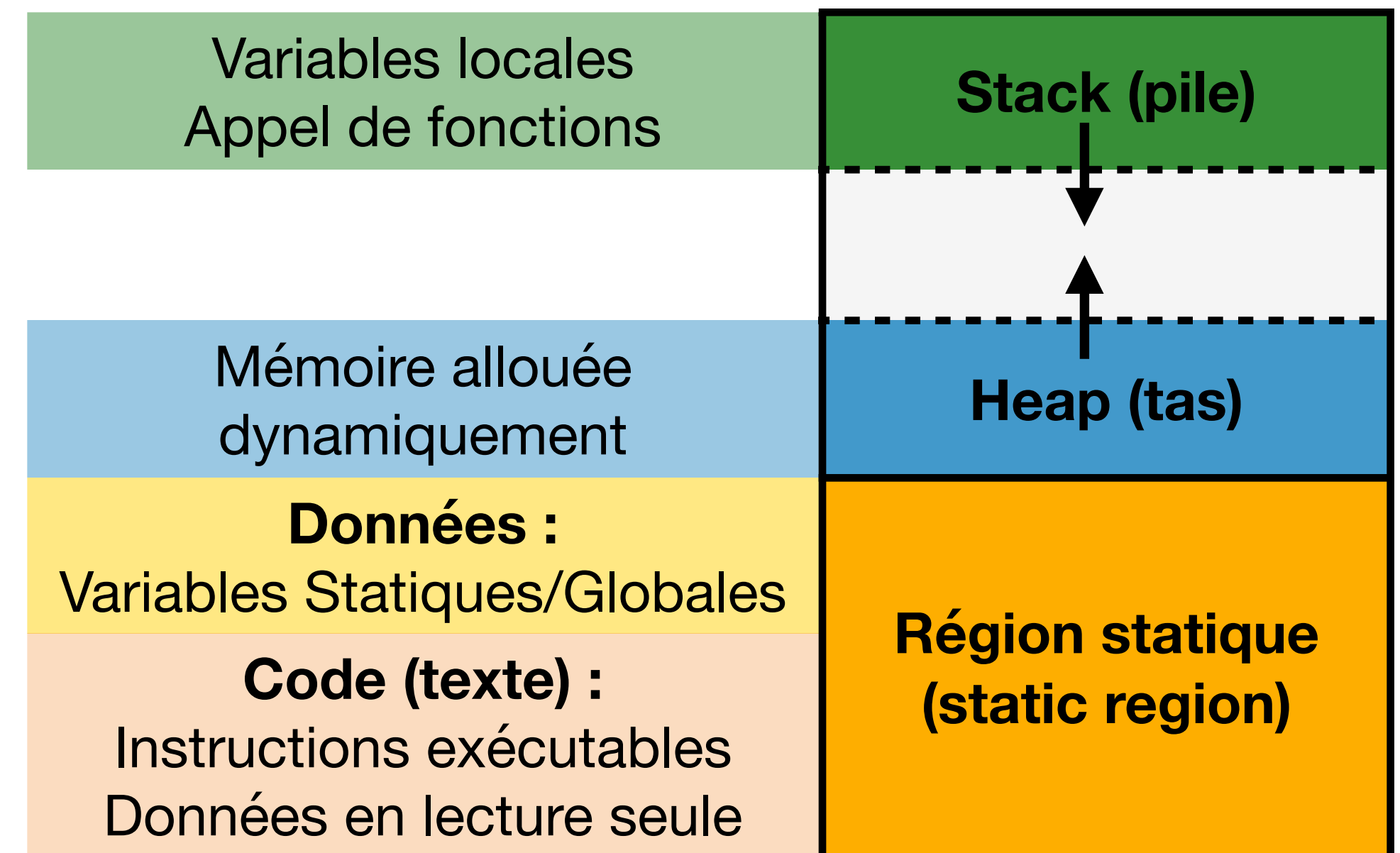


- **La région statique (static region)**, dans laquelle sont stockées les variables statiques et globales ainsi que les instructions du programme.
 - Disponible tout au long de l'exécution d'un programme.
 - Sa taille ne change pas pendant l'exécution du programme C++.
 - Lecture seule.
- **La pile (stack)**, dans laquelle sont stockées les trames de pile. Une nouvelle trame de pile est créée pour chaque appel de fonction (et ses variables locales).
 - Détruite lorsque cette fonction a été exécutée.
- **Le tas (heap)**, dans lequel la mémoire allouée dynamiquement est stockée.

Pour optimiser l'utilisation de la mémoire, le tas et la pile se développent généralement l'un vers l'autre, comme illustré dans la figure ci-contre.



```

#include <iostream>
using namespace std;

int total;

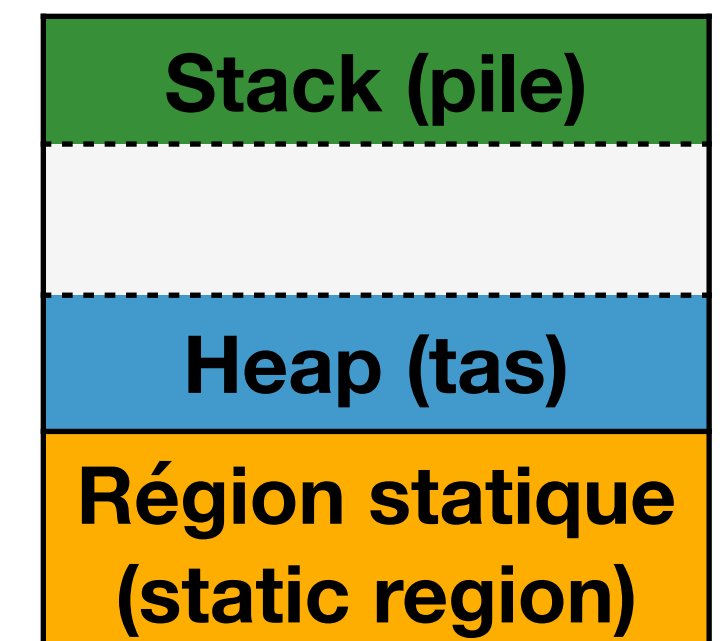
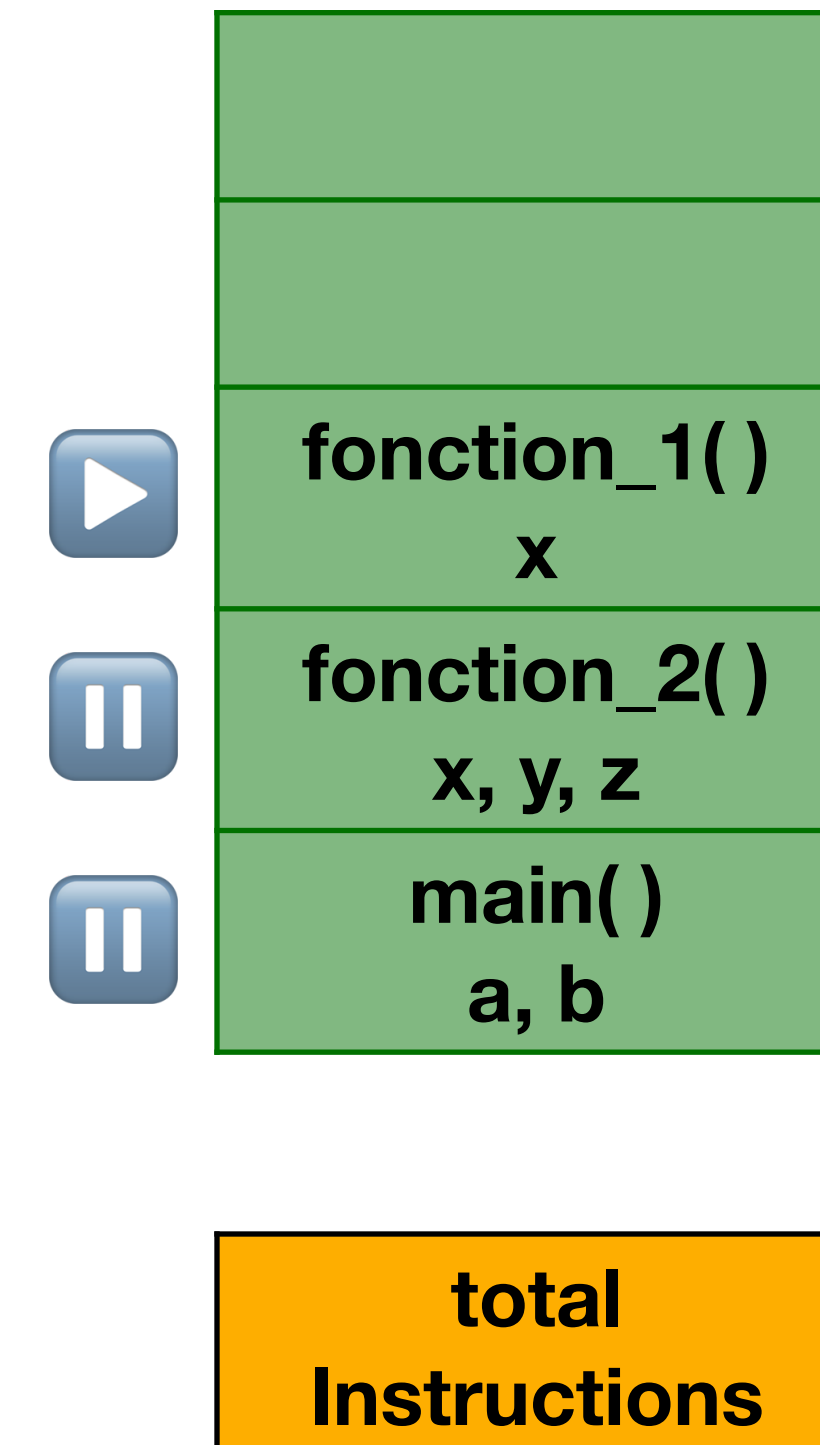
int fonction_1(int x) { // x²
    return x*x;
}

int fonction_2(int x, int y) { //(x+y)²
    int z = fonction_1(x+y);
    return z;
}

int main() {
    int a = 4, b = 8;
    total = fonction_2(a,b);
    cout << "Résultat : " << total << endl;

    return 0;
}

```



```

#include <iostream>
using namespace std;

int total;

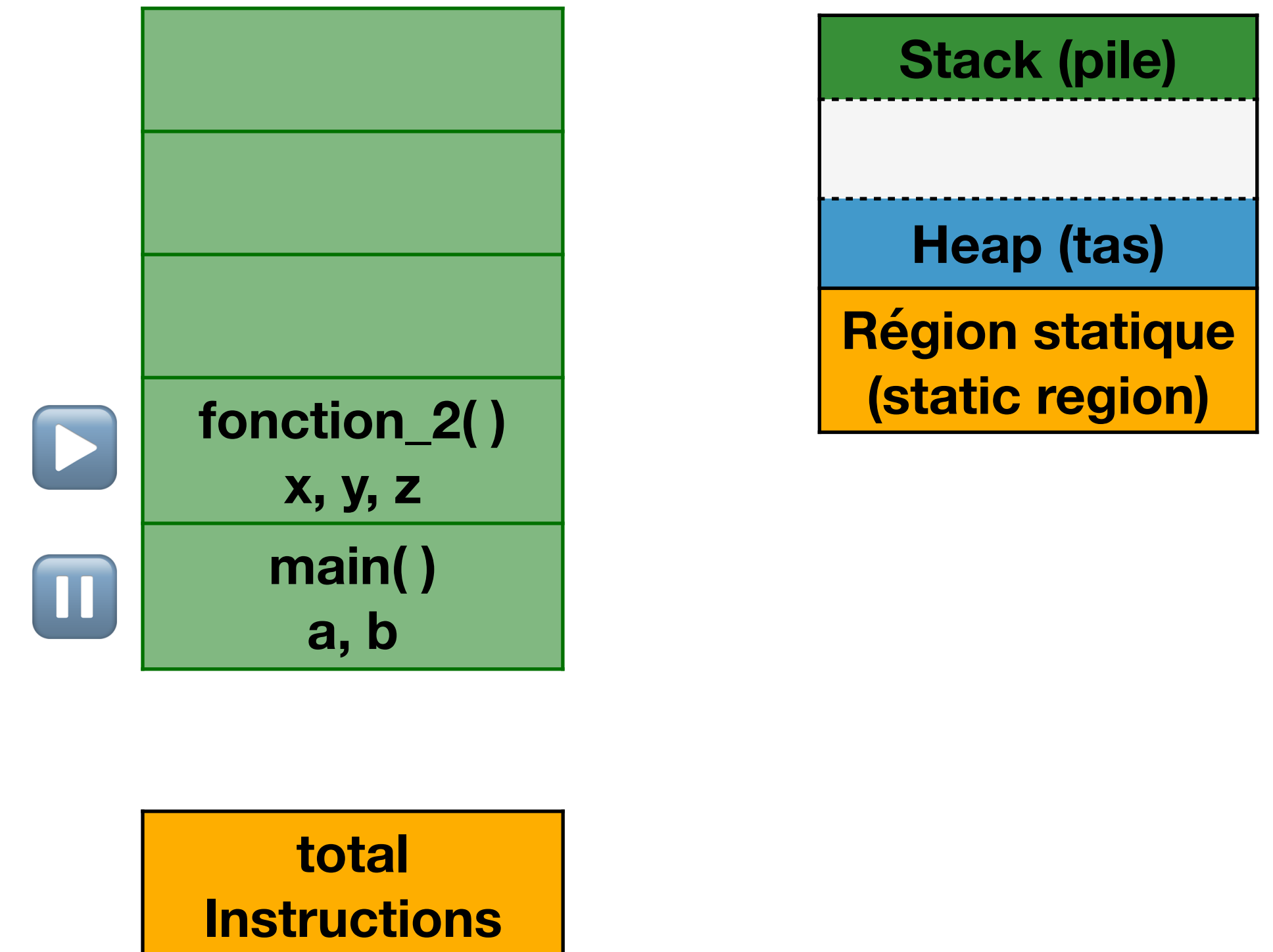
int fonction_1(int x) { // x²
    return x*x;
}

int fonction_2(int x, int y) { //(x+y)²
    int z = fonction_1(x+y);
    return z;
}

int main() {
    int a = 4, b = 8;
    total = fonction_2(a,b);
    cout << "Résultat : " << total << endl;

    return 0;
}

```



```

#include <iostream>
using namespace std;

int total;

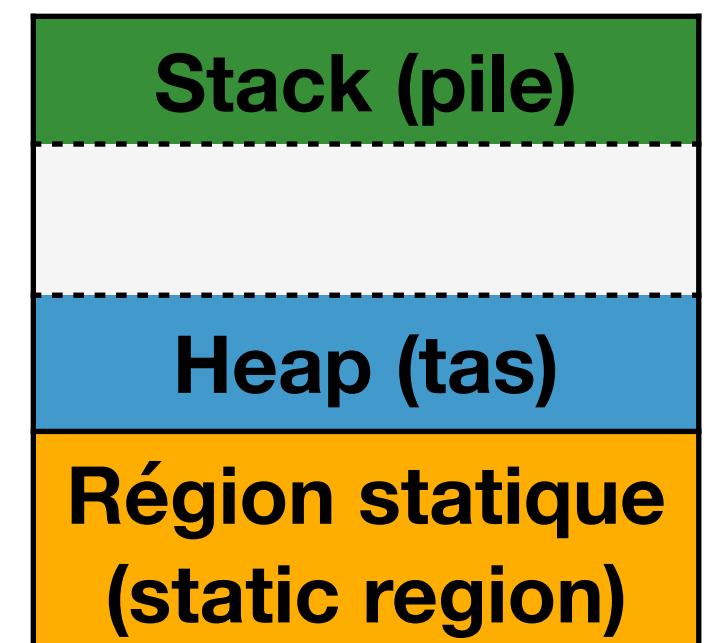
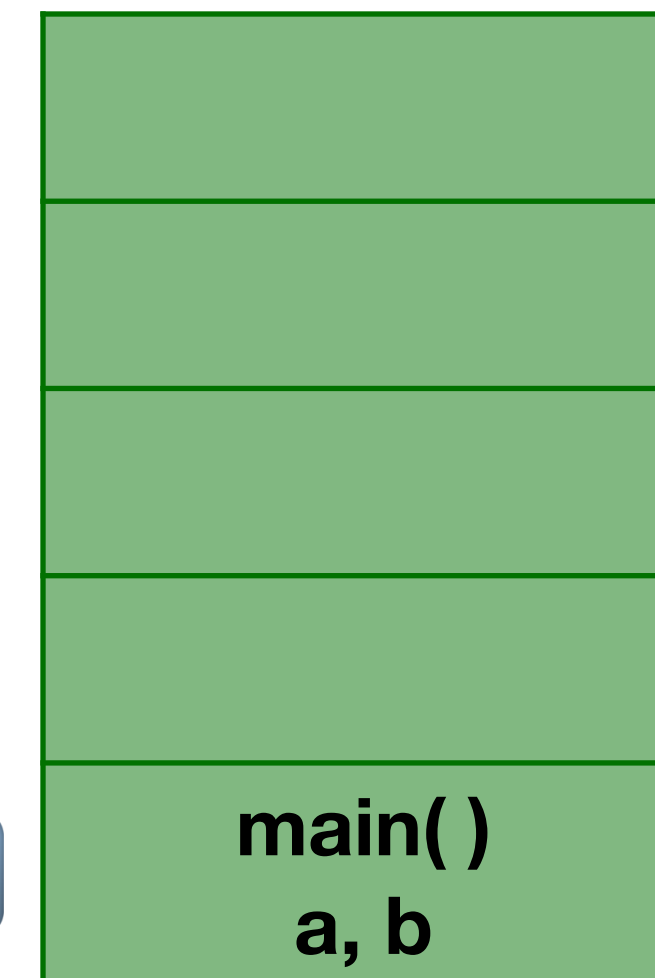
int fonction_1(int x) { // x²
    return x*x;
}

int fonction_2(int x, int y) { //(x+y)²
    int z = fonction_1(x+y);
    return z;
}

int main() {
    int a = 4, b = 8;
    total = fonction_2(a,b);
    cout << "Résultat : " << total << endl;

    return 0;
}

```



```

#include <iostream>
using namespace std;

int total;

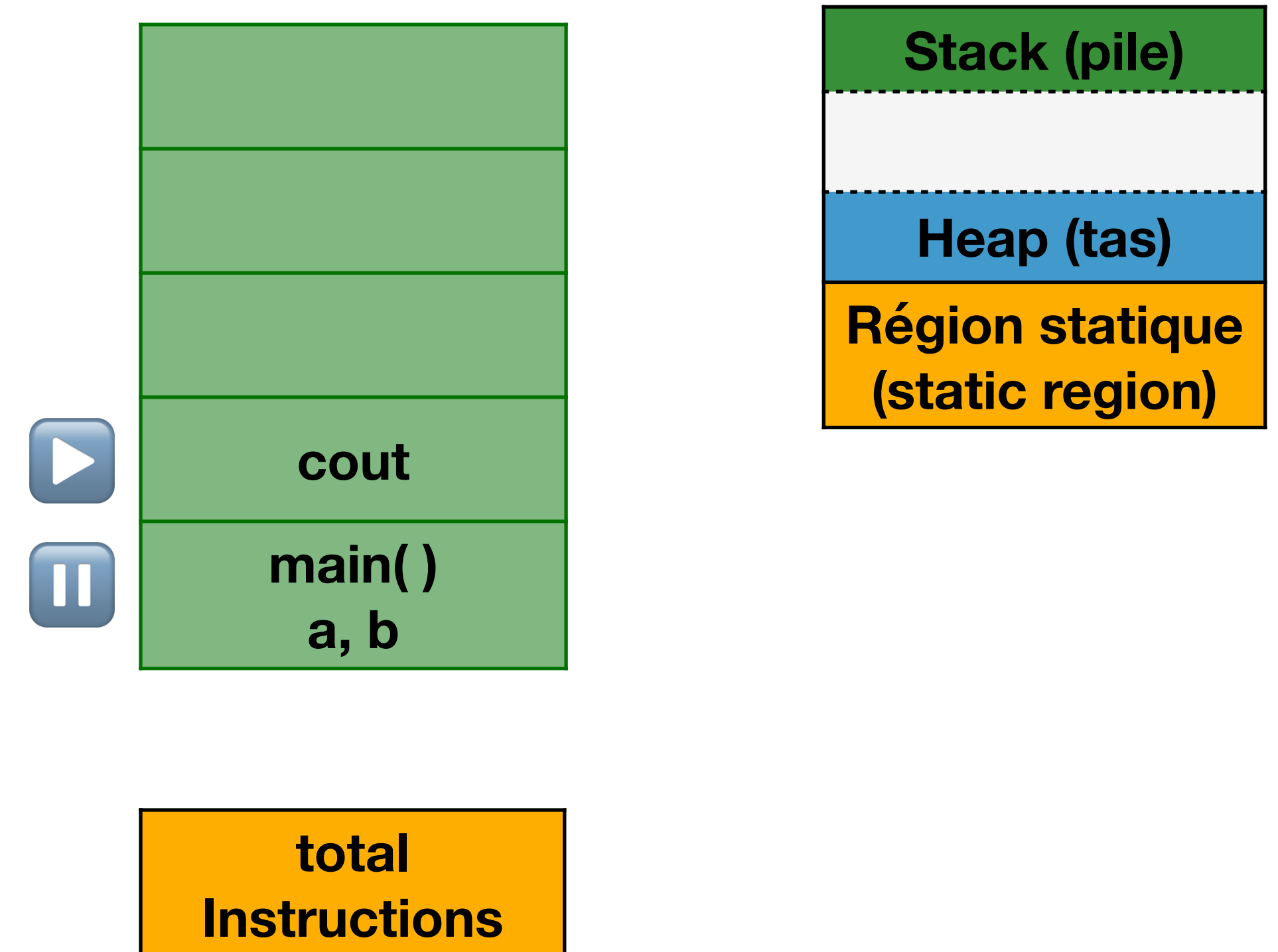
int fonction_1(int x) { // x²
    return x*x;
}

int fonction_2(int x, int y) { //(x+y)²
    int z = fonction_1(x+y);
    return z;
}

int main() {
    int a = 4, b = 8;
    total = fonction_2(a,b);
    cout << "Résultat : " << total << endl;

    return 0;
}

```



```

#include <iostream>
using namespace std;

int total;

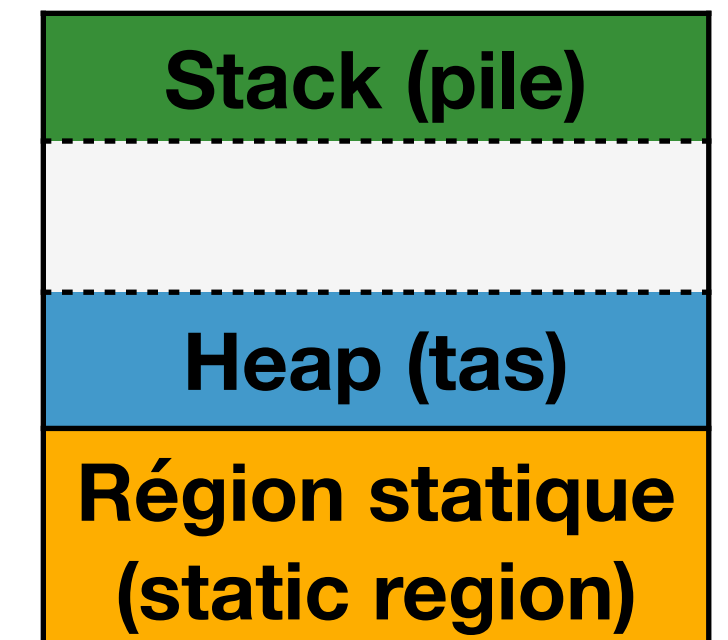
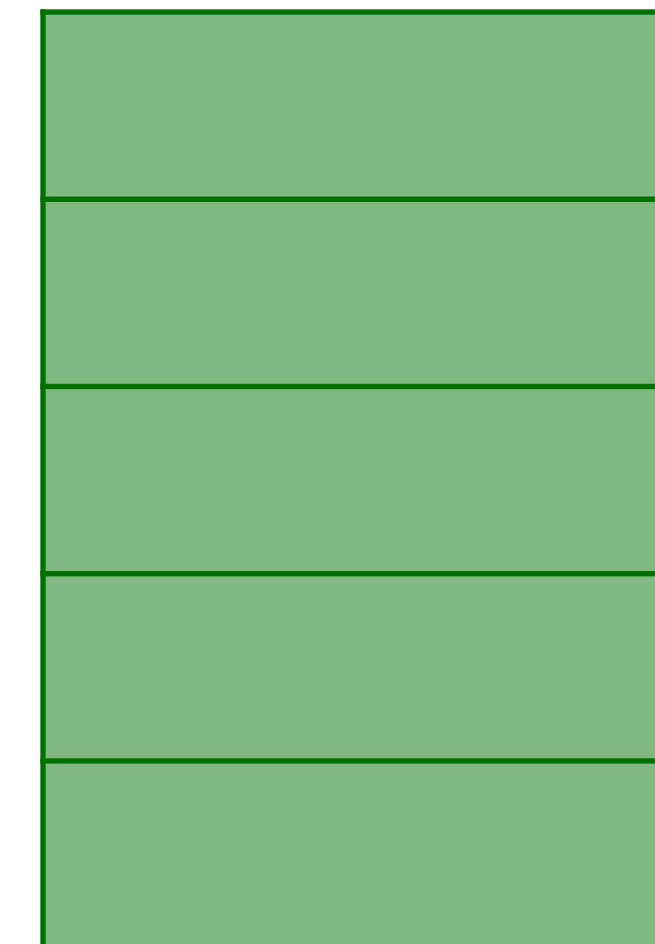
int fonction_1(int x) { // x²
    return x*x;
}

int fonction_2(int x, int y) { //(x+y)²
    int z = fonction_1(x+y);
    return z;
}

int main() {
    int a = 4, b = 8;
    total = fonction_2(a,b);
    cout << "Résultat : " << total << endl;

    return 0;
}

```



```

#include <iostream>
using namespace std;

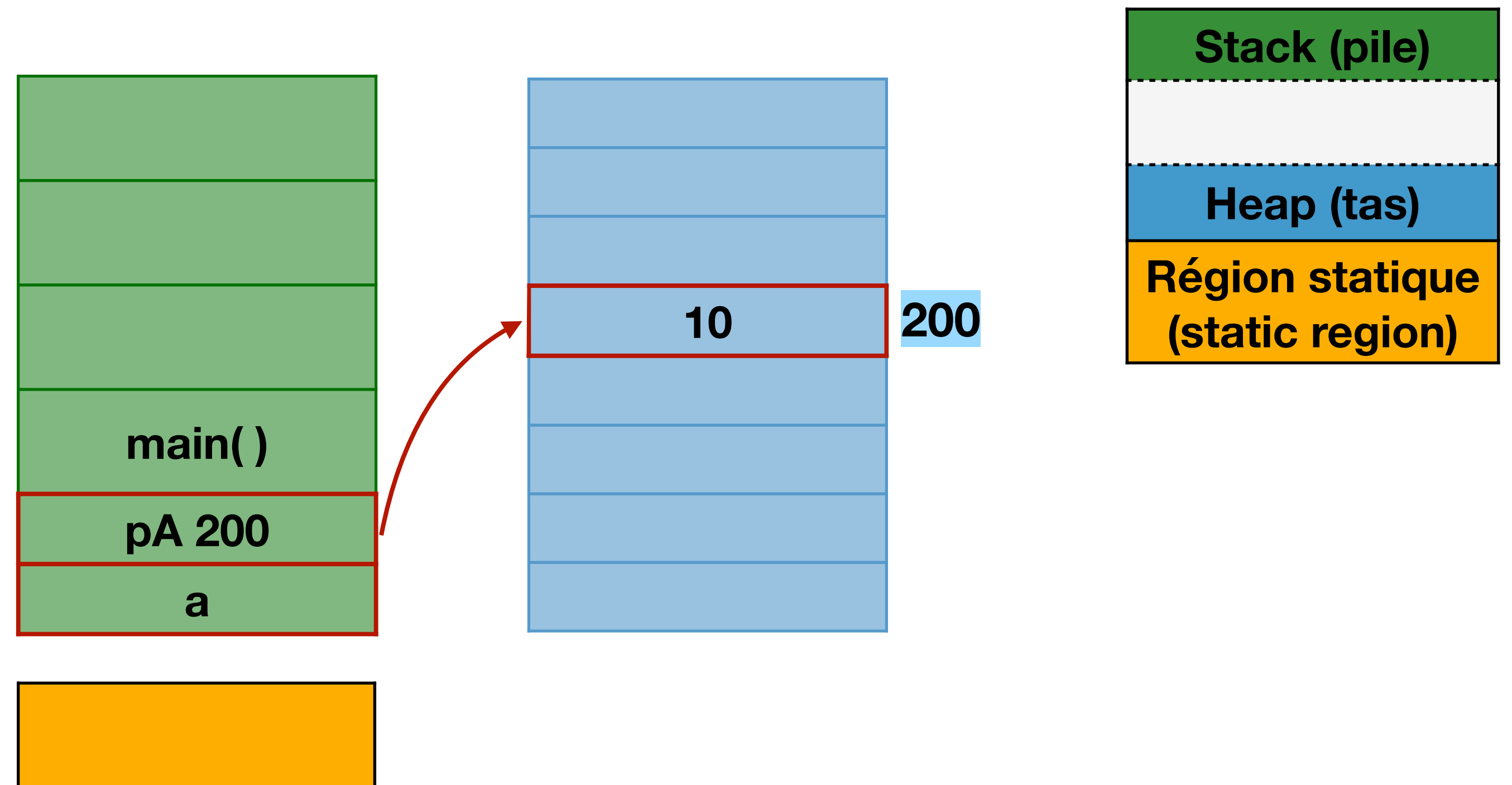
int main() {
    int a;
    int *pA;
    pA = new int;
    *pA = 10;
    delete pA;

    pA = new int;
    *pA = 20;
    delete pA;

    pA = new int[20];
    delete[] pA;

    return 0;
}

```



```

#include <iostream>
using namespace std;

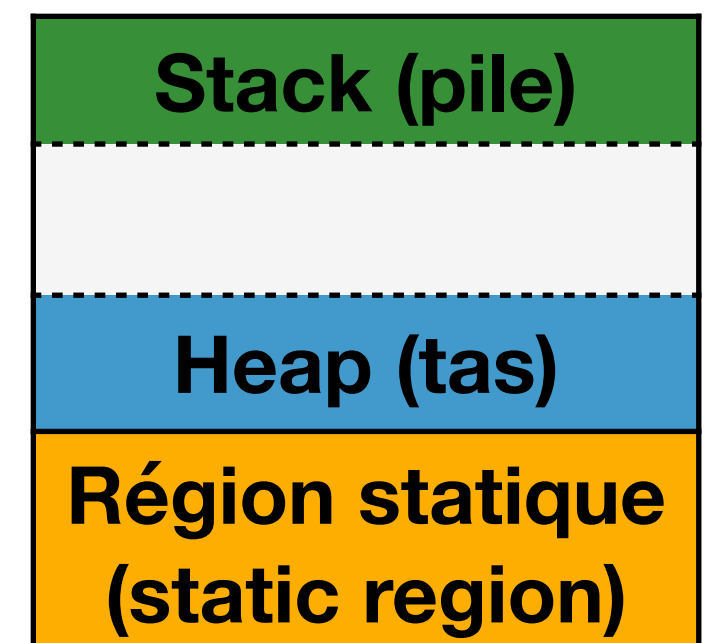
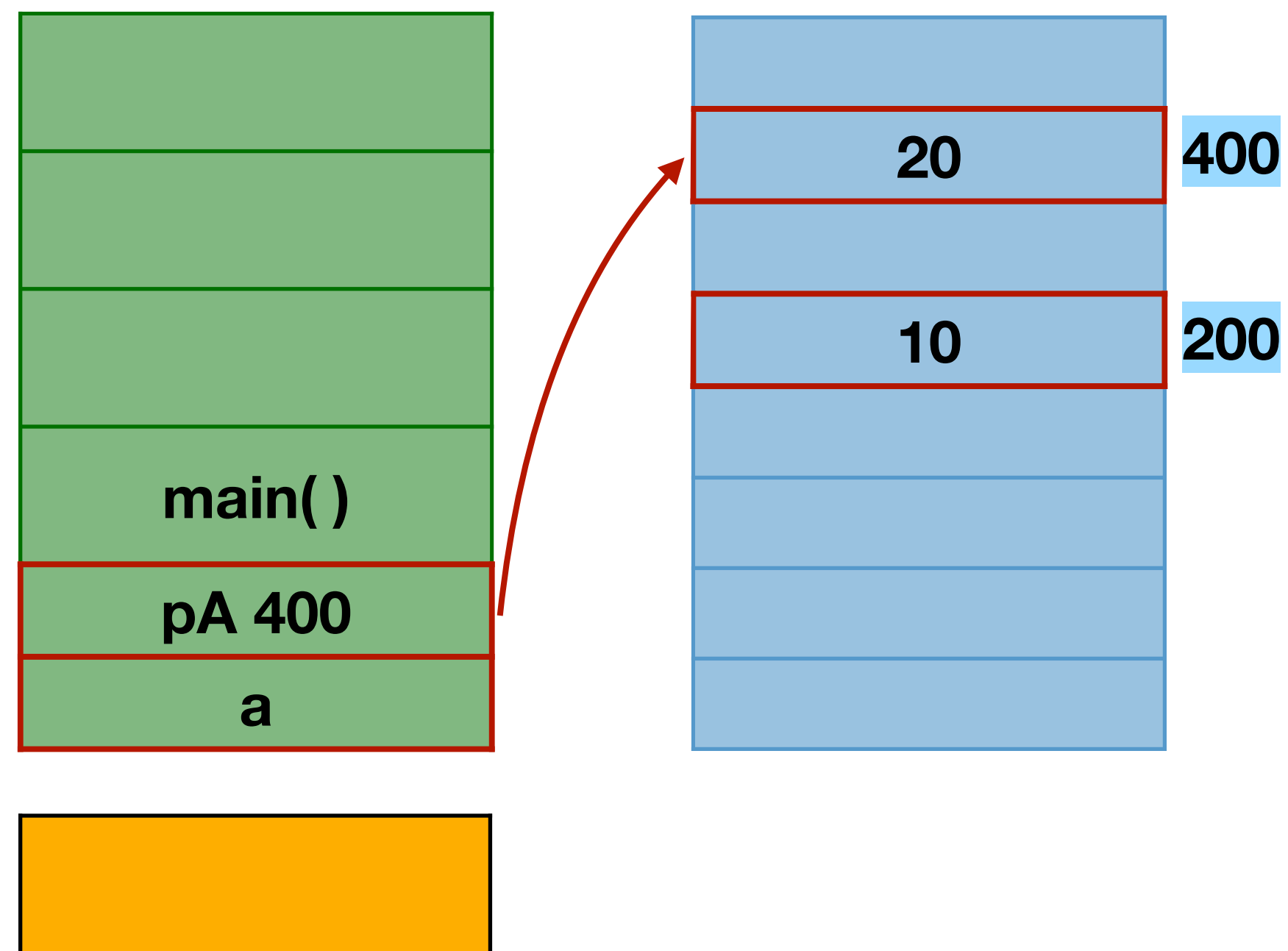
int main() {
    int a;
    int *pA;
    pA = new int;
    *pA = 10;
    delete pA;

    pA = new int;
    *pA = 20;
    delete pA;

    pA = new int[20];
    delete[] pA;

    return 0;
}

```




```

#include <iostream>
using namespace std;

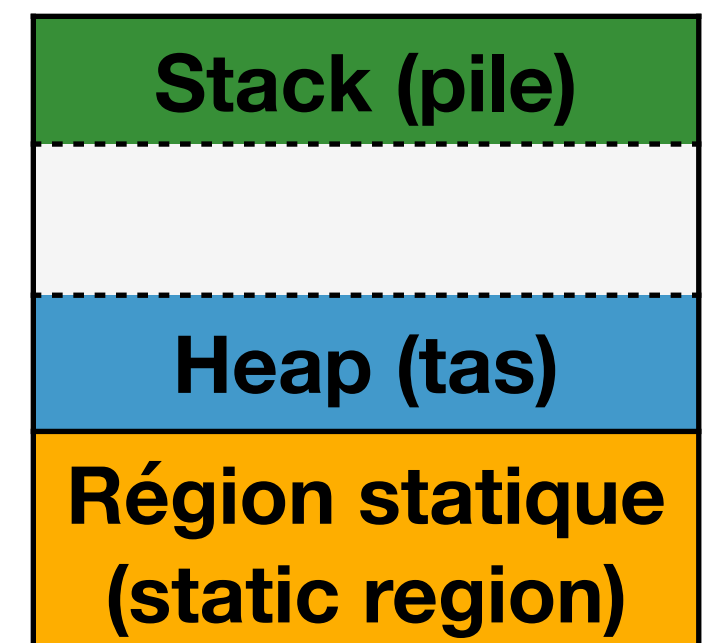
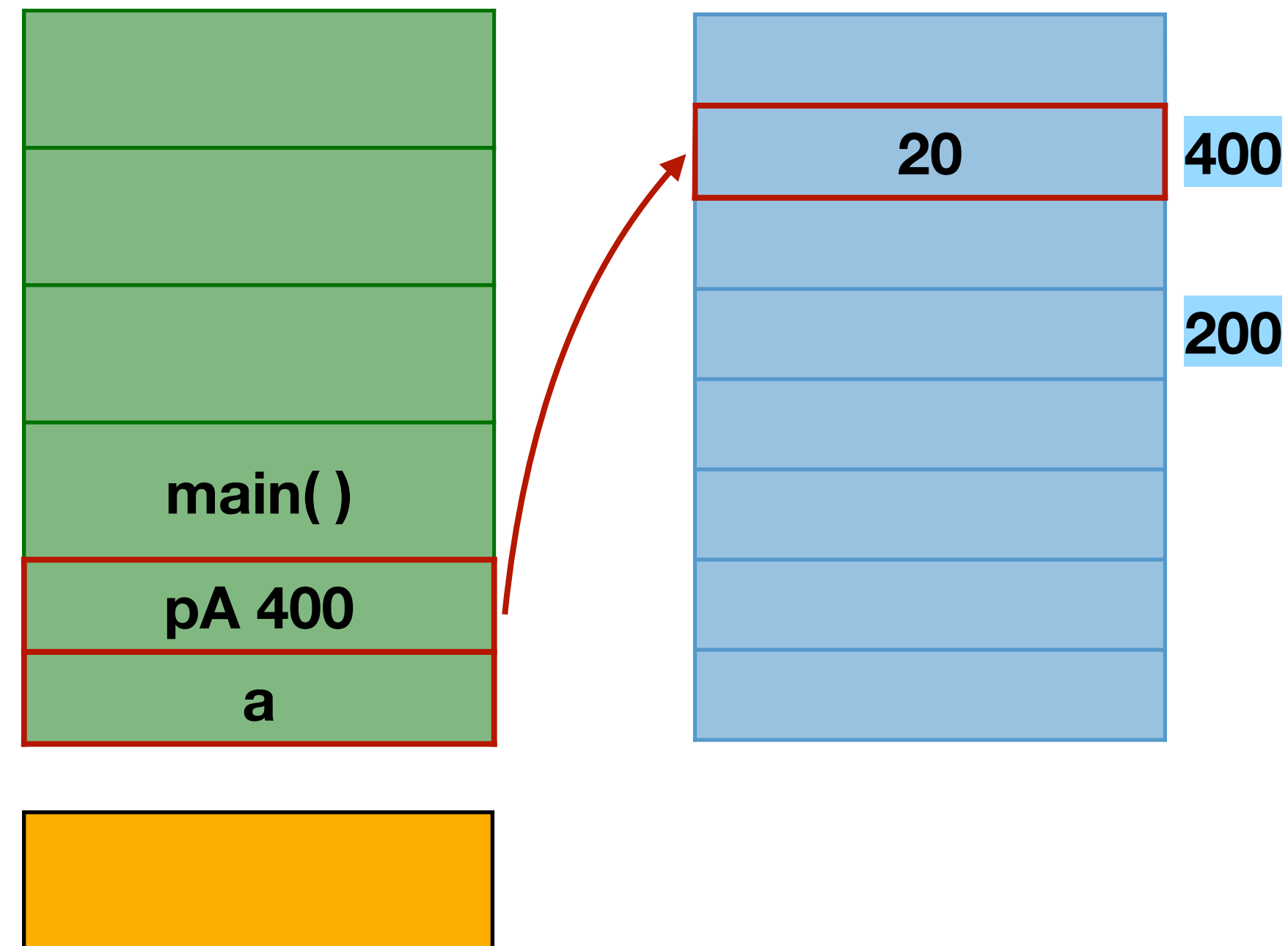
int main() {
    int a;
    int *pA;
    pA = new int;
    *pA = 10;
    delete pA;

    pA = new int;
    *pA = 20;
    delete pA;

    pA = new int[20];
    delete[] pA;

    return 0;
}

```



```

#include <iostream>
using namespace std;

int main() {
    int a;
    int *pA;
    pA = new int;
    *pA = 10;
    delete pA;

    pA = new int;
    *pA = 20;
    delete pA;

    pA = new int[20];
    delete[] pA;

    return 0;
}

```

