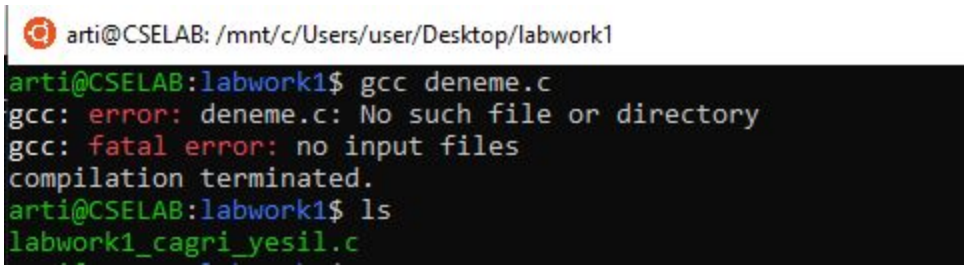
TABLE OF CONTENTS

COMMON ERRORS IN C LANGUAGE

The common errors and solutions are explained in this document. If you have an error in your code, please first check this document and try to solve your problem yourself. If the error you have encountered is not in this list, then ask us.

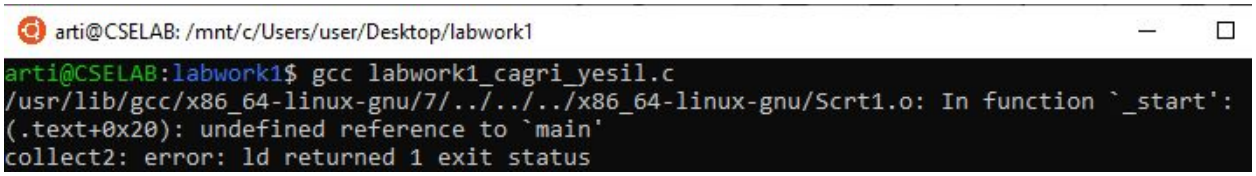
1) No such file or directory

If you have this error. You try to compile a file that does not exist. If you believe the file exists, then maybe you have misspelled your file name or you are in the wrong directory. So you can the file in the current directory by writing the “ls” command as it is seen in the image below.



2) Undefined reference to ‘main’

You probably forget to write the main function or misspelled it. Check your main function.



3) Expected ‘;’ before

It says that in file error1.c, there must a “;” symbol but you did not put it in line 8 or before line 8.

WRONG	**CORRECT**

#include<stdio.h>
void main()
{
int x = 10;
int y = 15;
 	
printf("%d", x) // semicolon missed
}
	#include<stdio.h>
	void main()
	{
	int x = 10;
	int y = 15;
	
	printf("%d", x);
	}

4) ‘x’ undeclared (first use in this function)

It says that in file error1.c, you used a variable named “x” in line 6, but you did not declare it.

WRONG	**CORRECT**

#include<stdio.h>
void main()
{
int y = 15;
printf("%d", x);
}
	#include<stdio.h>
	void main()
	{
	int x = 15;
	printf("%d", x);
	}

5) Implicit declaration of function ‘mypow’

- It says that in file error1.c, you try to access a function named “‘mypow’”, but there is not such a function
- The reason can be different reasons for this problem.
    - You may used the name of the function incorrectly
    - You may not declare the function
    - You may declare the function before you call it

WRONG	**CORRECT**

#include<stdio.h>
 	
int myPow(int z)
{
return z*z;
}
void main()
{
int x = 15,result;
result = mypow(x);
printf("%d", result);
}
	#include<stdio.h>
	
	int myPow(int z)
	{
	return z*z;
	}
	void main()
	{
	int x = 15,result;
	result = myPow(x);
	printf("%d", result);
	}

#include<stdio.h>
 	
void main()
{
int x = 15,result;
result = myPow(x);
printf("%d", result);
}
 	
int myPow(int z)
{
return z*z;
}
	#include<stdio.h>
	
	int myPow(int z)
	{
	return z*z;
	}
	
	void main()
	{
	int x = 15,result;
	result = mypow(x);
	printf("%d", result);
	}

6) Too few arguments to function ‘MyFunction’

MyFunction requires more arguments than you supplied.
In the example below, you supplied no (0) arguments to MyFunction whereas it required an (1) argument.

WRONG	**CORRECT**

#include <stdio.h>
 	
void MyFunction(int number)
{
  printf("CSE %d\n", number);
}
 	
int main()
{
  MyFunction();
}
	#include <stdio.h>
	
	void MyFunction(int number)
	{
	  printf("CSE %d\n", number);
	}
	
	int main()
	{
	  int i = 114;
	  MyFunction(i);
	}

7) Too many arguments to function ‘MyFunction’

MyFunction requires less arguments than you supplied.
In the example below, you supplied 2 arguments to MyFunction whereas it required only 1 argument.

WRONG	**CORRECT**

#include <stdio.h>
 	
void MyFunction(int number)
{
printf("CSE %d\n", number);
}
 	
int main()
{
int i = 114;
MyFunction(i, 3.14);
}
	#include <stdio.h>
	
	void MyFunction(int number)
	{
	printf("CSE %d\n", number);
	}
	
	int main()
	{
	int i = 114;
	MyFunction(i);
	}

8) ‘Return’ with a value, in function returning void

MyFunction has been declared as returning void but you tried to return a value inside.

WRONG	**CORRECT**

#include <stdio.h>
 	
void MyFunction(int number)
{
printf("CSE %d\n", number);
return number*2;
}
 	
void main()
{
int i = 114;
MyFunction(i);
}
	#include <stdio.h>
	
	int MyFunction(int number)
	{
	printf("CSE %d\n", number);
	return number*2;
	}
	
	void main()
	{
	int i = 114;
	MyFunction(i);
	}

9) Format expects argument of type ‘*’

When you are taking an input from the user and passing it to a variable by using scanf function, you need to use ‘&’ sign.

WRONG	**CORRECT**

#include <stdio.h>
 	
void main()
{
int i;
scanf("%d", i);
}
	#include <stdio.h>
	
	void main()
	{
	int i;
	scanf("%d", &i);
	}

10) Format ‘%d’ expects a matching ‘int’/’int *’ argument

In the scanf and printf functions, lack of parameter number leads to this error.

WRONG	**CORRECT**

#include <stdio.h>
 	
void main()
{
int i, j;
scanf("%d%d", &i);
}
	#include <stdio.h>
	
	void main()
	{
	int i, j;
	scanf("%d%d", &i, &j);
	}

WRONG	CORRECT

#include <stdio.h>
 	
void main()
{
int i=114, j=1;
printf("CSE %d Section %d", i);
}
	#include <stdio.h>
	
	void main()
	{
	int i=114, j=1;
	printf("CSE %d Section %d", i, j);
	}

11) Too many arguments for format

This error occurs if there are too many parameters in the scanf and printf functions.

WRONG	CORRECT

#include <stdio.h>
 	
void main()
{
int i, j;
scanf("%d", &i, &j);
}
	#include <stdio.h>
	
	void main()
	{
	int i, j;
	scanf("%d%d", &i, &j);
	}

WRONG	CORRECT

#include <stdio.h>
 	
void main()
{
int i=114, j=1;
printf("CSE %d", i, j);
}
	#include <stdio.h>
	
	void main()
	{
	int i=114, j=1;
	printf("CSE %d Section %d", i, j);
	}

12) Expected ‘;’, ‘,’ or ‘)’ before ‘{’ token

This error occurs if there are missing brackets, parentheses or semicolons in your code.

WRONG	CORRECT

#include <stdio.h>
int sum(int x, int y{
  return x + y;
}
int main()
{
  int x = 5;
  int y = 10;
  int result = sum(x,y);
  printf("%d", result);
  return 0;
}
	#include <stdio.h>
	int sum(int x, int y){
	  return x + y;
	}
	int main()
	{
	  int x = 5;
	  int y = 10;
	  int result = sum(x,y);
	  printf("%d", result);
	  return 0;
	}

13) Expected ‘char *’ but argument is of type ‘int’

This error occurs where the datatype of the function argument is **not** the same as that of the argument passed during the call.

WRONG	CORRECT

#include <stdio.h>
void myfunc (char *x){
  //...
}
 	
int main() {
  int a = 4;
  //...
  myfunc(a);
}
	#include <stdio.h>
	void myfunc (int x){
	  //...
	}
	
	int main() {
	  int a = 4;
	  //...
	  myfunc(a);
	}

14) Redefinition of 'main'

Since the main function is the starting point of a C program, there cannot be more than 1 *main*.

WRONG	**CORRECT**

#include <stdio.h>
 	
int main()
{
  printf(“1st part of the labwork.\n”);
  int x;
}
 	
int main()
{
  printf(“2nd part of the labwork.\n”);
  int y;
}
	int main()
	{
	  printf(“1st part of the labwork.\n”);
	  int x;
	  printf(“2nd part of the labwork.\n”);
	  int y;
	}

15) Redeclaration of 'x' with no linkage

Variables must only be declared once. If you want to change the value of a previously declared variable, you can do this by using only its name.

WRONG	**CORRECT**

int main()
{
  int x = 5;
  int x = 10;
}
	int main()
	{
	  int x = 5;
	  x = 10;
	}

16) Expected expression before 'int'

When calling a function, you must send it a previously declared variable by only its name.

WRONG	**CORRECT**

int Function(int x)
{
  return x + 5;
}
 	
int main()
{
  Function(**int** x);
}
	int Function(int x)
	{
	  return x + 5;
	}
	
	int main()
	{
	  int x;
	  Function(**x**);
	}