Intersection of Sets

$A = \{1, 2, 3\}$

$B = \{3, 4\}$

$A \cap B = \{3\}$

```
void  intersect (vector<int>& a, vector<int> &b          )
{      vector<int> c;
       for (int  i=0;  i<a.size();  ++i)  {
            for (int  j=0;  j<b.size();  ++j) {
                 if ( a[i] == b[j])   {
                      // Store  in  the  resulting  array.
                 }     c.push_back(a[i]);
            }
       }
}
```

```
bool  exists(vector<int>&  v,  int q)
{
       for (int  i=0;  i< v.size();  ++i)  {
            if ( v[i] == q)  {
                 return  true;
            }
       }
       return  false;
}
```

```
void  intersect ( vector<int>& a, vector<int> & b)
{
      vector<int> c;
      for (int i=0;  i<a.size() ; ++i)   {
            if ( exists (b, a[i])) {
                c.push_back(a[i]);
            }
      }
}
```

Union of Two Sets

$A = \{1,2,3\}$

$B = \{3,4\}$

$A \cup B = \{1,2,3,4\}$

```
void   union1 ( vector<int>& a,  vector<int>& b)
{
      vector <int>  c;
      for (int  i=0;  i< a.size();  ++i)  {
            c.push_back(a[i]);
      }
      for (int  i=0;  i< b.size();  ++i) {
            if (exists(a, b[i])) {
                continue;
            }
            else  {
```

```
for(int i:a)
{

}
```

```
            c.push_back(b[i])
        }
    }
}
```

Find a Sub-string

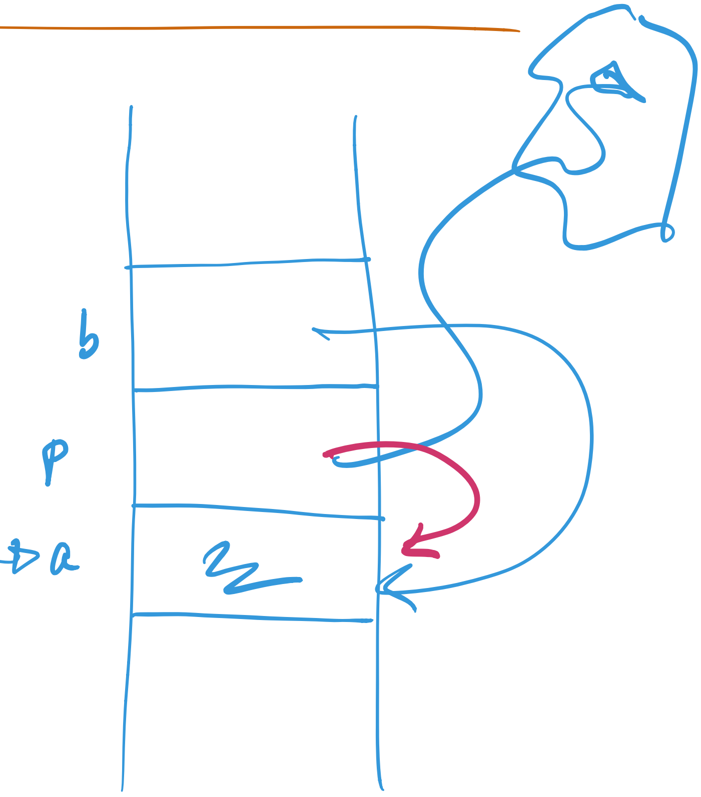
```
      0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
str:  i   a m   t e a c h i  n  g     c  +  +  .
q  :  a m
```

&str[10]

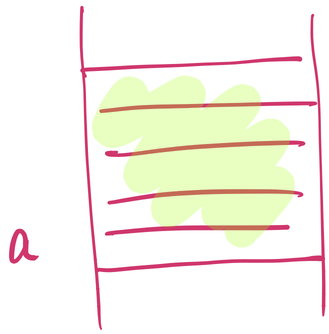
(str+10)

```
bool match (char *s, char *q, int lenq)
{
    for (int i=0; i< lenq; ++i)
    {
        if (q[i] != s[i]) {
            return false;
        }
    }
    return true;
}

int find(char* s, int lens, char* q, int lenq)
{


    for (int i=0; i< lens-lenq; ++i) {
        if (matches(& S[i], q, lenq) {
            return i;
        }
      }
    return -1;
}
```

```
int a;
int * p;

&a

int (&b) = a;

p = &a;
```



```
cout << &a << endl;
cout << a << endl;
```



Longest Sub Sequence

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
a a a 3 d a b d e x x  x  x  3  3  j    → 16
```

letter = x

location = 9

length = 4

```
void ss(char* s, int lens, char& c, int& p, int& l)
{
    int current_best_len = 1;
    int current_best_pos = 0;
    char current_best_ch = s[0];

    int l1 = 1;
    int p1 = 0;
    int c1 = s[0];

    for (int i=1; i<lens; ++i) {
        if (s[i] == c1) {
            ++l1;
            if (l1 > current_best_len) {
                current_best_len = l1;
                current_best_Char = c1;
                current_best_pos = p1;
        }
        } continue.
        c1 = s[i];
        l1 = 1;
        p1 = 0;
    }
```

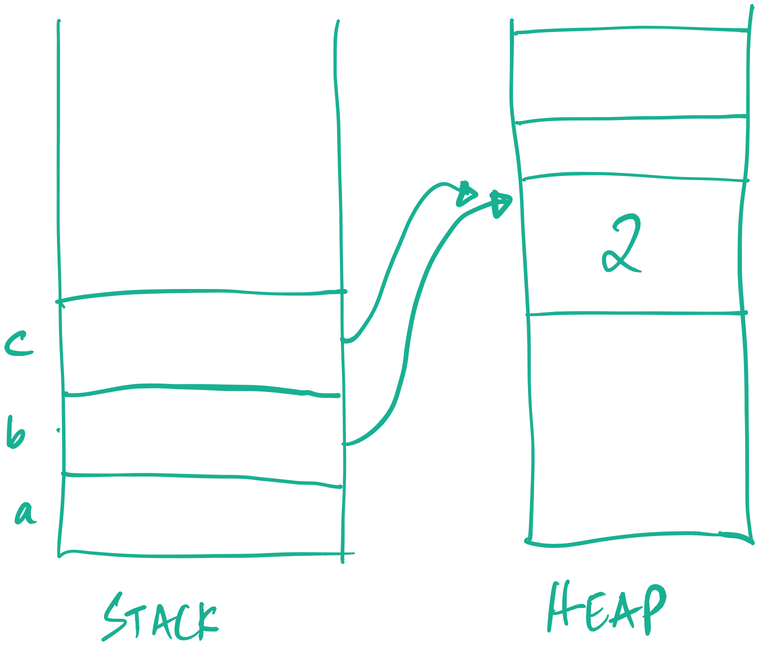
```
cout << a;
cin >> b;

ostream& operator<<(ostream&, ...)
```

```
class A {
   void foo();
   void boo() const;
};

A a;
```

```
a.foo();
a.boo();
```

```
const A& b = a;
b.foo(); X
b.boo();
```

```
const int& operator[](int i) const { ... }
a[0] = 33;   X
```


c
b
a
2
STACK
HEAP