

## if-else assignment 1-july.

Question 1 : 

```
int num=20;
if(num>20){
    printf("Num is greater than 20");
}
printf("num: %d\n", num);
```

o/p:- num: 20

Reason → the if condition says num greater (>) than 20 not equal to, if this was condition then it would enter the if block but condition is num>20 & num is 20, so it skip the if block & prints the line below it.

2) 

```
int a = 10, b = 20;
if((a < b++) && (a++ || ++b)) {
    printf("a: %d\n", a);
    printf("b: %d\n", b);
}
printf("out of if");
```

O/P:- a: 11  
b: 21  
out of if.

Reason:- ①  $a < b++$   
 $10 < 20 = \underline{1}$

②  $a++ || ++b$   
 $11 || \text{---} = \underline{1}$

$\swarrow$   $\boxed{20}$  21  
temp  $\boxed{0}$  20

$\swarrow$   $\boxed{10}$  11  
temp  $\boxed{0}$  10

In if the condition is true (1) as  $1 < 1 = 1$   
so it will enter if block & execute it

In if block the value of a will be  $\boxed{11}$   
so above mentioned and as the a is true  
it will not check b because  $1 || \text{anything} = 1$   
so ans is a: 11 b: 21

```

3) int ilc = 15, oic = 25;
   if (oic > ilc) {
       printf("oic: %d\n", oic);
   }

```

```

3.
   if (ilc < oic) {
       printf("ilc: %d\n", ilc);
   }
   printf("oic: %d\n", oic);
   printf("ilc: %d\n", ilc);

```

ans: oic : 25  
 ilc : 15  
 oic : 25  
 ilc : 15

Reason: ① 1st if block has condition true so it will get executed.

2nd if block is also true condition but it will get binded with ';' (Null statement)

(Null statement)  
 semicolon

```

if (ilc < oic) {
    // Null statement
}
printf("ilc: %d\n", ilc);

```

binded. separate

Q bool val = false, var = true

```

if (val) {
    printf("true");
}
if (var) {
    printf("false");
}

```

```

if (val) {
    // Null statement
}
printf("true");

```

binded. separate

ans: true } If we include stdbool.h but if we not then error.

- ① - Bool → should be used
- ② val → 0 instead of false
- ③ var → 1 instead of true

```

5) float x = 5.2, y = 10.5;
   if (x == 5.2) {
       printf("x: %.d\n", x);
   }
   if (y == 10.5); {
       printf("y: %.d\n", y);
   }
}

```

ans: will give warning for %.d as it is float not integer. 1st if block doesn't get executed and 2nd if has null statement but y value is garbage value.

Reason:- if block accepts only .5 values not greater not lesser only .5 eg 5.1, 1.5, 2.5, 3.5, etc. ...

```

6) int subVal = 15;
   if (subVal++ < 6 & subVal--);
       printf("subVal: %.d\n", subVal);
}
   if (subVal-- || ++subVal);
       printf("subVal: %.d\n", subVal);
}
   if (subVal > subVal); {
       printf("subVal: %.d\n", subVal);
   }
}

```

ans: subVal: 15  
subVal: 14  
subVal: 14

Reason.

1st if block:  $\text{subVal}++ \& \text{subVal}--$

	15	&	16
sub	<u>15</u>		
temp			15
sub	<u>16</u>		15
temp			16



2<sup>nd</sup> if block.

subVal -- 11 ++ subVal  
sub 14 15 11 does not executed becoz  
or got true 1st  
temp 15

3<sup>rd</sup> if block.

subVal > subVal;

printf("subVal: %d")

sub 14 sub 14 ← 14

7) int weight = 50;

if (weight < 50 && weight > 30) {

printf("you are under weight")

}

if (weight == 50 && weight <= 70) {

printf("you are well maintained")

}

if (weight > 70) {

printf("work hard");

}

ans: you are well maintained.

you are overweight you have to work hard

8) int year = 2020.

if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0) {

printf("%d year is leap year\n");

}

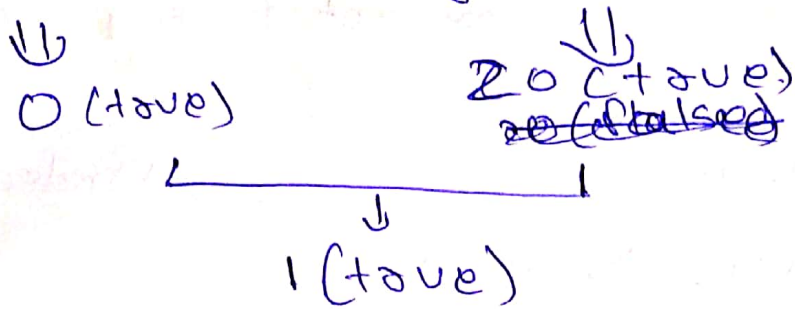
ans:- In above code there is no year in pf.

i.e printf("%d year is leap year\n", year)

so the value is garbage but I'll assume that  
there is so the value is 2020 is leap year

Reuse ~~in~~ in if block the condition is,

$(year \% 4 == 0 \& \& year \% 100 != 0) \vee year \% 400 == 0$



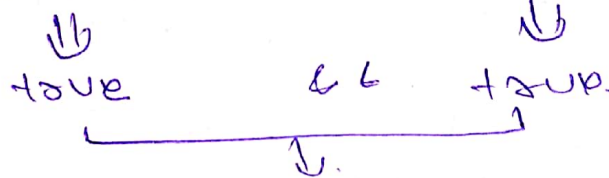
Does not execute as  
or check if  
there is false  
first.

so if block is 1 (true) & executes it.

Q. ans: 10 is positive  
-5 is negative.  
number is zero

Reason: 1st if block.

①  $if (number != 0 \& \& number > 0)$



1st if block is executed.

② 2nd if block.

②  $if (number < 0)$

↓  
true. → As ~~number~~ - number was initialized to -5

2nd if block is executed.

③  $if (number == 0)$

↓  
true → number was initialized to 0



if (number % 4 == 0)

↓  
false → number was initialized  
to 15 & 15 % 4 != 0

3<sup>rd</sup> if block is not executed.

10) int aNum = 0;  
char aChar;

if (!aNum) {

printf("Inside first if... \n");

aNum = 20;

aChar = "T";

}

if (aNum % 4 != 0) {

printf("Inside second if... \n");

aNum = 20;

aChar = "F";

}

if (aNum == 0) {

printf("Inside Third if... \n");

aChar = "T";

}

printf("After all if, values are: %d & %c",  
aNum, aChar);

}

Reason:

1<sup>st</sup> if block:

!aNum

~~is~~

~~00000000~~

is true so

1<sup>st</sup>

block is executed.

2<sup>nd</sup> if block.

(aNum % 4 != 0) = (0 != 0) = false  
not executed.

3rd if block.

$num == 0 \Rightarrow 20 == 0 \Rightarrow \text{false.}$   
Not executed.

so answer is:

After all if, values are  $20 < T$

1) o/p: Enter two numbers.

55 66.

$55 < 66.$

Enter two numbers

55 55

$55 = 55.$

Enter two numbers.

55 2

$55 > 2.$

Reason: Using input fun<sup>n</sup> i.e scanf

2) 120 is divisible by 2

120 is divisible by 3

120 is divisible by 5.

Reason:- 120 is divisible by 2 so it enters inside 2<sup>nd</sup> if block i.e  $(num \% 3 == 0)$

120 is also divisible by 3, so further it enters into 3<sup>rd</sup> if block  $(num \% 5 \neq 0)$ . 120 is also divisible by 5

② 30 is divisible by 2

30 is divisible by 3

30 is divisible by 5

Reason:- Above reason all apply's here too...

③ 76 is divisible by 2

76 is <sup>not</sup> divisible by 3.

Reason:- 76 is divisible by 2 but not by 3 so it enters into  $(num \% 2 == 0)$



if block, executes if moves value to num.  $3 == 0$ , but it is not a valid condition, so it goes into else block.

- ③ 36 is divisible by 2  
36 is divisible by 3  
36 is not divisible by 5

Reason: 36 enters into 1st if block, condition is valid, executes it, moves to  $(num \% 3 == 0)$  is a valid condition executes it, move to  $(num \% 5 == 0)$ , not valid condition goes to else

13) ans: float n1, n2, n3;  
printf("Enter n1, n2, n3: \n");  
scanf("%f %f %f", &n1, &n2, &n3);

Reason: Using scanf we can store the address & printf with point the content of that address.

14) ans: int num;  
printf("Enter number: \n");  
scanf("%d", &num);  
if (num >= 25 && num <= 50) {  
    printf("%d is number 25 to 50 \n", num);  
}  
else {  
    printf("%d not ranges from 25 to 50 \n", num);  
}

2.



```

15) char c
printf("Enter a capital alphabet: \n");
scanf("%c", &c);
if (c == 'A' || c == 'E' || c == 'I' || c == 'O' ||
    c == 'U') {
    printf("%c is vowel \n", c);
} else {
    printf("%c is not a vowel \n", c);
}
}

```

```

16) ans:- char ch;
printf("Enter : ");
scanf("%c", &ch);
if (ch >= 'A' && ch <= 'Z') {
    printf("Upper case \n");
} else if (ch >= 'a' && ch <= 'z') {
    printf("Lower case \n");
} else {
    printf("special character \n");
}
}

```

~~17) ans:- if (n1~~

```

17) int n1, n2, n3;
printf("Enter n1, n2, n3 \n");
scanf("%d %d %d", &n1, &n2, &n3);
if (n1 >= n2 && n1 >= n3) {
    if (n1 % 2 != 0) {
        printf("odd");
    }
}
}

```

```

if (n2 > n1 & n2 > n3) {
    if (n2 % 2 != 0) {
        printf("odd")
    }
}

```

```

if (n3 > n1 & n3 > n2) {
    if (n3 % 2 != 0) {
        printf("odd")
    }
}

```

O/P:- ① Enter n1, n2, n3:

1  
2  
3

Largest number is 3 & 3 is odd

② Enter n1, n2, n3:

20  
21  
12

Largest number is 21 & 21 is odd.