم ضرب کنیم. پس محاسبه c_0، c_1 و c_2 کافیست حاصل را همانطور که در الگوریتم آمده است بدست آوریم. به جای اینکه در توان ها از m استفاده کنیم، به دلیل اینکه m می‌تواند عددی فرد باشد، از $n - m$ استفاده می‌کنیم زیرا تعداد صفرهای قسمت بزرگتر برابر این عدد است.