Important Question
Pointer

Q1: Consider the following two statements

```
int *p = &i;
p = &i;
```

The first statement is declaration and second statement is simple assignment statement. why isn't the second statement, p is preceded by * symbol?

(*p) → value of the location

(p) → location → *p = &i mean go to the location change its value to address which isn't possible.

In C, * symbol has different meaning depending on the context in which it is used.

At the time of declaration, * symbol isn't acting as an indicator operator/dereference operator. * symbol in the first statement tells the compiler that p is a pointer to an integer.

But if we write *p = &i; then it is wrong, because here, * symbol indicates the dereference operator and we cannot assign the address of some integer variable (p=&i;) means assigning the address to a pointer.

Q2: What is the Output of the following program?

```
void fun (const int *p) {
    *p = 0
}
int main() {
    const int i = 10;
    fun (&i);
    return 0;
}
```

Output = ~~0~~

Output = ERROR

cos "const int"
so we can't change the value even with pointer!

Q3: ~~to~~ How to print the address of a variable?

Use %p as a formal specifire in printf function.

```
int main() {
    int i = 10;
    int *p = &i;
    printf ("Address of i is %p", p);
    return 0;
}
```

[Output will be in Hexadecimal format]

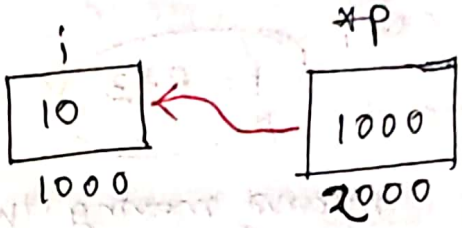
Q4: If i is a variable and p points to i, which of the ~~four~~ following ~~statements~~ expressions are aliases of i?

a) *p b) *&p c) &p d) *i e) *&i

both will gave same value as (i)

Example:

```
int i = 10;
int *p = &i;
```



a) *p = *(1000) = 10

go to the address 1000 an access its value.

b) *&p = *(2000) = 1000

c) &p = 2000

d) *i = *(10) [It doesn't make sense coz, 10 is a value not address]

e) *&i = *(1000) = 10