1. #include < stdio.h>
static int y = 1; // file scope
int main() {
	static int x; // error -- data type is must.
		// storage class if not given in function, then it is auto.
		// if given then it can be static or register (as given).
	static int z; // function scope
	printf("%d %d", y, z);
	// y = 1, z = 0
	return 0;
}

Answer 1: Garbage value

Answer 2: 0 0

*** Answer 3: 1 0

Answer 4: 1 1

2. What will be the storage class of variable i in the code written below?
#include< stdio.h>
int main() {
	int i = 10; // auto int i = 10;
	printf(%d", i);
	return 0;
}

*** Answer 1: Automatic/local storage class

Answer 2: Extern storage class

Answer 3: Static storage class

Answer 4: Register storage class

3. void fn() {
	static int i=10; // initialized only once -- in data section -- so not destroyed
	printf("%d",++i); // in first call fn() --> ++10 --> 11
}				// in first call fn() --> ++11 --> 12
int main(void) {
	fn();
	fn();
}

Answer 1: 10 10

Answer 2: 11 11

*** Answer 3: 11 12

Answer 4: 12 1

4. Value of static storage variable (similar to global variables)

Answer 1: Changes during different function calls

*** Answer 2: persist between different function calls

Answer 3: increases during different function calls

Answer 4: decreses during different function calls

5.
```
#include <stdio.h>
int main() {
      register static int i = 10; // error: register vars cannot be
global or static.
      i = 11;
      printf("%d\n", i);
}
```

Answer 1: 10

*** Answer 2: Compile time error

Answer 3: Undefined behaviour

Answer 4: 11

7.
```
#include <stdio.h>
int main(void) {
      extern int var=1000; // in declaration, var should not be
initialized --> error
      printf("var = %d",++var);
      return 0;
}
// can be initialized in definition: int var = 1000;
```

Answer 1: var = 1000

Answer 2: var = 0

Answer 3: var = 1001

*** Answer 4: compile time error