**5.Реалізація викликів функцій за допомогою стеку. Означення стеку (дисципліна, структура даних, складові). Пояснити роль стеку у здійсненні викликів функцій і повертання у програму, що викликала. Приклади.**

Семантика виклику функції реалізується за допомогою структури даних СТЕК.

Стек - структура даних, дисципліна якої визначена двома операціями: push(x)-вштовхнути в стек, та pop(x)-виштовхнути із стеку. Стек представлений у вигляді одновимірного масиву. Стек-це

структура даних, яка працює за

- Елементи масиву.

принципом FILO (first in — last out;

перший прийшов — останній пішов).

У стеку елемент, який увійшов

-bottom

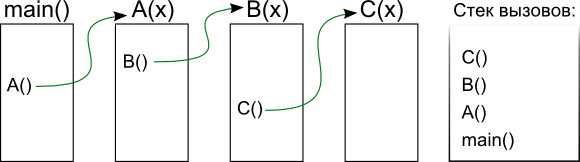
найперший-вийде самим останнім.

Виходить, якщо ви додали три елементи в стек першим буде видалений Останній доданий елемент.

Стек викликів — (call stack) - стек, що зберігає інформацію для повернення управління з підпрограм (функцій) в програму або підпрограму (при вкладених або рекурсивних викликах).

При виклику підпрограми в стек заноситься адреса повернення-адреса в пам'яті наступної інструкції призупиняється програма, а управління передається підпрограмі. При подальшому вкладеному або рекурсивному виклику в стек заноситься чергова адреса повернення і так далі.

При поверненні з підпрограми адреса повернення знімається зі стека, і управління передається на наступну інструкцію призупиненої програми (або підпрограми).



*Приклад: 1.* «Stackoverflow» - виконання програми зупиниться :стек не може зростати нескінченно.

void f(){f();}

int main(){f();}

2. Сума пос-ті без використання масиву

int main { cout<<sum();}-при кожному виклику 2 змінні вштовхуються в стек.

int sum(){int a; if(cin>>a) return a+sum(); return 0;}