1. The code below is missing the implementation of `dupArgv()`, which returns a copy of the `argv` parameter to `main`. **The returned copy is completely independent of `argv`** – changes to `argv` or anything reachable starting from `argv` do not affect the duplicate.

Implement `dupArgv()`. Assume your code will be inserted right just the `#includes`.

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void printArgv(char *argv[]) {
    char **p;
    for (p = argv; *p; p++) { printf("%s
", *p); }
}

int main(int argc, char *argv[]) {
    printArgv(argv);
    char **dup = dupArgv(argv);
    printArgv(dup);
    return 0;
}
```
2. Briefly describe what happens when I try to compile and then (if it successfully compiles) run this code:

```c
#include <stdio.h>
typedef struct name_st {
    char *firstName;
    char *lastName;
} Name;

typedef struct address_st {
    char *streetNumber;
    char *street;
    char *city;
    char *state;
} Address;

int main(int argc, char *argv[]) {
    Name name = {"firstname", "lastname"};
    Address address = {"streetnumber", "street", "city", "state"};
    Name *namePtr = (Name*)&address;
    printf("%s %s\n", namePtr->firstName, namePtr->lastName);
    return 0;
}
```