ABC 160E Red and Green Apples

You are going to eat x green apples and y red apples. You have a green apples, each with g_i tastiness, b red apples, each with r_i tastiness, and c white apples, each with w_i tastiness. You can paint a white apple either red or green. Find the maximum total tastiness.

$x \leq a \quad y \leq b \quad a, b, c \leq 10^5$

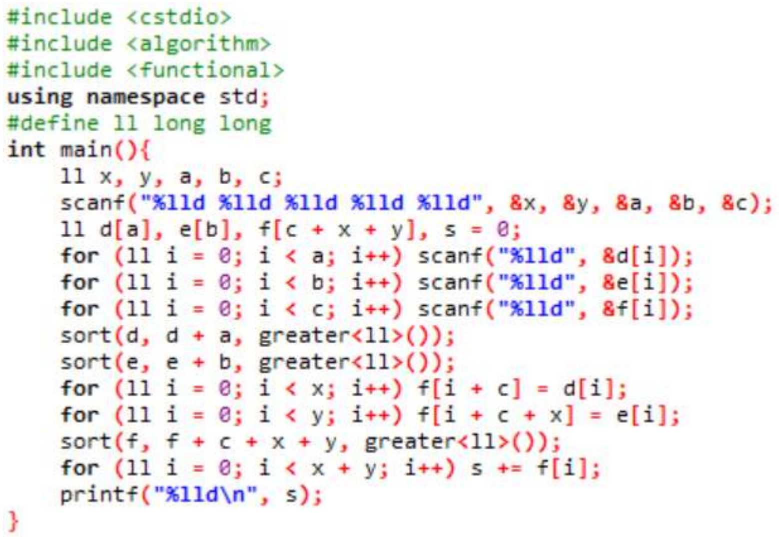
Try to think of a solution before reading on!

Firstly, you see the constraint $\leq 10^5$. This gives you a clue the that solution should be $O(n)$ or $O(n \log n)$, and sorting is likely used.

Let us sort the arrays g and r (in descending order). Since $x \leq a$ and $y \leq b$, we only need to use the first x elements of g and the first y elements of r.

How do we deal with the write apples? Let's say we pick $x + y$ apples out of the $a + b + c$ apples. Then, we see that any combination is valid as long as we picked $\leq x$ green apples and $\leq y$ red apples because we can just paint the white apples we picked green or red! For example, if $x = 3$ and $y = 2$, and we chose 2 green apples, 1 red apple and 2 white apples, we can just paint 1 of the white apples green and another red.

Therefore, the solution is as follows: pick the x tastiest apples out of g, the y tastiest apples out of r, and the c white apples, and put them into an array of size $x + y + c$. Then, sort them and just take the $x + y$ tastiest apples!



```
#include <cstdio>
#include <algorithm>
#include <functional>
using namespace std;
#define ll long long
int main(){
    ll x, y, a, b, c;
    scanf("%lld %lld %lld %lld %lld", &x, &y, &a, &b, &c);
    ll d[a], e[b], f[c + x + y], s = 0;
    for (ll i = 0; i < a; i++) scanf("%lld", &d[i]);
    for (ll i = 0; i < b; i++) scanf("%lld", &e[i]);
    for (ll i = 0; i < c; i++) scanf("%lld", &f[i]);
    sort(d, d + a, greater<ll>());
    sort(e, e + b, greater<ll>());
    for (ll i = 0; i < x; i++) f[i + c] = d[i];
    for (ll i = 0; i < y; i++) f[i + c + x] = e[i];
    sort(f, f + c + x + y, greater<ll>());
    for (ll i = 0; i < x + y; i++) s += f[i];
    printf("%lld\n", s);
}
```