

C - PROGRAM STRUCTURE

It illustrates[Describes] how to write a program in c-language.

Every programming language is having a particular structure and we should have to follow this structure.

C-Programming structure is divided into the following parts.

- **[documentation section]**
- **Header files / Proto types / Preprocessor**
- **[global variables]**
- **[function declarations & definitions]**
- **void main() / main() / int main()**
- **Other statements.**

Generally documentation section consists of program headings, definitions etc and They should be represented with comments.

The statements that are enclosed in between `/*` and `*/` are called comments.

Comments never participate in program execution. They are only for user understandability or display purpose.

C-Language supports comment block only.

Eg:

```
/*  
.....;  
.....;  
*/
```

C++ supports comment block and single line comments.

Eg: `//`

Header files consists of function definitions, global variables, macros etc.

We can declare the header files at any place of our program. But before going to use the relevant function, its header file should be declared. It is recommended to declare the header files at the top of the program.

Every header file should be started with **#include**. Here **#** is a **preprocessor** indicator.

We can place header files in angled brackets **< >** or double quotes **" "**.

Header file never ends with **semicolon(;)**.

Note: In C++, we should have to declare header files at the top only.

The variables that are declared before **main()** or top of the program are called **global variables** and they can be accessed from anywhere in our program. They are optional.

Function declarations and definitions contain function header and body.

- * Every C-Program execution starts from main() function and travel towards down. Hence it is also called **top-down** approach.

- * The operating system is calling the main() for execution.

- * Without main(), C-Program never executed but compiled.

- * main() is predefined function with user defined body. main() doesn't have any header file. One program have to maintain one main() only. **We can create alternate for main().** Other statements are changed from program to program.

Note: It is recommended to write C programs in lower case only. Every statement should have to end with semicolon main().

Finding no of conversion characters in scanf():

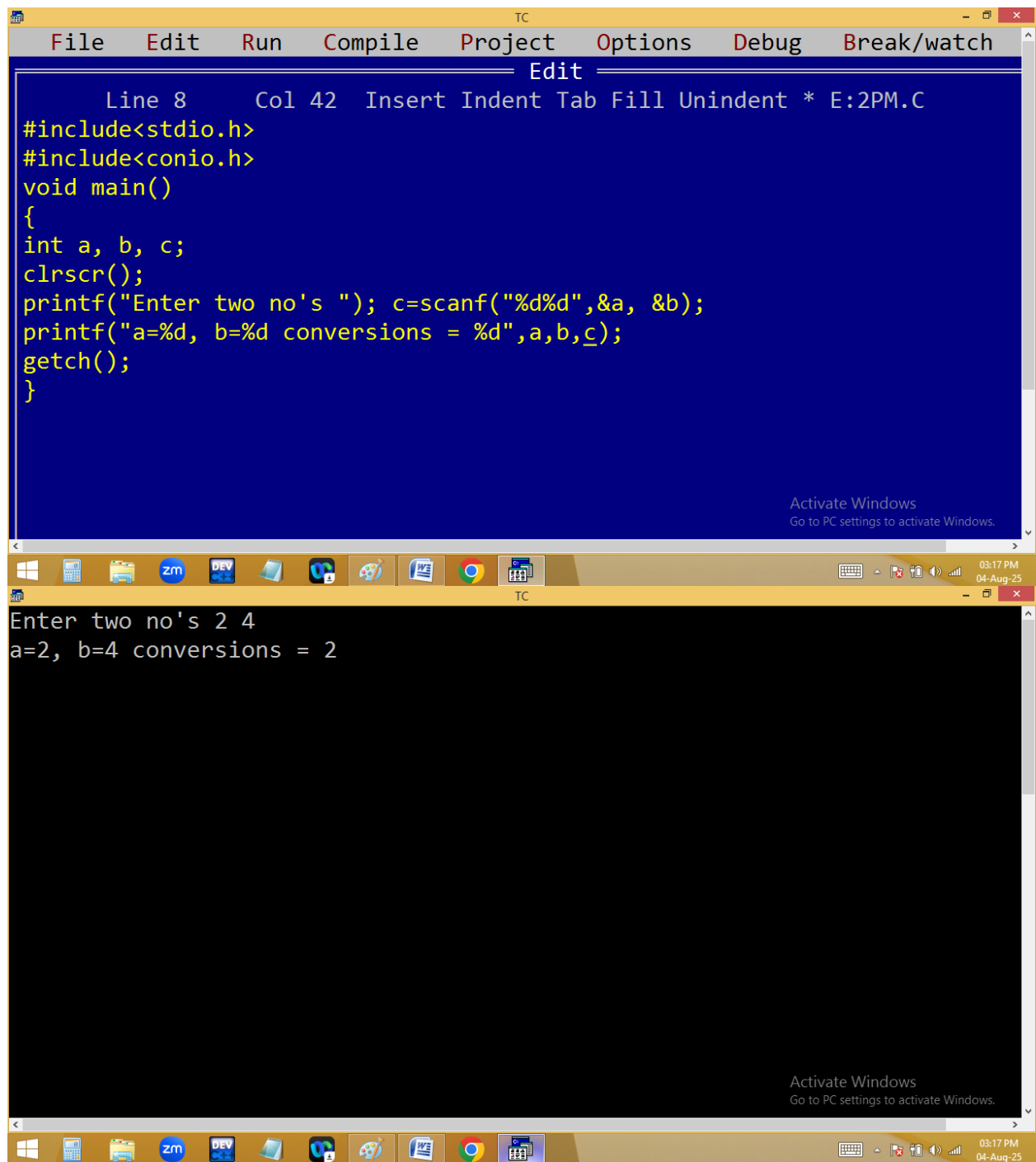
The image shows a screenshot of the Turbo C++ (TC) IDE. The top window is the 'Edit' window, which contains the following C code:

```
Line 2      Col 9      Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
printf("conversions = %d",scanf("%d%d"));
getch();
}
```

The bottom window is the 'Output' window, which displays the execution results:

```
100
200
conversions = 2_
```

The Windows taskbar at the bottom shows the time as 03:15 PM on 04-Aug-23. The system tray includes icons for keyboard, network, and volume. An 'Activate Windows' watermark is visible in the bottom right corner of the output window.



The image shows two screenshots of the Turbo C++ (TC) IDE. The top screenshot displays the source code of a C program in the 'Edit' window. The code includes `<stdio.h>` and `<conio.h>`, and defines a `main` function that declares variables `a`, `b`, and `c`, clears the screen, prompts the user for two numbers, reads them using `scanf`, and prints the values of `a`, `b`, and the number of conversions (which is `c`). The bottom screenshot shows the program's execution output in the 'TC' window. It displays the prompt 'Enter two no's' followed by the user input '2 4', and then the output 'a=2, b=4 conversions = 2'. Both windows show a Windows taskbar at the bottom with various application icons and a system clock indicating 03:17 PM on 04-Aug-23.

```
File Edit Run Compile Project Options Debug Break/watch
Edit
Line 8 Col 42 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a, b, c;
clrscr();
printf("Enter two no's "); c=scanf("%d%d",&a, &b);
printf("a=%d, b=%d conversions = %d",a,b,c);
getch();
}
```

Enter two no's 2 4
a=2, b=4 conversions = 2

Conversions:

```
Line 1      Col 33  Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
printf("%d\n",012); /* octal to decimal */
printf("%o\n",45); /* dec  to octal */
printf("%d\n",0x35); /* hex to decimal */
printf("%x\n",35); /* dec to hexa */
printf("%X\n",45);
printf("%X\n",95);
printf("%x\n",025); /* oct to hexa */
printf("%o\n", 0x44); /* hexa to oct */
printf("%x",-2<<2); /*  fff8 */
getch();
}
```

10
55
53
23
2d
5F
15
104
fff8

octal to decimal

$$\begin{array}{r} 012 \\ | \quad | \\ 8 \times 1 + 8 \times 2 \\ \hline 8 + 16 = 24 \end{array}$$

dec to octal

$$\begin{array}{r} 8 \overline{)45} \\ \underline{5} \\ 5 - 5 = 0 \end{array}$$

hexa to decimal

$$\begin{array}{r} 0 \times 3 5 \\ | \quad | \\ 16 \times 3 16 \times 5 \\ \hline 48 + 80 = 128 \end{array}$$

decimal to hexa

$$\begin{array}{r} 16 \overline{)35} \\ \underline{2} \\ 2 - 2 = 0 \end{array}$$

dec to hexa

$$16 \overline{) 45}$$

2 - 13

hex to dec

$$0 \times 44$$

$$16^1 \times 4 \quad 16^0 \times 4$$

$$64 + 4 = 68 \rightarrow$$

dec to oct

$$8 \overline{) 68}$$

$$8 \overline{) 8 - 4}$$

$$1 - 0$$

10-a
11-b
12-c
13-d
14-e
15-f

16 $\overline{) 95}$
5 - 15

oct to dec

$$025$$

$$8^1 \times 2 + 8^0 \times 5$$

$$16 + 5 = 21$$

dec to hex

$$16 \overline{) 21}$$

$$1 - 5 \checkmark$$

-2 << 2 = fff8

-2 << 2 = -8

8 = 0000 0000 0000 1000

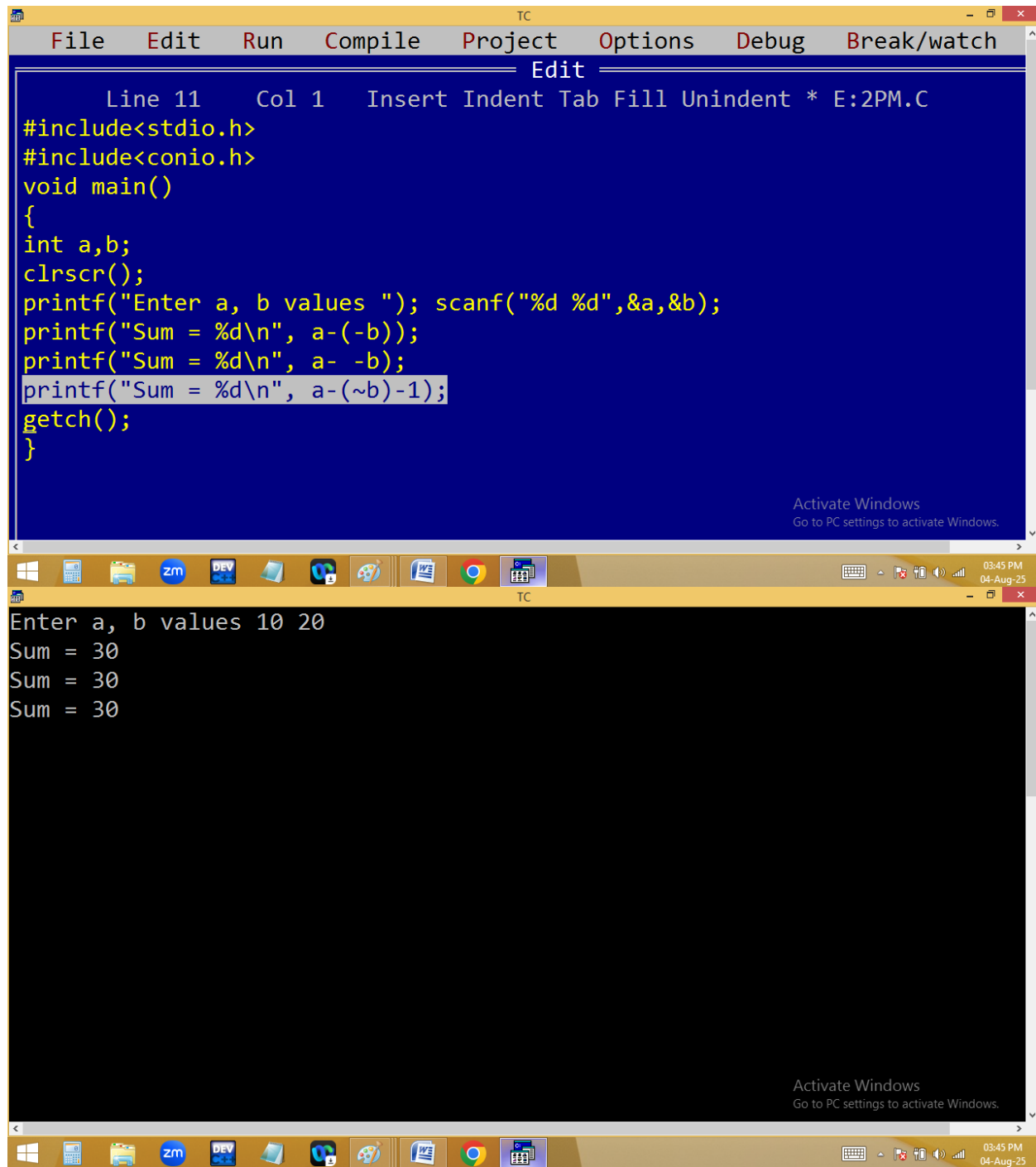
~ = 1111 1111 1111 0111

2~ = 0000 0000 0000 0001

$$\begin{array}{cccc} 1111 & 1111 & 1111 & 0000 \\ \hline f & f & f & 0 \end{array}$$

$$\begin{array}{r} 2 \overline{) 8} \\ 2 \overline{) 4} 0 \\ 2 \overline{) 2} 0 \\ \hline 1 - 0 \end{array}$$

Read two numbers and add without using + operator?



```
TC
File Edit Run Compile Project Options Debug Break/watch
Edit
Line 11 Col 1 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b;
clrscr();
printf("Enter a, b values "); scanf("%d %d",&a,&b);
printf("Sum = %d\n", a-(-b));
printf("Sum = %d\n", a- -b);
printf("Sum = %d\n", a-(~b)-1);
getch();
}
```

Enter a, b values 10 20
Sum = 30
Sum = 30
Sum = 30

a=10

b=20

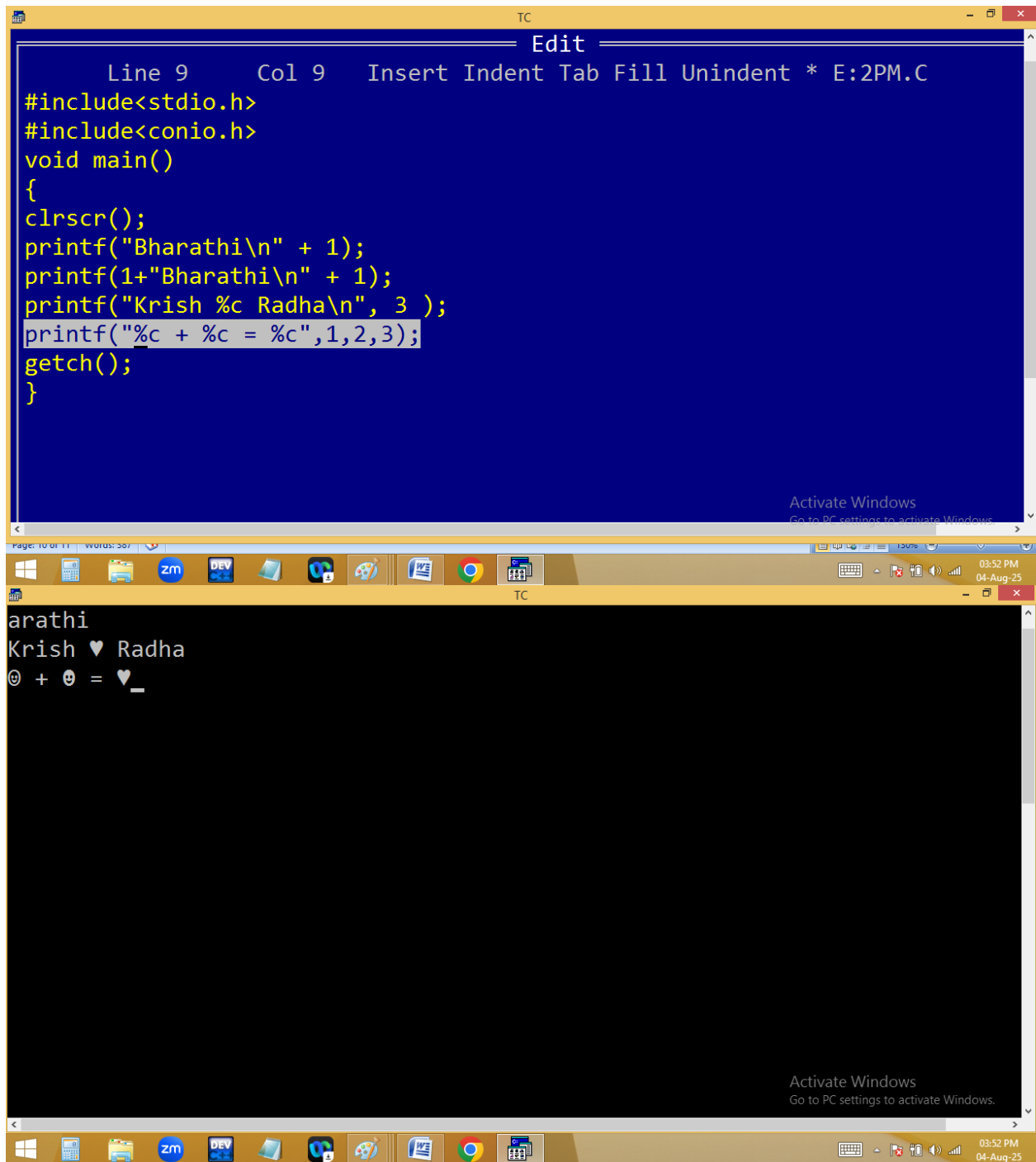
a-(~b)-1

n = -(n+1) ==> -(20+1)==> -21

10-(~20)-1

10-(-21) -1

10+21-1=30



```
TC
Edit
Line 9 Col 9 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
printf("Bharathi\n" + 1);
printf(1+"Bharathi\n" + 1);
printf("Krish %c Radha\n", 3 );
printf("%c + %c = %c",1,2,3);
getch();
}
```

arathi
Krish ♥ Radha
☺ + ☹ = ♥_