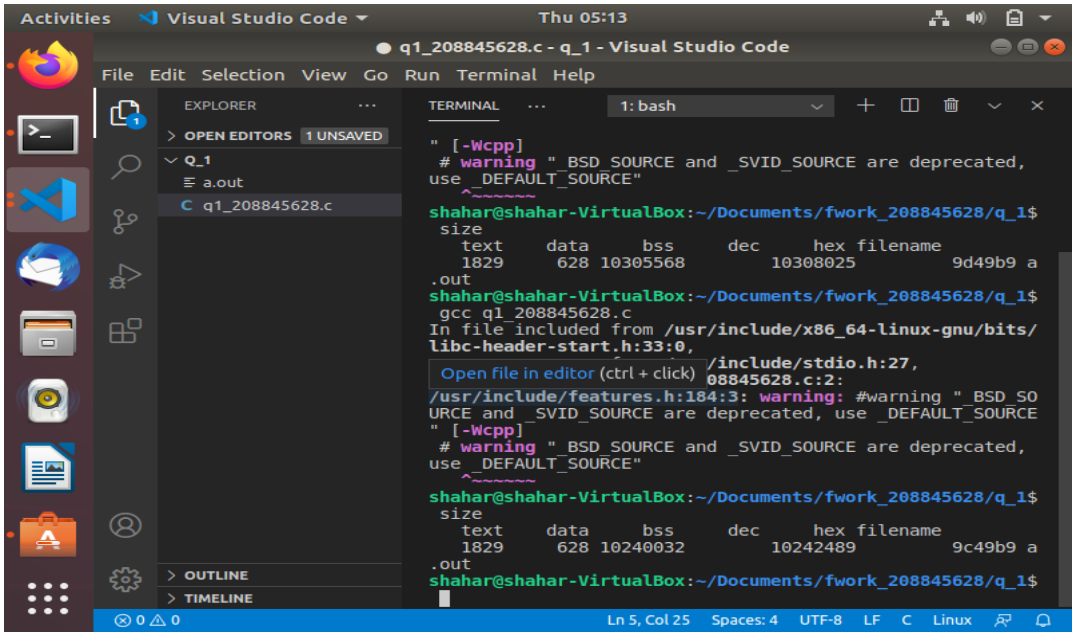
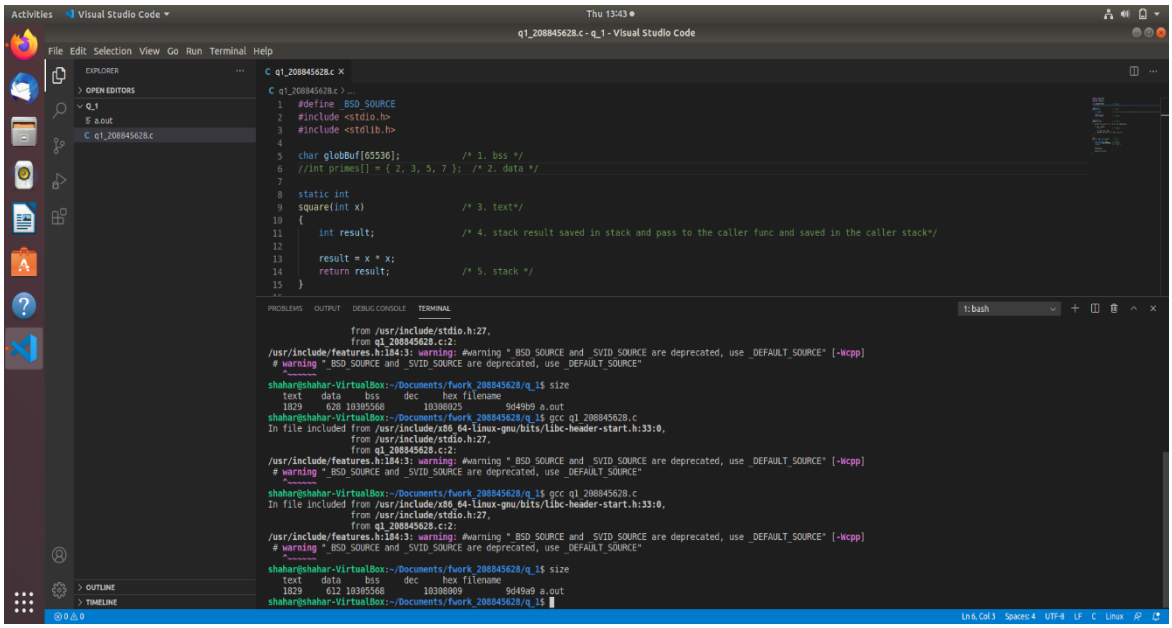
Question 1:

```
/*1. bss */ 5 char globBuf[65536];
```



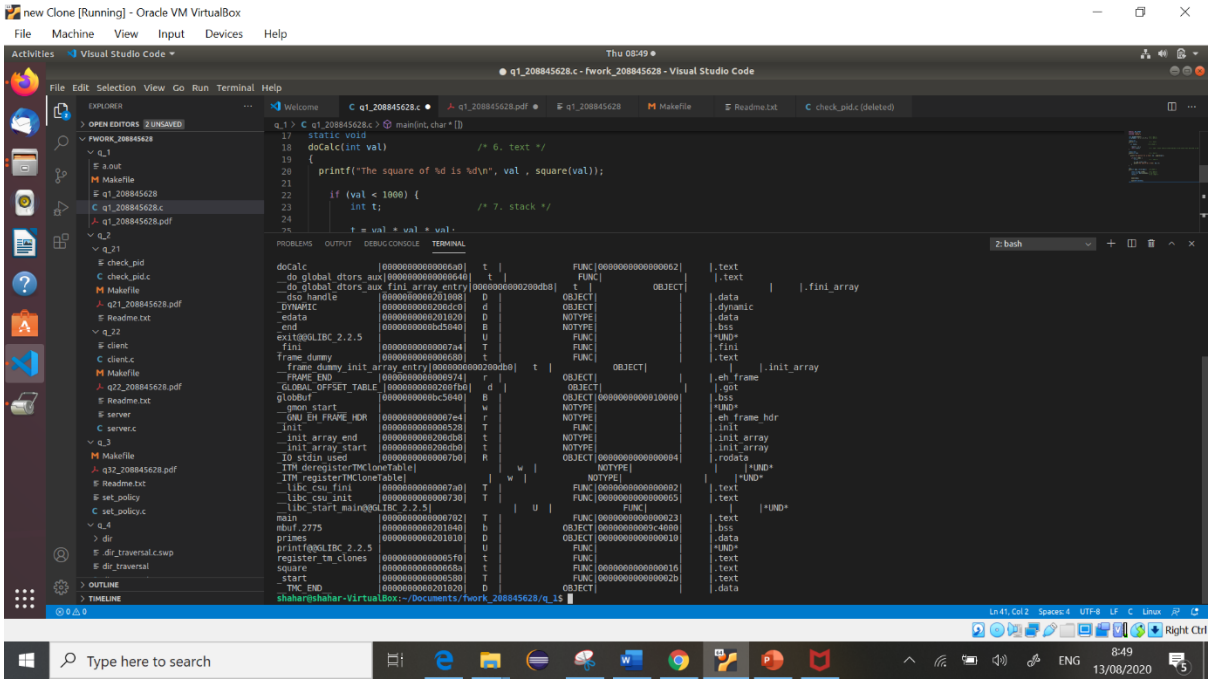
I used size command before and after putting this line in comment and the bss size decreased.

```
/*2. data */ 6 int primes[] = { 2, 3, 5, 7 };
```



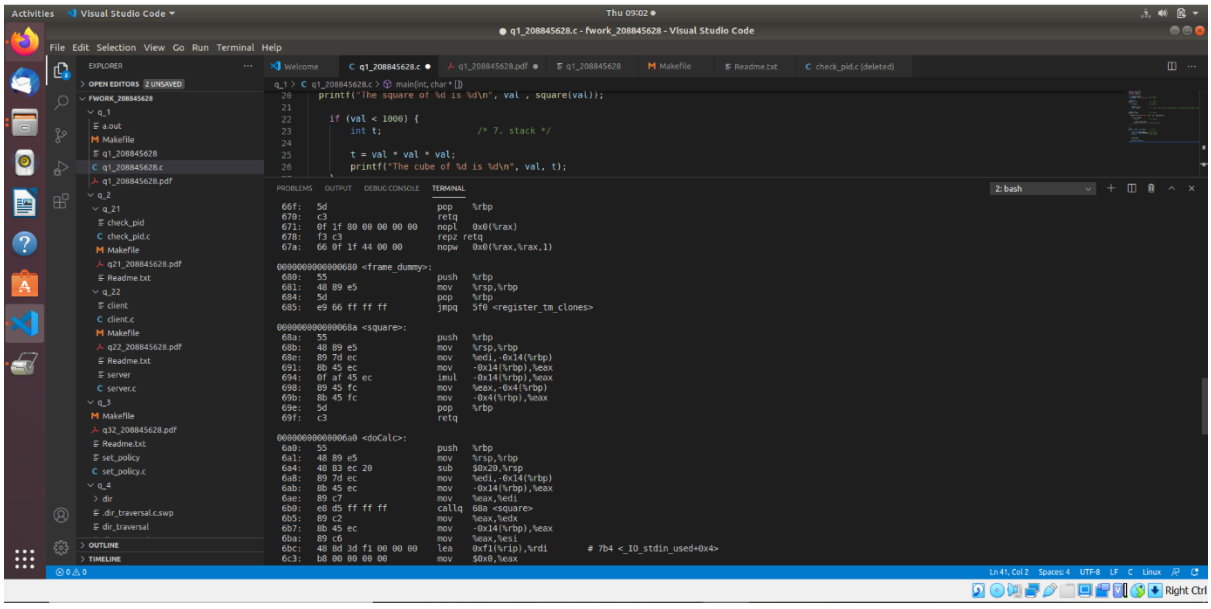
I used size command before and after putting this line in comment and the data size decreased.

```
/*3. text */ 9 static int square(int x)
```



I used the nm command to show where every line is allocated.(the text keep the “instructions” that the program need)

```
/*4. stack */ 11. int result;
```



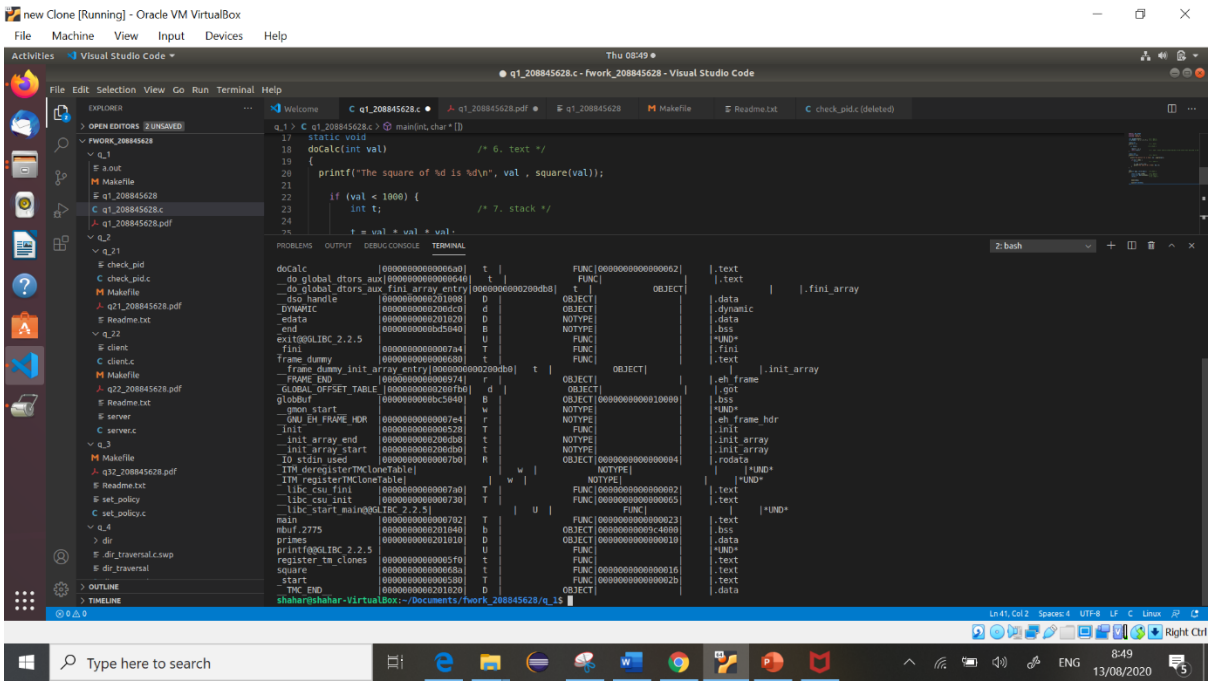
I used the objdump command to so show that in function square we use stack functions like push , pop and mov.

Therefor ,local variables inside a scope (not global or static) is kept in stack .

/*5. **stack** */ 14 **return** result;

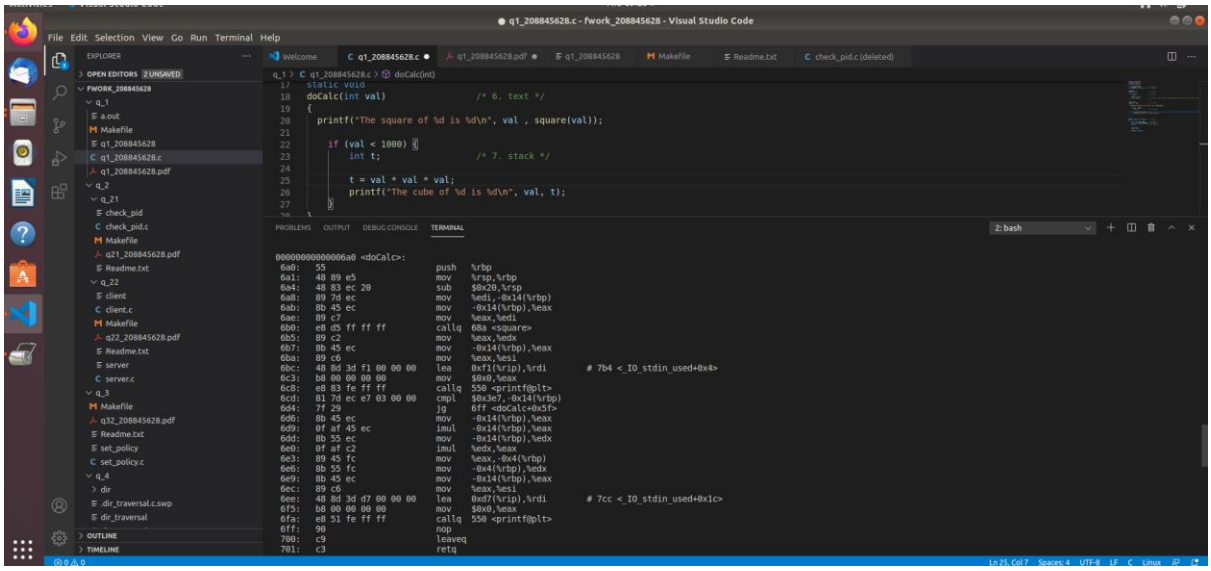
The memory saves the variable result and move it to the caller function's stack to save it there.(same pic as in 4)

```
/*6. text */ 18 static void doCalc(int val)
```



I used the nm command to show where every line is allocated.

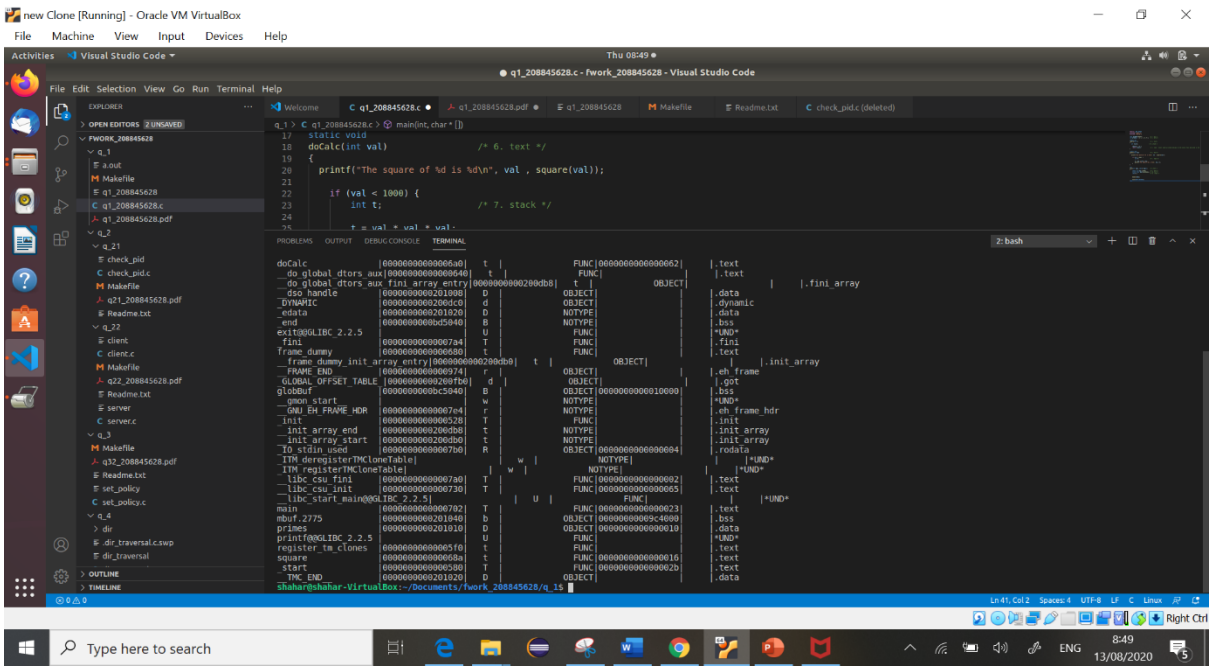
```
/*7. stack */ 23 int t;
```



I used the objdump command to so show that in function doCalc we use stack functions like push , pop and mov.

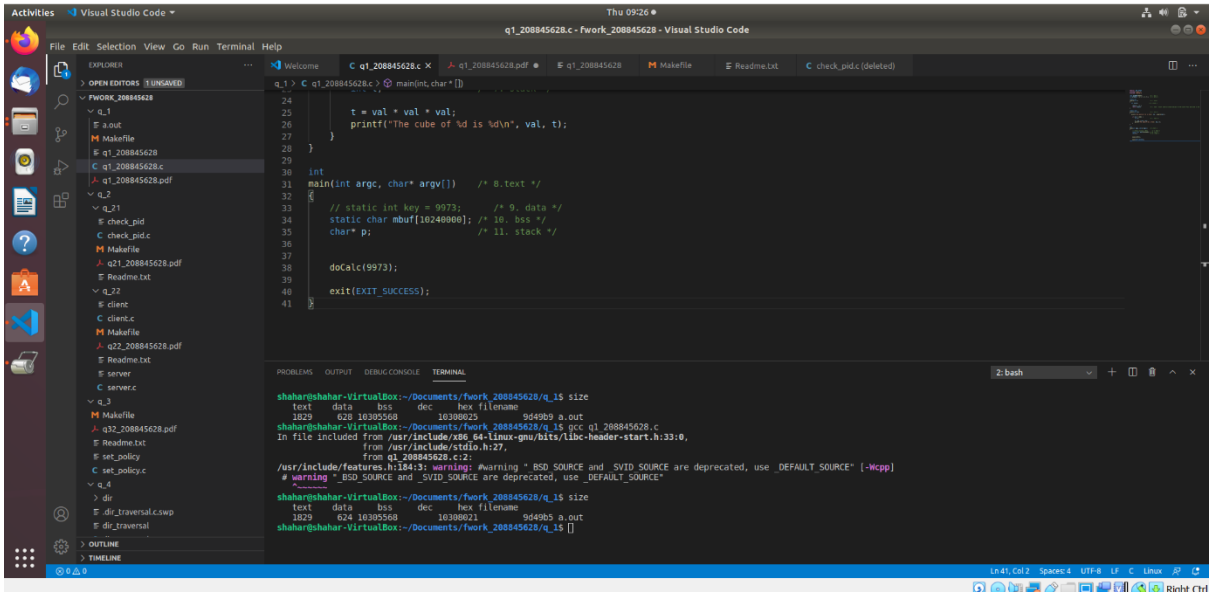
Therefor ,local variables inside a scope (not global or static) is kept in stack .

```
/*8.text */  31 int main(int argc, char* argv[])
```



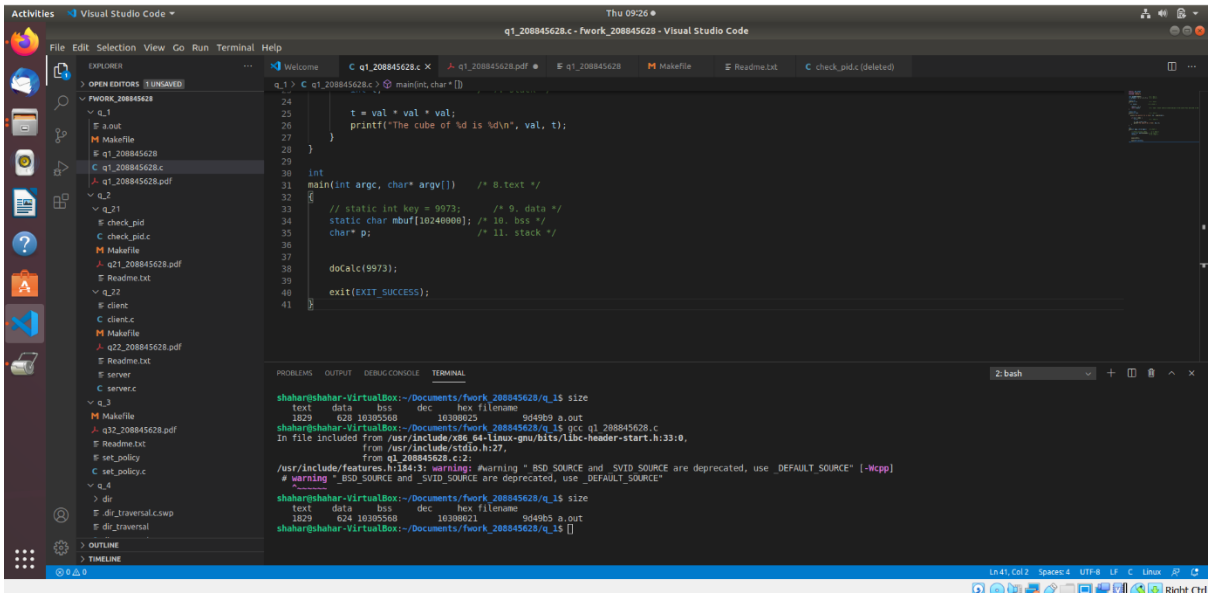
I used the nm command to show where every line is allocated.

```
/*9.data */  33 static int key = 9973;
```



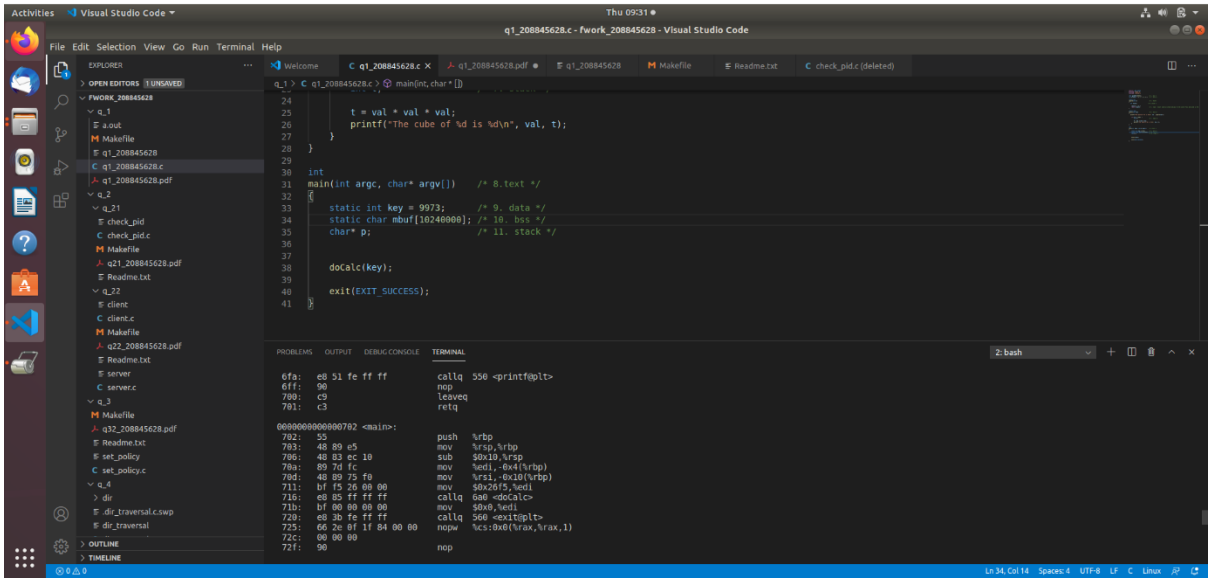
I used size command before and after putting this line in comment and the data size decreased.

```
/*10.bss */ 34 static char mbuf[10240000];
```



I used size command before and after putting this line in comment and the bss size decreased.

```
/*11. stack */  35 char* p;
```



I used the objdump command to so show that in function main we use stack functions like push , pop and mov.