Board has PWM mapped on the PB04

It supports native PWM the Timer.

---

logical questions

1) int a=1;
   int b=0;
   Printf("%d", a&&b||!a); =0

2). **volatile**

The volatile keyword in c is a type qualifier that inform the compiler that a variable value may change

3). **int fun() and int fun(void)**

int fun()	→ Function with unspecified arguments.	yes (compiler allow arguments)
int fun (Void)	→ Function with no arguments at all	not all arguments

4). **break is used in loop**

It will terminates the loop still the condition is ~~for~~ true.

5) Printf("%d", sizeof(i));

=Ans : 4

6) int a=1, b=2

int c=a+++b;

Print(c)

$\overset{1}{a++} + \overset{2}{b}$

$= 3$

---

7). int a=5

Print("%d", ++a++)

first it will do post increment then it will try to do pre increment.

It is an l value error

---

8).

8 4 2 1

5 — 0 1 0 1

1's = 1 0 1 0

2' 1 0 1 0
      1

1 0 1 1

— 5 ←

← 0 0,0 1 1 0

1 1 1 0 0 1

10

← 1 1 1 0 1 0

9). what happens if you access an array out of bound?

It may compiles, may run but it is undefined behavior.

① segmentation fault

② may overwrite other variables

③ show garbage output

④ corrupt memory stack.

ex

```
int arr[3] = {10, 20, 30};
printf("%d", arr[3]);
```

$\Rightarrow$ It returns the garbage value.

```
int a = 5  printf
{
int a = 10;   printf(a).
int
}
```

[Scope of a variable]

Outp
5
10.

---

can a function return an array in C,

No directly not.

using pointers we can

```
int i;
for (i=0; i<10; i++);
printf("%d", i);
```

// $\boxed{10}$

Due to ;

Inside loop nothing

will happen

1 1<10 i++ → 2

2 2<10 i++ → 3

: : :

8 8<10 i++ → 9

9 9<10 i++ → $\boxed{10}$

10 will be the output.

$\boxed{10}$ $\underset{\times}{\underline{10<10}}$

Int *ptr; //already it contains garbage,
*ptr = 5; // Trying to write the 5 in the garbage...

~~exa~~ ~~affects unoffiadi~~ garbage

[undefined behavior]

1] may garbage value.
2) Segmentation fault
3). corrupts random memory

Compiler - okay
CPU - mmm okay
memory controller ⇒ Nope

---

const int x=10;
x=20;

x=10 already declared in constant so again we can't change its value.