Problem 4.1

(a)

```
void shift_element(int* pElement) {
    int ivalue;

    for(ivalue = *pElement; pElement != arr && *(pElement - 1) > *pElement; --
pElement) {
        *pElement = *(pElement - 1);
    }

    *pElement = ivalue;
}
```

(b)

```
void insertion_sort(void) {
    int *pstart = arr, *pend = arr + array_length(arr);

    for(pstart += 1; pstart < pend; pstart++) {
        if(*pstart < *(pstart - 1))
            shift_element(pstart);
    }
}
```

Problem 4.2

(a)

```
unsigned int strspn(const char* str, const char* delims) {
    unsigned int count = 0;
    while(*str != '\0') {
        if(strpos(delims, *str) == -1) {
            break;
        }

        ++count;
        ++str;
    }

    return count;
}
```

(b)

```c
unsigned int strcspn(const char* str, const char* delims) {
    unsigned int count = 0;
    while(*str != '\0' && strpos(delims, *str) == -1) {
        ++str;
        ++count;
    }

    return count;
}
```

Problem 4.3

```c
void shift_element_by_gap(unsigned int i, unsigned int gap) {
    int ivalue;

    for(ivalue = arr[i]; i >= gap && arr[i-gap] > ivalue; i -= gap)
        arr[i] = arr[i-gap];
    arr[i] = ivalue;
}

void shell_sort(void) {
    unsigned int gap, i;
    len = array_length(arr);

    for(gap = len / 2; gap > 0; gap /= 2) {
        for(i = gap; i < len; ++i) {
            if(arr[i-gap] > arr[i]) {
                shift_element_by_gap(i, gap);
            }
        }
    }
}
```