C PROGRAMMING STANDARDS

1.INT MAIN() FUNCTION

It is mandatory to use int data type before the main function and return 0. It is not recommended to use void,use it only when you are not returning any value.

2. **NEVER USE KEYWORDS AS A VARIABLE NAME**

We know that there are some preprocessors in the C programming, so when compiler starts compiling our program which is having keyword names as variable name there is a possibility that it will confuse between it's variables and our variables and may crash.

3. DON'T DEFINE A VARIABLE NAME STARTING WITH NUMBERS OR SPECIAL CHARACTERS

Use logical variable names to the variables based on the problem to avoid any further confusion. Variable names can start with uppercase,lowercase or with underscore.

EX: If you are finding the different sums use sum 1,sum2 like that instead of your names. So that it will be understandable for others too.

Don't give space if your variable name is having two words use underscore to combine.

4. ALWAYS REPRESENT A BLOCK OF CODE WITHIN FLOWER BRACKETS{}

It is mandatory to represent everything within blocks, so that compiler should understand what lines to iterate within a block and how to proceed further. If any **LOOP** or IF STATEMENT is having single statement it may not require to keep flower brackets,but if there are many statements then you should .

5. ALWAYS SPECIFY AN INITIAL POINT , A CONDITION AND AN INCREMENT VALUE INCASE OF LOOPS.

If you fail to do them, then program won't be in our control, it may lead to several error or sometimes infinite loop.

EX: int i=1;

```
while(i<=10)
{       printf(i);
        i=i+1;
}
printf("HELLO WORLD");
```

OUTPUT=1,2,3,4,5,6,7,8,9,10
HELLO WORLD

6.ALWAYS KEEP SEMICOLON AFTER WRITING A STATEMENT

Never leave any statement without keeping a semicolon at the end, otherwise it will throw an error.

But if you keep semicolon at the end of any conditional statement or loops ,it will consider them also as statement instead of conditional statements

7. BE CAUTIOUS WHEN YOU ARE USING CONTROL STATEMENTS

EX: BREAK: Exists from the loop in which that loop is present
CONTINUE: Exists from that iteration and then continues from next iteration of that loop.

8. **AVOID SPECIFYING THE SIZE OF THE ARRAY**
Sometimes it may prove to be deleterious if we enter a size greater than that of the array.

9. MAKE USE OF SWITCH STATEMENTS INSTEAD OF NESTED IF STATEMENTS

They made the program look simple and clean.

10. NEVER TRY TO DO ARITHMETIC OPERATIONS ON DIFFERENT DATA TYPES

11. ALWAYS PASS THE SAME COUNT OF ARGUMENTS TO THE FUNCTION

12. NEVER LEAVE POINTERS UNINITIALISED
It may point to some random memory locations and may cause the system to crash

13. NEVER LEAVE VARIABLES UNINTIALISED
It may store garbage values in it if you don't intialise and will give you the wrong output.