Choice of Area

```
#include <bits/stdc++.h>

using namespace std;

struct Area {
    int A;
    int B;
};

int getMaxSurvivalTime(int powerA, int powerB, Area &X, Area &Y, Area &Z, int lastArea) {
    if (powerA <= 0 || powerB <= 0) {
        return 0;
    }
    int count = 0;
    switch (lastArea) {
        case 1:
            count = 1 + max(getMaxSurvivalTime(powerA + Y.A, powerB + Y.B, X, Y, Z, 2),
                    getMaxSurvivalTime(powerA + Z.A, powerB + Z.B, X, Y, Z, 3));
            break;
        case 2:
            count = 1 + max(getMaxSurvivalTime(powerA + X.A, powerB + X.B, X, Y, Z, 1),
                            getMaxSurvivalTime(powerA + Z.A, powerB + Z.B, X, Y, Z, 3));
            break;
        case 3:
            count = 1 + max(getMaxSurvivalTime(powerA + X.A, powerB + X.B, X, Y, Z, 1),
                            getMaxSurvivalTime(powerA + Y.A, powerB + Y.B, X, Y, Z, 2));
            break;
        default:
            break;
    }
    return count;
}

int maxSurvivalTime(int powerA, int powerB, Area &X, Area &Y, Area &Z) {
    if (powerA <= 0 || powerB <= 0)
        return 0;
    return max(getMaxSurvivalTime(powerA + X.A, powerB + X.B, X, Y, Z, 1),
               max(getMaxSurvivalTime(powerA + Y.A, powerB + Y.B, X, Y, Z, 2),
                   getMaxSurvivalTime(powerA + Z.A, powerB + Z.B, X, Y, Z, 3)));

}

int main() {
    int powerA = 20, powerB = 8;
    Area X = {3, 2}, Y = {-5, -10}, Z = {-20, 5};
    printf("%d\n", maxSurvivalTime(powerA, powerB, X, Y, Z));
}
```