Variable Scope and Storage Specifier_assignment

1. Refer the code snippet below and answer the queries

```
int val;
extern void display();
static int function()
{
val++;
int x = 10;
int i = 0;
static int j = 20;
for (; i < 3; i++)
{
int x = 20;
printf("\n %d", x+i);
x+=3;
j++;
display();
}
return val;
}
int main(int argc, char *argv[])
{
val= 0;
function();
return 0;
}
```

Output:

A. What is the change required if val declaration line below is to be moved to an other file?

If val is declared in a separate file (for example, a different .c file), you need to declare it as extern in the file where function() is defined (or wherever it needs to be accessed). This tells the compiler that the variable val is declared in another file and it will be resolved at link time.

b. What is the value of x after for loop execution?

Inside the for loop, the variable x is shadowed by a local variable x = 20; each time the loop runs. This means the x defined inside the loop doesn't affect the x defined outside the loop.

c. What does the keyword static in following lines mean?

```
static int function()
 static int j = 20 ;
```

static int function**()**:

- The static keyword here means that the function function() is restricted to the file in which it is defined (file scope). It cannot be called from other files or modules. This is useful when you want to limit the scope of the function to just that particular file (i.e., it is not visible outside the file).

static int j = 20;:

- The static keyword applied to the variable j means that the variable retains its value between function calls. It is initialized only once (to 20 in this case) and its value persists across function calls. This is in contrast to a regular local variable, which would be reinitialized each time the function is called.

d. What is the value of j after for loop execution?

-> The final value of j after the loop finishes is **23**.

e. Identify the variables which would be in the stack of function()

-> The stack typically stores local variables and function call information. The following variables are local to the function() and would be placed on the stack:

- x (local to the function): Inside the loop, there are two variables named x:
    - x = 10; (initialized at the start of the function, but doesn't change in the loop)
    - x = 20; (this shadows the previous x inside the loop)
- i (loop index): A local variable used to control the loop.
- j (static variable, but it is not stored on the stack; it's stored in a special data section for static variables).
- val: This is declared outside the function but is a global variable, so it's not part of the stack in this function.
- The return address and other necessary function call information will also be stored in the stack.

f. What does extern in the following line mean?

extern means that the display() function is defined in another file.