

Answer on Question #62642 - Programming & Computer Science - C

Question 62642:

A game is played with three piles of stones and two players. At her turn, a player removes one or more stones from the piles. However, if she takes stones from more than one pile, she must remove the same number of stones from each of the selected piles.

In other words, the player chooses some $N > 0$ and removes:

- N stones from any single pile; or
- N stones from each of any two piles ($2N$ total); or
- N stones from each of the three piles ($3N$ total).

Find $\sum(x_i + y_i + z_i)$ where (x_i, y_i, z_i) ranges over the losing configurations with $x_i \leq y_i \leq z_i \leq 1000$.

Answer:

```
#include <stdio.h>
#include <stdbool.h>

void subset(size_t a, size_t b, size_t c);
void swap(size_t *p1, size_t *p2);

#define N__ (1000 + 1)
bool w[N__][N__][N__];

int main(void)
{
    size_t qsum = 0;
    const size_t n = N__ - 1;
    size_t i, j, k, x;

    for (i = 0; i <= n; ++i)
        for (j = 0; j <= i; ++j)
            for (k = 0; k <= j; ++k)
                if (!w[i][j][k]) {
                    for (x = 1; x <= n - i; ++x)
                        subset(i + x, j, k);
                    for (x = 1; x <= n - j; ++x)
                        subset(i, j + x, k);
                    for (x = 1; x <= n - k; ++x)
                        subset(i, j, k + x);
                    for (x = 1; x <= n - i && x <= n - j; ++x)
                        subset(i + x, j + x, k);
```

```

        for (x = 1; x <= n - j && x <= n - k; ++x)
            subset(i, j + x, k + x);
        for (x = 1; x <= n - k && x <= n - i; ++x)
            subset(i + x, j, k + x);
        for (x = 1; x <= n - i && x <= n - j && x <= n - k; ++x)
            subset(i + x, j + x, k + x);
        qsum += i + j + k;
    }

/* 213 for 10, 173895 for 100, 167542057 for 1000 */
printf("Answer: %lu\n", (unsigned long) qsum);

return 0;
}

void subset(size_t a, size_t b, size_t c)
{
    if (a < b) swap(&a, &b);
    if (b < c) swap(&b, &c);
    if (a < b) swap(&a, &b);

    w[a][b][c] = true;

    return;
}

void swap(size_t *p1, size_t *p2)
{
    size_t t;

    t = *p1; *p1 = *p2; *p2 = t;

    return;
}

```