Pointers

A pointer to a variable is the address in the memory of that variable.

```
int ten = 10;         // declare an integer variable
int *p = &ten;        // get a pointer to the variable


cout << p << end;     // print the pointer (memory address)


cout << *p << endl;   // print the value the pointer points to
                      // (dereferencing the pointer p)
```

&x is the address of the variable x

***p** dereferences the pointer p
(returns the value the pointer p points at)

Pointer arithmetic

```
int    *i = new int;
double *d = new double;

cout << i   << "   " << d   << endl;

cout << i+1 << "   " << d+1 << endl;
```

Pointer arithmetic

```
int    *i = new int;
double *d = new double;

cout << i   << "   " << d   << endl; // 0x2537010  0x2537030

cout << i+1 << "   " << d+1 << endl; // 0x2537014  0x2537038
```

the pointer to `int` moved forward by *4 bytes*, and
the pointer to `double` moved forward by *8 bytes*.

Pointer arithmetic

If p is a pointer to a dynamically allocated array:

```
int n = 12;
double *p = new double [n];

cout << p << "   " << p+1 << "   " << p+2 << endl;

delete[] p;
```

Then:

- p is the address of the element [0],
- p+1 is the address of the element [1],
- p+2 is the address of the element [2],

  ...

Analyze the function

Somebody wrote this function, but did not leave any comments

```
void operation(char *s, int n) {

  char *first = &s[0];
  char *last = &s[n-1];

  while(first < last){
    char temp = *first;

    *first = *last;
    ++first;

    *last = temp;
    --last;
  }
}
```

Can we implement something like C++ vector?

We want to be able to:

```
MyVector v; // creates an empty MyVector
v.push_back(1);
v.push_back(4);
v.push_back(12);

for(int i = 0; i < v.size(); i++) {
  cout << v.get(i) << ' ';            //  printing: 1  4  12
}
```

Classes that themselves allocate in the heap

If a class or structure (for example in its constructor) allocates some memory in the heap, this memory should be deallocated when the object gets “destroyed” (e.g.) when the program execution leaves the scope where the varaible is defined

In C++, the user can create a so called destructor, the function that will be called automatically when the object passes out of scope.

Classes that themselves allocate in the heap

"Rule of 3": If a class defines one of the following it should probably explicitly define all three:

- destructor
- copy constructor
- copy assignment operator

In C++11, things got a bit crazier, and there is “Rule of 5”