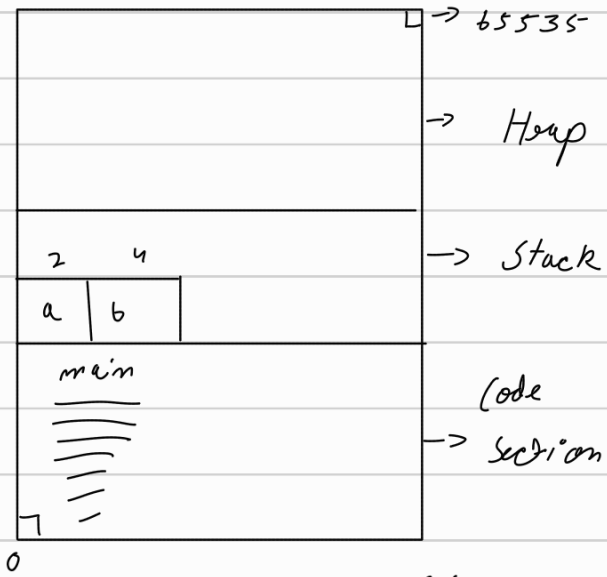
Static vs Dynamic Memory Allocation



Stack frame Activation Record of main func

```
void main () {
    int a;      // 2 bytes
    float b;    // 4 bytes
}
```

☆ Static memory Allocation => Size of the memory which was required by the function was decided at compile time or before run-time.

↳ Occurs in Stack Segment or data Segment

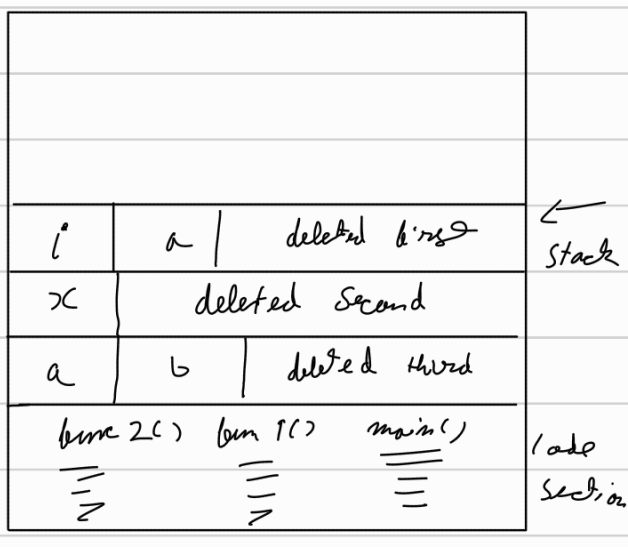
Example

```
void fun2 ( int i ) {
    int a;
}

void func1 ( ) {
    int x;
    fun2(x);
}

int main ( ) {
    int a;
    float b;

    fun1 ( );
}
```



☆ Memory required by a function depends upon no. of variables and its sizes.

☆ Compiler handles this automatically.

Dynamic Memory Allocation (Heap Segment)

Heap => Can be used for both organized & unorganized, but here it is used for unorganized memory.

☆ Heap memory should be treated as a resource.

☆ Program can't directly access heap memory.

☆ To access heap, we need pointer

☆ Pointer takes amount of memory dependent upon size of int.

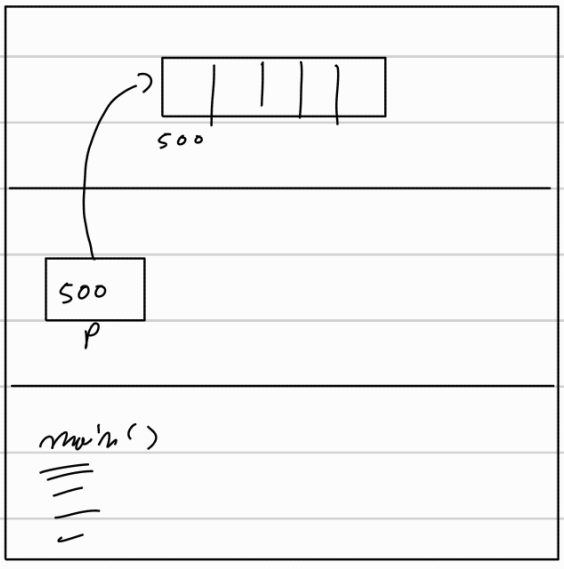
Example

```
Void main() {
int *p;

p = new int[5];
p = (int *) malloc (2*5);

delete p[];

}
```



☆ Not releasing memory may cause memory leaks and over utilization.