1. Put an "X" in front of the names in the following list which are NOT valid C identifiers:

☐	polyArea	☐	_counter
☐	BAD_SS	☐	default_union
☐	16_candles	☐	George&Martha
☐	While	☐	lisp
☐	register	☐	fortran

2. What does the following snippet of code produce when compiled and executed:

```
int a = 13; // a is 13
a = a // + 3;
/* a= 21 */* 2;
int c = /* a + /* 3 + /* 4 */ + 12 /*
- 13 */ ; /*
if (a<30)
*/ { printf("a=%d, c=%d\n",a,c); }
// else { printf("a is too big\n");
```

3. Put an "X" in front of the statements in the following list which are NOT valid C declaration statements:

☐	int x;	☐	char mainCharacter='A';
☐	Float area;	☐	int asciiA='A';
☐	char thisIsARealCharacter;	☐	float double=2.0;
☐	int yearsSinceInit	☐	int minus1=0xFFFFFFFF;
☐	double Double;	☐	int asciiB="B";

4. What does the following snippet of code produce when compiled and executed:

```
int a = 13;
int b = ++a;
if (++b < ++a) { printf("Condition 1\n"); }
else {
   if (--b > a--) { printf("Condition 2\n"); }
   else {
      if (++b < a++) { printf("Condition 3\n"); }
      else { printf("Condition 4\n"); }
   }
}
printf("a=%d, b=%d\n",a,b);
```

5. What does the following snippet of code produce when compiled and executed:

```
int a = 13; int b=4; int c=7;
a = (a + b * c – b);
b = a % 11;
c = (7 + b) * (a – c);
printf("a=%d, b=%d c=%d\n",a,b,c);
```