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 = 1

② a++ || ++b

1 || 1 = 1

b 20 → 21
temp 0 20

a 10 → 11
temp 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 11
so above mentioned and as the a is true
it will not check b because 1 || 0/1 = 1
(0/1 ↑ anything)
So ans is is a: 11 b: 21

3) int ilc = 15, olc = 25;

```
if (olc > ilc) {
        printf("olc: %d\n", olc);
    }
if (ilc < olc); {
        printf("ilc: %d\n", ilc);
    }
    printf("olc: %d\n", olc);
    printf("ilc: %d\n", ilc);
}
```

ans: olc : 25
ilc : 15
olc : 25
ilc : 15,

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

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

(Null statement) semicolon

```
if (ilc < olc);{
}
pf("ilc: %d\n", ilc)
```

binded.

seperate.

4) bool val = false, var = true.

```
if (val); {
        print("true\n");
    }
if (var){
        printf("false\n");
    }
```

```
if (val)
    ;{
pf ....
}
```

binded.

seperate

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

(1) _Bool → should be used
(2) val → 0 insted 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 2nd if has null statement but y value is garbage value.

reason:- if block accept 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 ++ && subVal--) {
        pf("subVal : %d \n", subVal);
}
if (subVal-- || ++subVal) {
        pf("subVal : %d \n", subVal);
}
if (subVal >= subVal), {
        pf("subVal : %d \n", subVal
}
```

ans: subVal : 15
subVal : 14
subVal : 14

Reason.

1st if block = subVal ++ && subVal --
15 && 16

sub 16

temp 15

sub 15

temp 16

~~3.~~ 2nd if block.

~~1~~ 14 subVal -- // ++subVal

sub. [~~15~~] 15 // Doest executed breoz or got true 1st

temp [~~0~~] 15

3rd if block.

subval > subVal; → binded.

printf("subval : %d"

~~sub~~ [~~15~~] sub - [14] ← 14

7) int weight = 50;

```
if (weight <50 && weight > 30) {
        pf("your are under weight"
}
if (weight >= 50 && weigh <= 70) {
        pf("you are well maintained");
}
if (weight > 70); {
        pf ("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 pf("%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

Reason:- in if block the condition is,

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

↓ 0 (true)

20 (true)

↓ 1 (true)

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

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

a). ans: 10 is positive
-5 is negative.
number is zero

Reason: 1st if block.

① if (number != 0 && number > 0)

true && true.

↓ true.

1st if block is executed.

② 2nd if block.

② if (number < 0)

↓ true. → As 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

3rd if block is not executed.

10) int aNum = 0;
char aChar;
if (!aNum) {
pf("Inside first if...\n");
aNum += 20;
aChar = "T";
}.

if (aNum % 4 != 0) {.
pf("Inside second if..\n");
aNum += 20;
aChar = "F";
}

if (aNum == 0) {.
pf("Inside Third if ...\n);
aChar = "7";
}
printf("After all if, value are %d & %c",
aNum, aChar)
}.

Reason:
1st if block:
!aNum is true so 1st block is executed.
2nd if block.
(aNum % 4 != 0) = (0 != 0) => false
not executed.

3rd if block.

anum == 0. => 20 == 0 => false.

Not executed.

so answer is;

After all if, values are 20 & T

11) o/p: Enter two numbers.

55 66.

55 < 66.

Enter two number

55 55

55 = 55.

Enter two number.

55 2

55 > 2.

Reason: Using input fun^n i.e scanf

12) ① 120 is divisible by 2

120 is divisible by 3

120 is divisible by 5.

Reason:- 120 is divisible by 2 so it enter inside 2nd if block i.e (num%.3=0) 120 is also divisible by 3, so further it enter into 3rd if block (num%.5==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 allys apply's here too...

③ 76 is divisible by 2

76 is not divisible by 3

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

if block, executes if moves future to num%.3==0, but it is not 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 enter into 1st if block, condition is valid, executes it, moves to (num%.3==0) is 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 print the content of that address

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

15)
```
char c
printf("Enter a capital alphabet: \n");
scanf("%c", &c);
if (c == 'A' || c == 'E' || c == 'I' || c == 'O' ||
    c == 'U'){
        pf("%c is vowel\n", c);
else {
    pf("%c is that a vowel\n", c);
  }
}
```

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

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

17)
```
int n1, n2, n3;
pf("Enter n1, n2, n3\n");
sf("%d %d %d", &n1, &n2, &n3)
if (n1 >= n2 && n1 >= n3){
        if (n1 % 2 != 0){
            pf("Odd");
  }
}
```

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