C_89 ⇒ Functions in C → Part 6

Functions without Argument with return type

Syntax :-

Return type functionsname (void) ;

↓

int / char / float . . . .

Example:

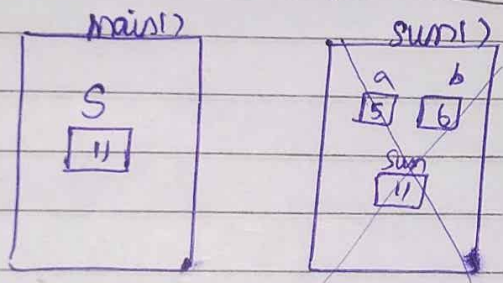
```
int sum(void);
void main()
{
    int s;
    s = sum();
    // printf("sum = %d", sum);
}
int sum()
{
    int a, b, sum=0;
    printf("Enter 2 numbers:");
    scanf("%d %d", &a, &b);
    sum = a+b;
    return sum;
    printf("Inside sum:");
}
```

s → (correct).

This line (error) bcoz sum is defined only within sum()

Here we return our sum() function value.

This line is not executed because the Control will go to return function.




```
// return a+b;   → return 11
// return sum, a;  → return a value
```

because comma operator; so operator procedure and ignores first value i.e sum.

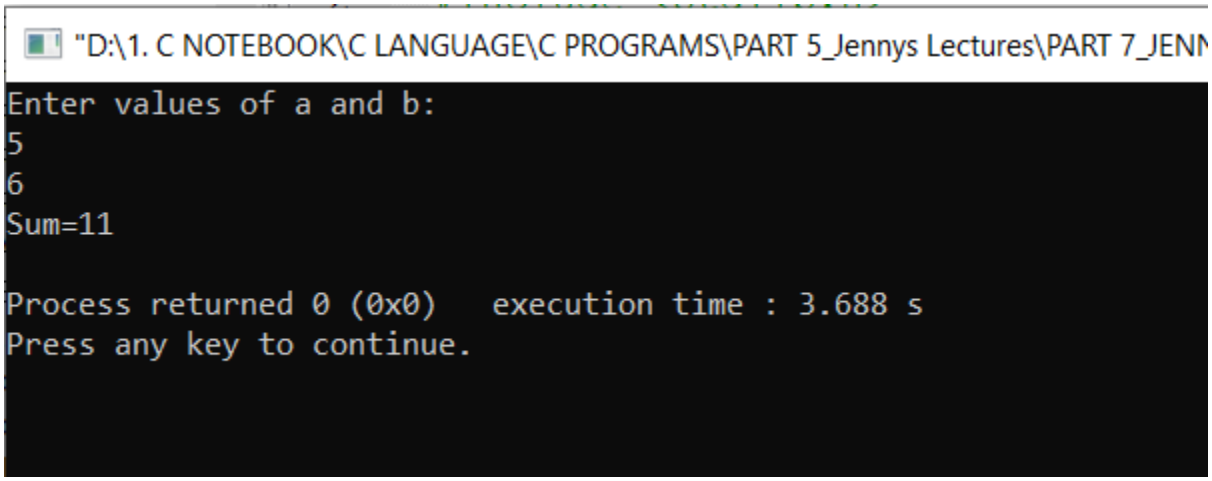
```
// return sum, a, b;  → return b value.

// return 'a', 'b', '1';
```

PROBLEM 1:

```c
#include <stdio.h>
#include <stdlib.h>
/** 1 - No Argument with Return Type **/

int sum(void);
int main()
{
 int s;
 s=sum();
 printf("Sum=%d\n",s);
}
int sum()
{
 int a,b,sum=0;
 printf("Enter values of a and b:\n");
 scanf("%d %d",&a,&b);
 sum=a+b;
 return sum;
}
```



PROBLEM 2:

```c
#include <stdio.h>
#include <stdlib.h>
/** 2 - No Argument with Return Type **/
int sum(void);
int main()
{
 int s;
 s=sum();
 printf("Sum=%d\n",addition);
 //addition variable is local to sum() function only
}
int sum()
{
 int a,b,addition=0;
 printf("Enter values of a and b:\n");
 scanf("%d %d",&a,&b);
 addition=a+b;
 return addition;
}
```

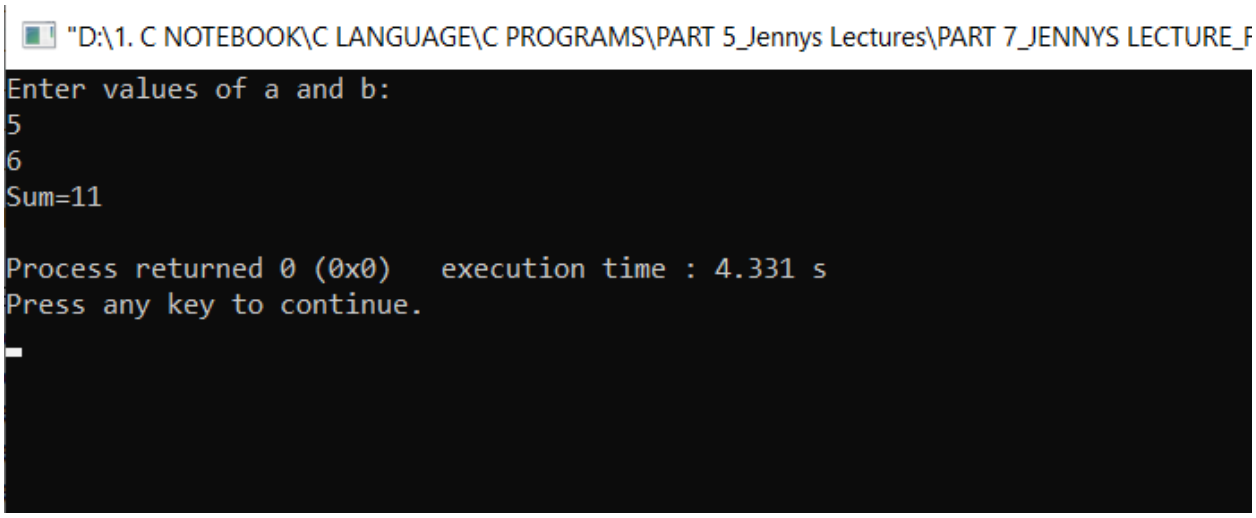
Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages | CppCheck/Vera++ | CppCheck/Vera++

File	Line	Message
D:\1. C NO...	10	error: 'addition' undeclared (first use in this function)
D:\1. C NO...	10	note: each undeclared identifier is reported only once for each fu...

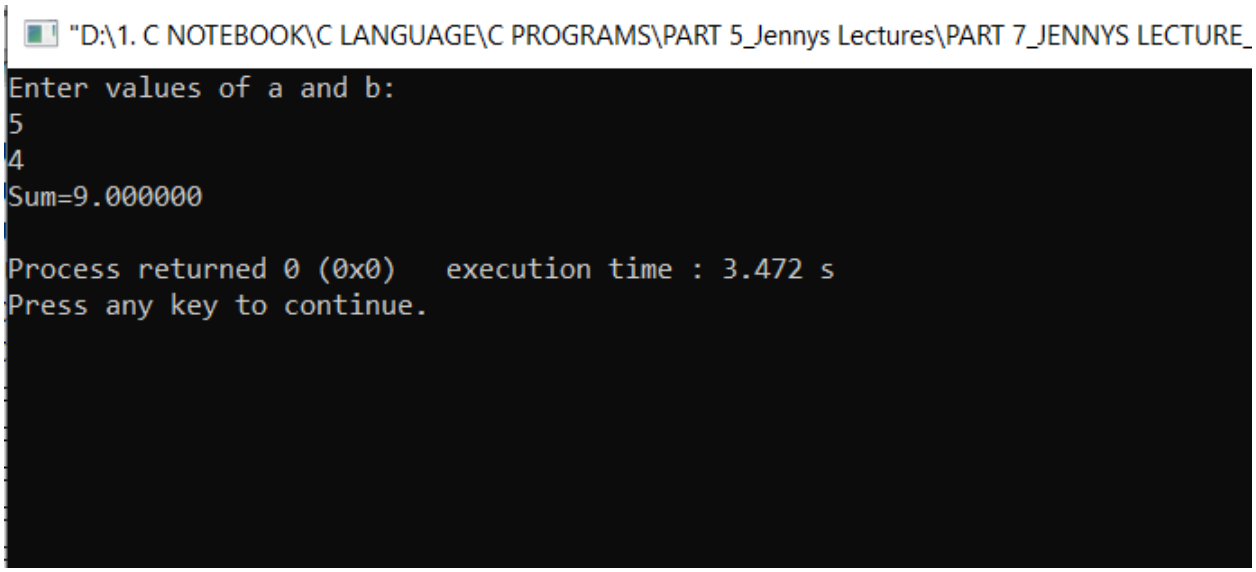
PROBLEM 3:

```c
#include <stdio.h>
#include <stdlib.h>
/** 3 - No Argument with Return Type **/
int sum(void);
int main()
{
 int s;
 s=sum();
 printf("Sum=%d\n",s);
}
int sum()
{
 int a,b;
 printf("Enter values of a and b:\n");
 scanf("%d %d",&a,&b);
 return a+b;
}
```



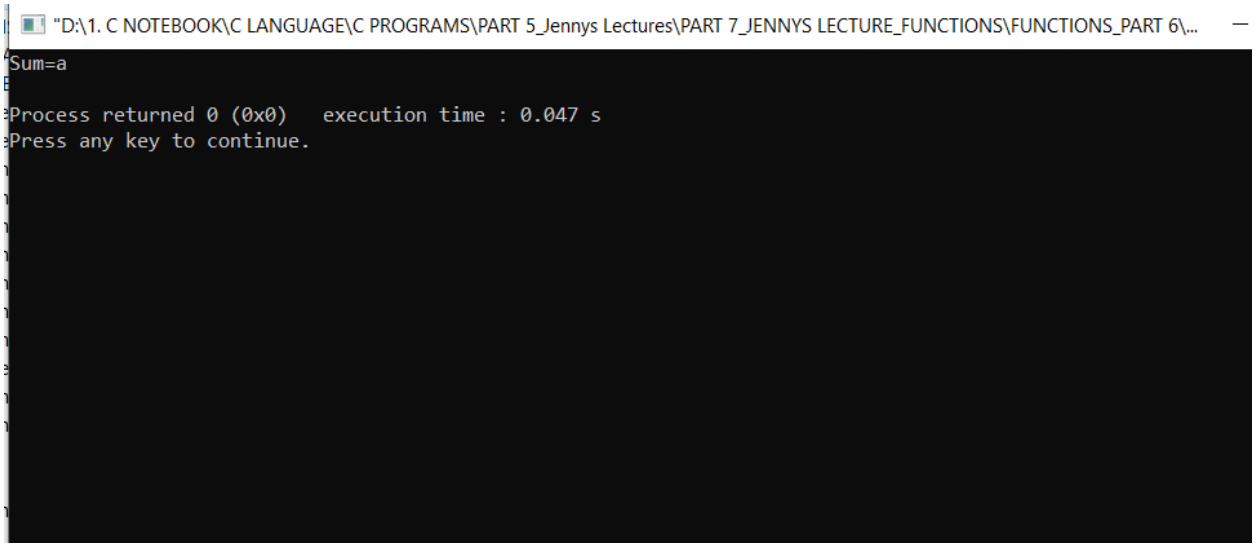
PROBLEM 4:

```c
#include <stdio.h>
#include <stdlib.h>
/** 4 - No Argument with Return Type **/
float sum(void);
int main()
{
 float s;
 s=sum();
 printf("Sum=%f\n",s);
}
float sum()
{
 int a,b;
 printf("Enter values of a and b:\n");
 scanf("%d %d",&a,&b);
 return a+b;
}
```



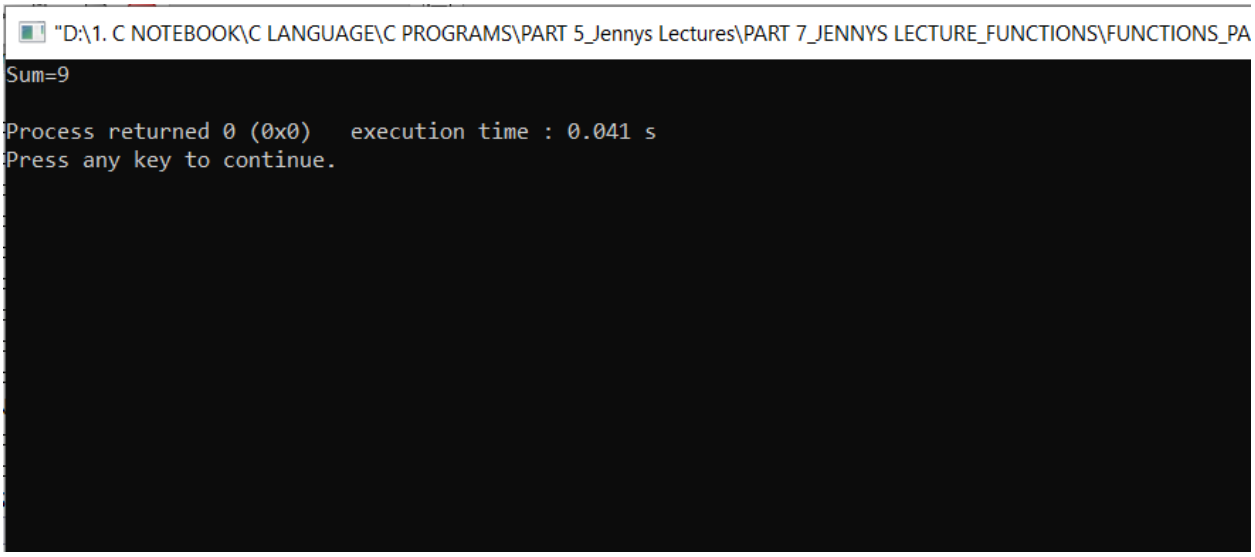
PROBLEM 5:

```c
#include <stdio.h>
#include <stdlib.h>
/** 5 - No Argument with Return Type **/
char sum(void);
int main()
{
 char s;
 s=sum();
 printf("Sum=%c\n",s);
}
char sum()
{
 int a=4,b=5;
 return 'a';
}
```



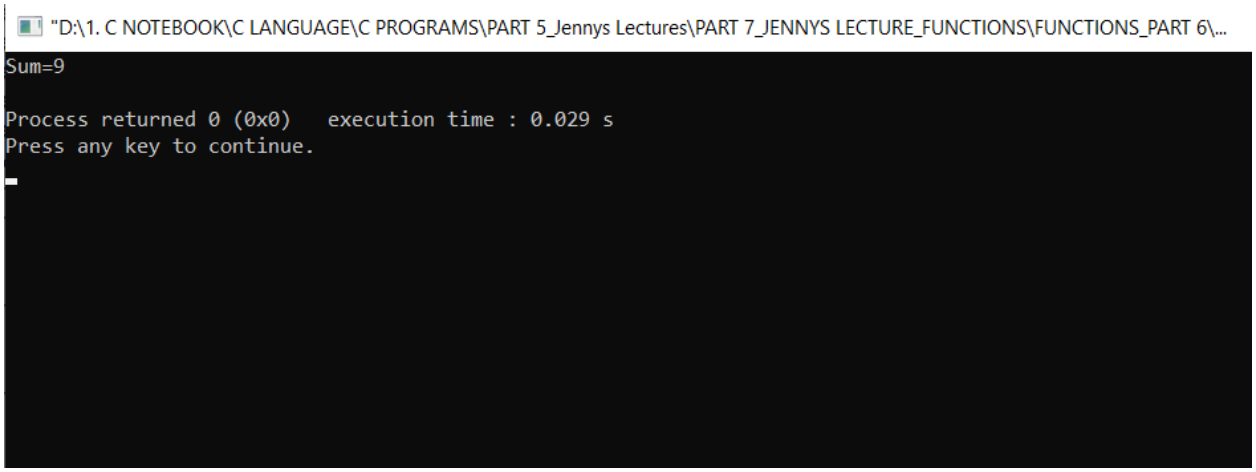
PROBLEM 6:

```c
#include <stdio.h>
#include <stdlib.h>
/** 6 - No Argument with Return Type **/
int sum(void);
int main()
{
 int s;
 s=sum();
 printf("Sum=%d\n",s);
}
int sum()
{
 int a=4,b=5;
 return a+b; //from this statement the function ends
 return 'a'; //this statement is not evaluated
}
```



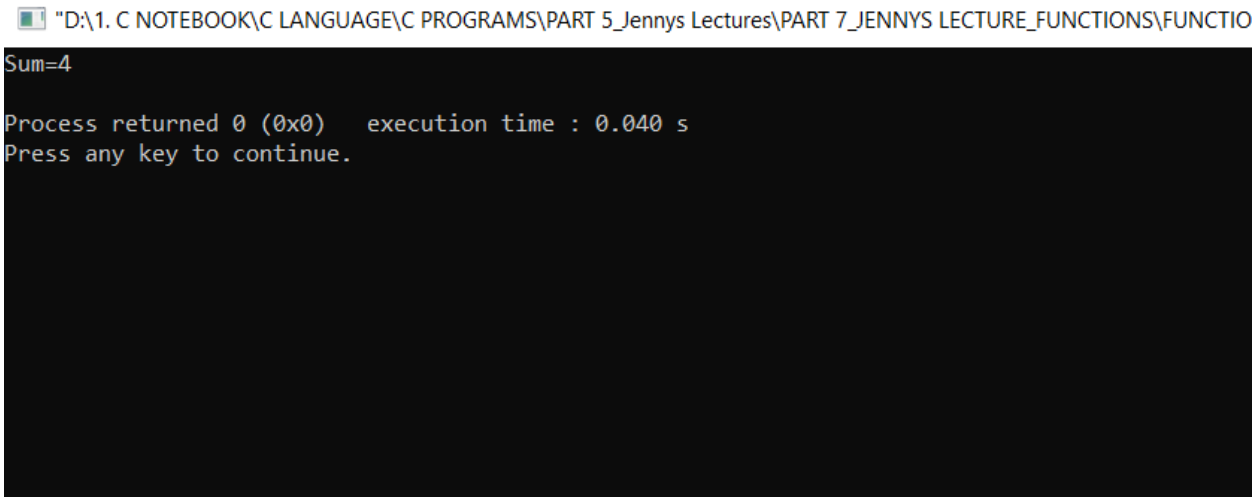
PROBLEM 7:

```c
#include <stdio.h>
#include <stdlib.h>
/** 7 - No Argument with Return Type **/
int sum(void);
int main()
{
 int s;
 s=sum();
 printf("Sum=%d\n",s);
}
int sum()
{
 int a=4,b=5,sum=0;
 sum=a+b;
 return a,sum;
}
```



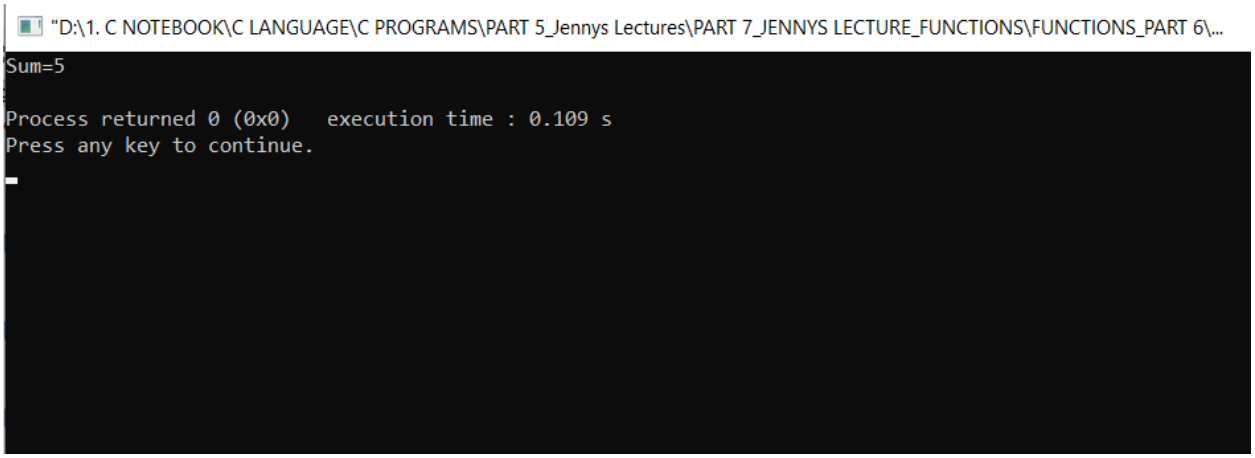
PROBLEM 8:

```c
#include <stdio.h>
#include <stdlib.h>
/** 8 - No Argument with Return Type **/
int sum(void);
int main()
{
 int s;
 s=sum();
 printf("Sum=%d\n",s);
}
int sum()
{
 int a=4,b=5,sum=0;
 sum=a+b;
 return sum,a; //comma operator precedence from right to left
}
```



PROBLEM 9:

```c
#include <stdio.h>
#include <stdlib.h>
/** 9 - No Argument with Return Type **/
int sum(void);
int main()
{
 int s;
 s=sum();
 printf("Sum=%d\n",s);
}
int sum()
{
 int a=4,b=5,sum=0;
 sum=a+b;
 return sum,a,b; //comma operator precedence from right to left
}
```



PROBLEM 10:

```c
#include <stdio.h>
#include <stdlib.h>
/** 10 - No Argument with Return Type **/
char sum(void);
int main()
{
 char s;
 s=sum();
 printf("Sum=%c\n",s);
}
char sum()
{
 int a=4,b=5,sum=0;
 sum=a+b;
 return 'a','0','1'; //comma operator precedence from right to left
}
```