W3. Memory exercises

Code Fragment	**Space?**	**Where?**	**De-allocated when?**

int main() {
    int i;
}
	sizeof(int)	stack frame for main	when program ends

int fun() {
    float i;
}
 	
int main() {
    fun();
}			

int fun(char i) {
    ...
}
int main() {
    fun('a');
}			

int main() {
    char i[10] = "hello";
}			

int main() {
    char *i;
}			

int main() {
    int *i;
}			

int main() {
    char *i = "hello";
}			

int fun(int *i) {
...
}
 	
int main() {
    int i[5] = {4,5,2,5,1};
    fun(i);
}			

int main() {
    int *i;
    i = malloc(sizeof(int));
}			

void fun(int **i) {
    *i = malloc(sizeof(int)*7);
}
 	
int main() {
    int *i;
    fun(&i);
    free(i);
}			

1. Write a program that declares 3 strings, named *first*, *second*, and *third* in the *main*.
The *first* string is declared as an array and stores the value “Monday” on the stack.
The *second* string is declared as a pointer and points to the string literal “Tuesday”.
The *third* string is declared as a pointer and stores value “Wednesday”, allocated on the heap.

2. Write statements to replace the strings with the abbreviations for the day names. For example, change "Monday" to "Mon". Which string cannot be changed in place? Why not?

3. Draw the memory model for your program.

4. Modify your program so that it declares an array string list of 3 char pointers and set the elements of this array to *first*, *second*, and *third*, respectively. So now you have an array of strings. Where is the memory allocated for this array? Add to your picture above.

5. So far the allocation has happened in the function *main*. What would happen if you changed *main* to be another function *func* and then returned from it? Which parts of your structure would remain allocated? Write a new function *build_month_list* that allocates, initializes and returns an array of 3 strings with the values "January", "February", and "March". All the strings should be mutable.