

Assignment 2: Binary numbers, data types and C Programs

Lavanya Sajwan

300381661

Core 1

$$1 = 2^3$$

$$0 = 2^2$$

$$2 = 2^1$$

$$1 = 2^0$$

11

Core 2

$$2^3 \quad 2^2 \quad 2^1 \quad 2^0$$

1101

Core 3

0101

$$0 = 2^3$$

$$1 = 2^2$$

$$0 = 2^1$$

$$1 = 2^0$$

5

Core 4

0110

$$0 = 2^3$$

$$1 = 2^2$$

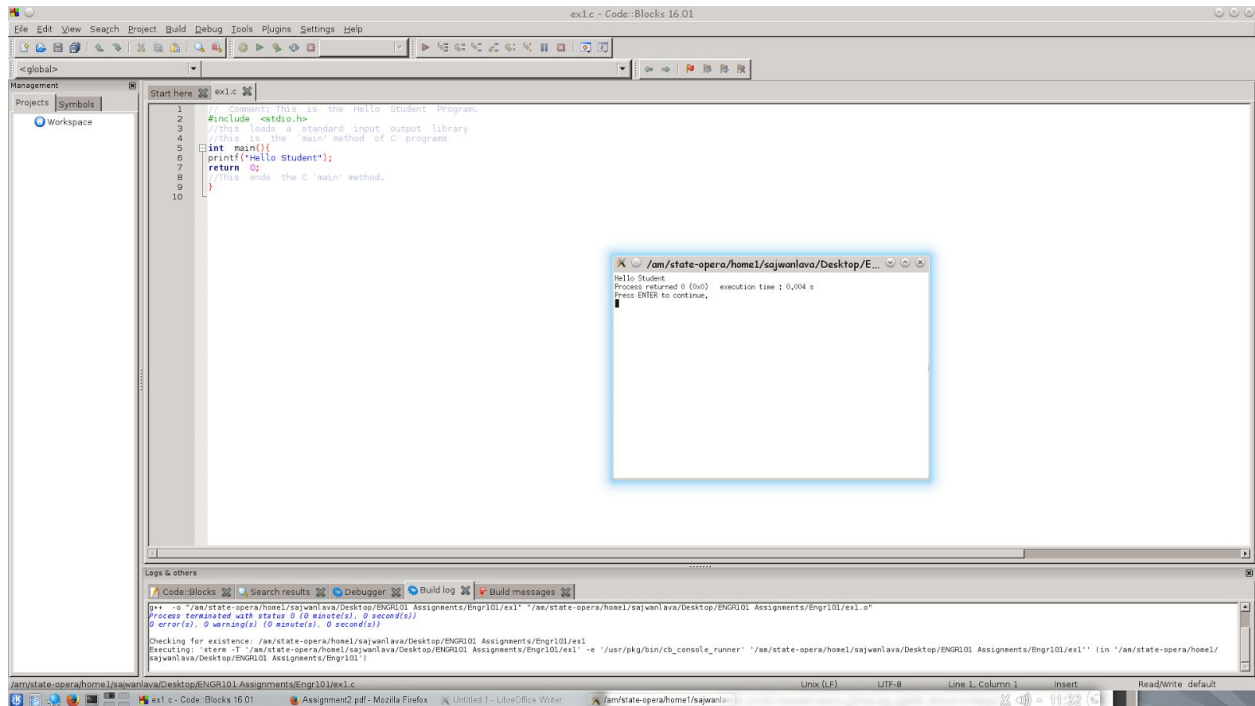
$$1 = 2^1$$

$$0 = 2^0$$

6

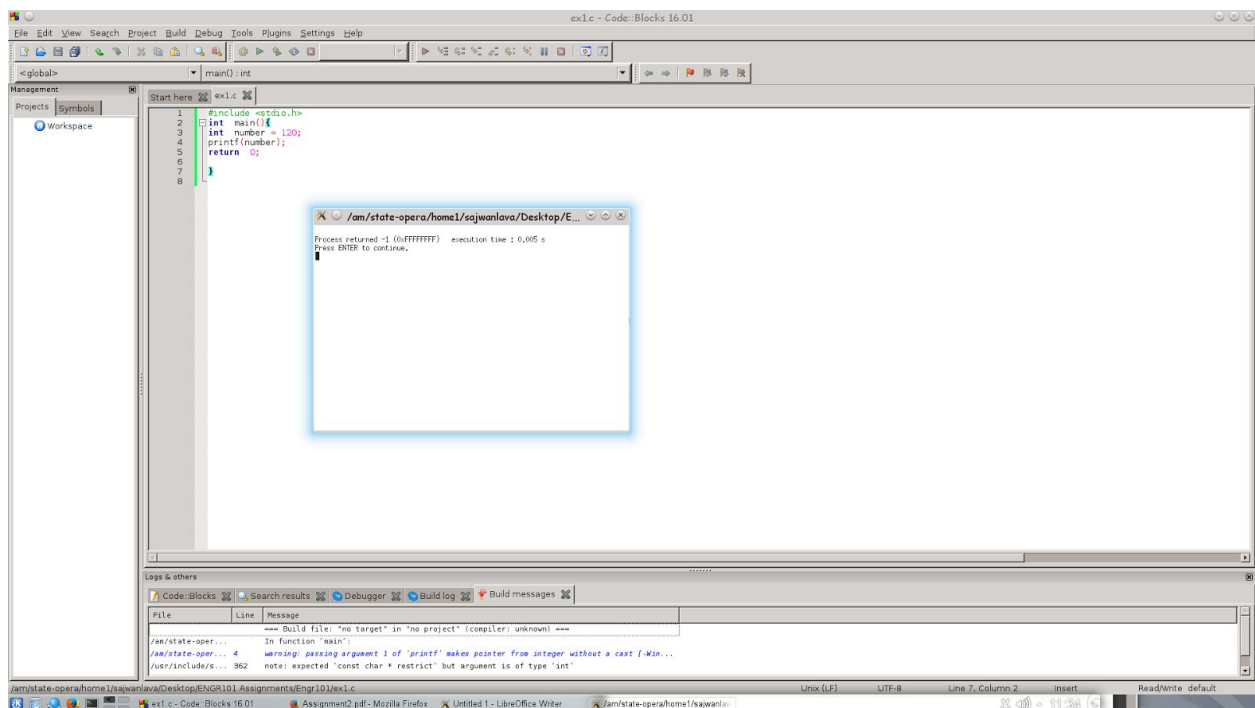
Core 5

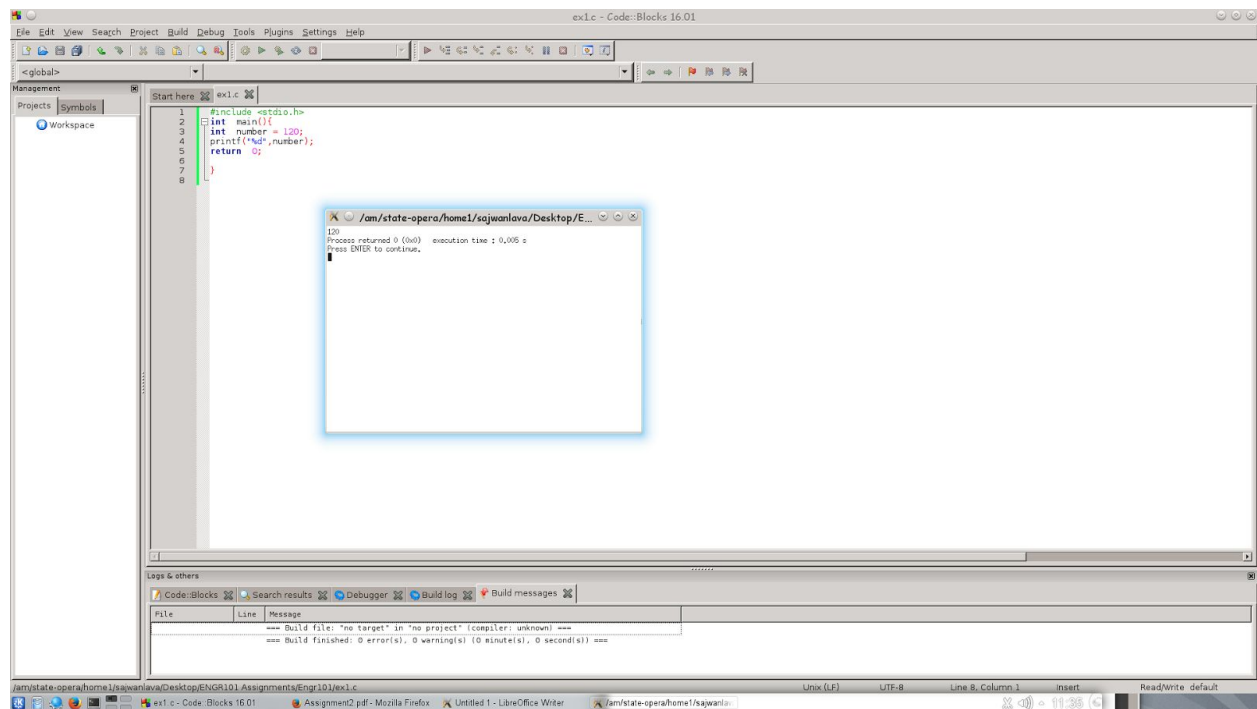
Compiler translates the code we have typed into another type of code that the computer can read.



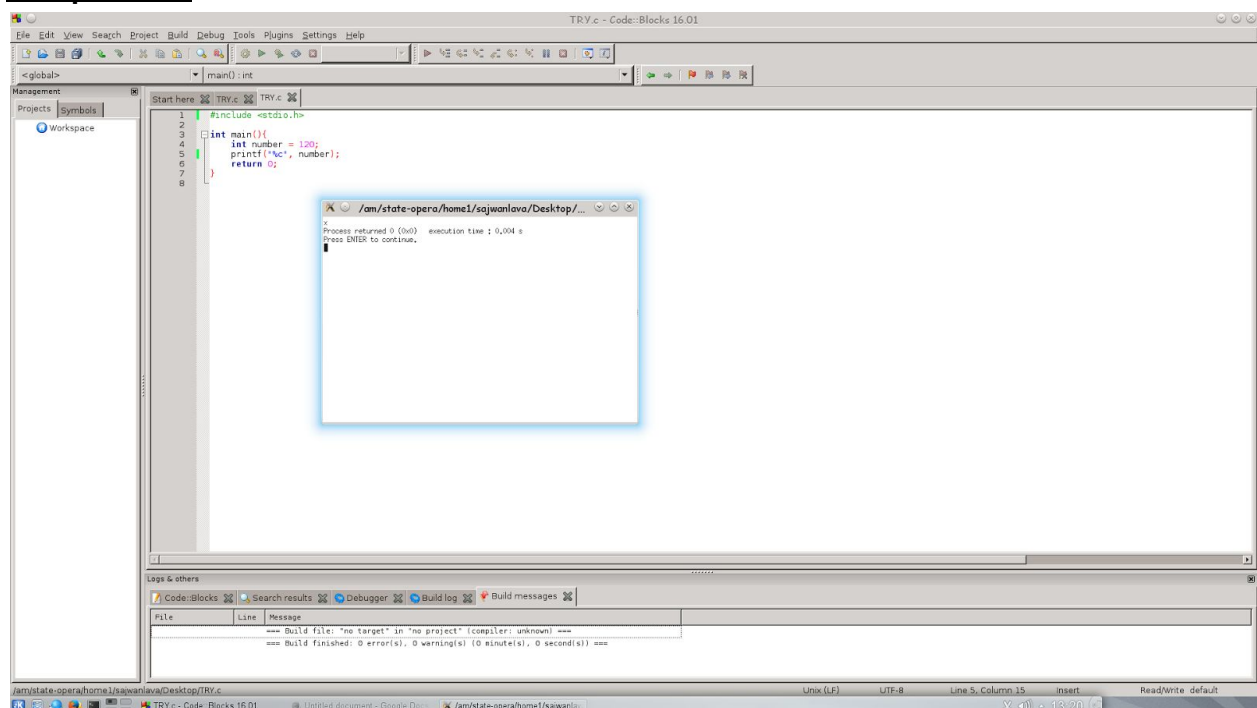
Core 7

The output of the code would be the number 120





Completion 1



The “%d” reads for decimals in the code and in this case we have defined 120. However when we replace the “%d” with the “%c”, the “%c” interprets the number as a character and in this case it comes out as a x.

Core 8

Variable	Size
Integer	4
Character	1
Long	8
Double	8

Challenge 1

Memory in computers can be retrieved and stored. Memory addresses are certain locations where the specific pieces of data are stored.

Core 9

Challenge 2

Core 10

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0
 128 64 32 16 8 4 2 1

00111100 = 60

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0
 128 64 32 16 8 4 2 1

00001101 = 13

Core 11

A	B	A AND B
----------	----------	----------------

0	0	0
0	0	0
1	0	0
1	0	0
1	1	1
1	1	1
0	0	0
0	1	0

00001100

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0
128 64 32 16 8 4 2 1

12

Core 12

A	B	A OR B
0	0	0
0	0	0
1	0	1
1	0	1
1	1	1
1	1	1
0	0	0
0	1	1

00111101

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

128 64 32 16 8 4 2 1

61

Core 13

A	B	A XOR B
0	0	0
0	0	0
1	0	1
1	0	1
1	1	0
1	1	0
0	0	0
0	1	1

00110001

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0
128 64 32 16 8 4 2 1

49

Completion 2

The screenshot shows a code editor with a C program that demonstrates bitwise operations. The program defines two unsigned integers, A and B, and performs AND, OR, and XOR operations on them. It also shows bit shifts and a final output window.

```
1 #include <stdio.h>
2
3 int main(){
4     unsigned int A = 60;
5     unsigned int B = 13;
6
7     // AND is A & B
8     // OR is A / B - this is the 'pipe' symbol
9     // XOR is A ^ B
10
11     // A << 2 indicated a bitshift of A 2 places to the left
12     // B >> 1 indicates a bitshift of B 1 places to the right
13
14     //For example
15     printf("%d", A & B);
16     printf(" is the AND operation ");
17     printf("%d", A / B);
18     printf(" is the OR operation ");
19     printf("%d", A ^ B);
20     printf(" is the XOR operation ");
21     return 0;
22 }
23
24
```

The output window shows the results of the operations:

```
12 is the AND operation 4 is the OR operation 49 is the XOR operation
Process returned 0 (0x0) execution time : 0.005 s
Press ENTER to continue.
```

The bottom of the screenshot shows the 'Logs & others' panel with the following messages:

```
=== Build file: "no target" in "no project" (compiler: unknown) ===
=== Build finished: 0 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===
```

Challenge 3

Challenge 4

A	B	$(A \cdot B)$	$(A + B)$	$(A \cdot B) + (A + B)$
1	1	1	1	1
1	1	1	1	1
1	0	0	1	1
1	0	0	1	1
0	1	0	1	1
0	1	0	1	1
0	0	0	0	0
0	0	0	0	0

\curvearrowright
=
 $\therefore (A \cdot B) + (A + B) = (A + B)$