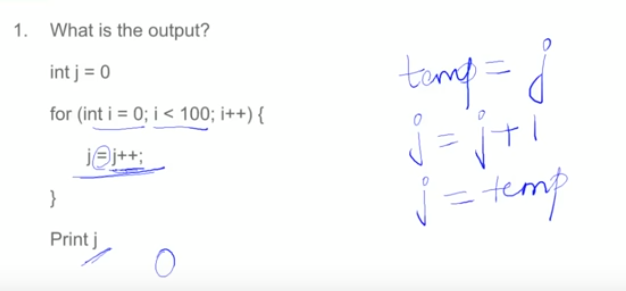
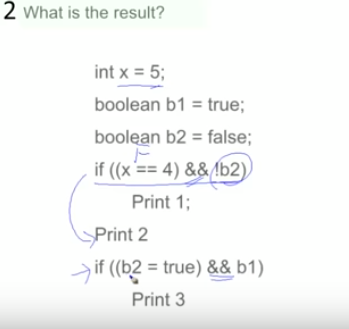
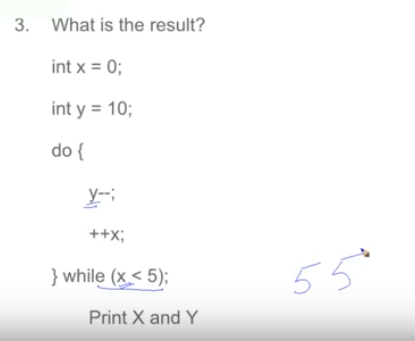
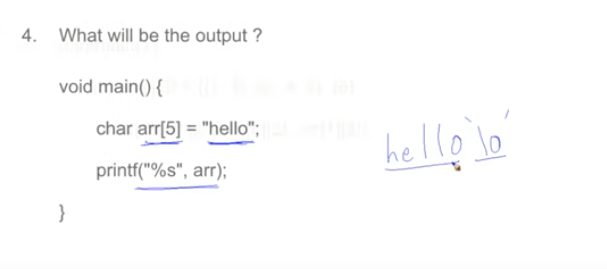
****

****

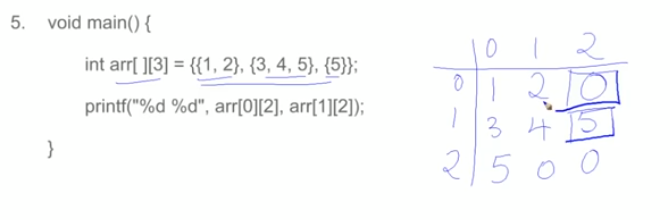
**2.Output:-print 2**

****

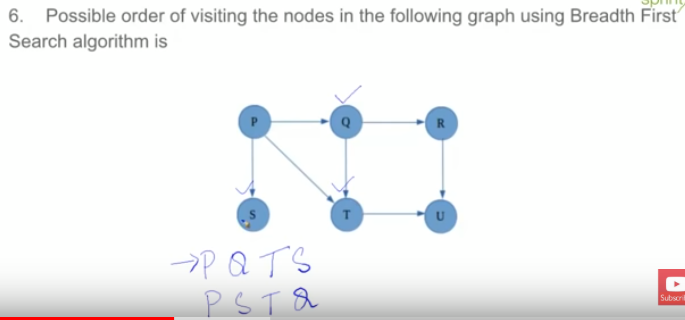
**3.output:-5 5(firstly enter in the loop so in final stage ,first x=5 and y=5,then check 5<5.so condition is false.)**

****

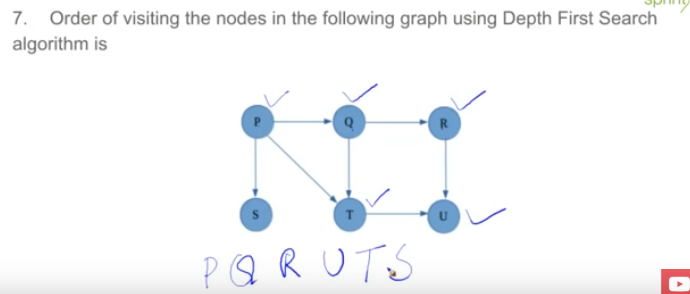
**When we assign a string to an Array of character the null value i.e % also need location Array size is 5 but String is end with % ,so it print :-hello and some garbage value like hello232.**

****

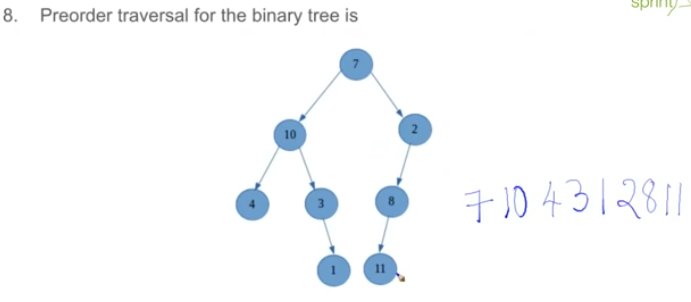
**Output : 0 5 If no value is given then 0 zero is consider for int datatype.as above**

****

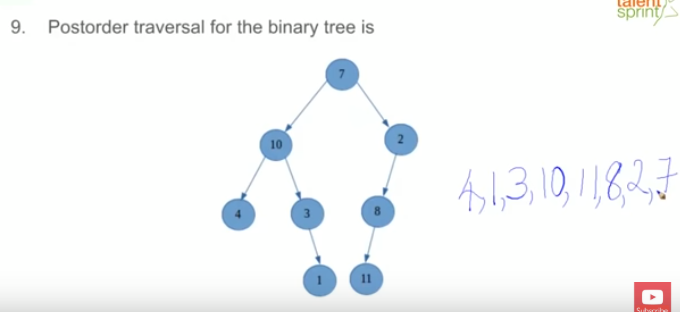
**6. Start with P and then QTSRU or STQ RU**

****

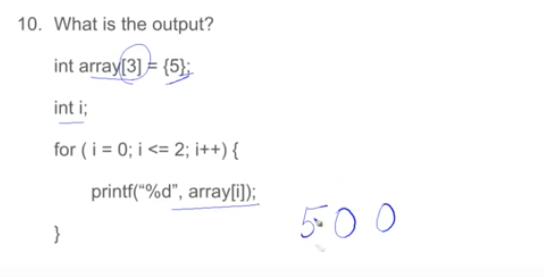
**7.Start with P and then PQRUTS**

****

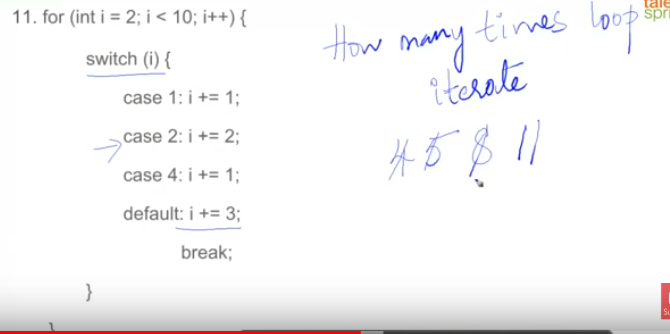
**8. 10 7 10 4 3 1 2 8 11**

****

**9. 4 1 3 10 11 8 2 7**

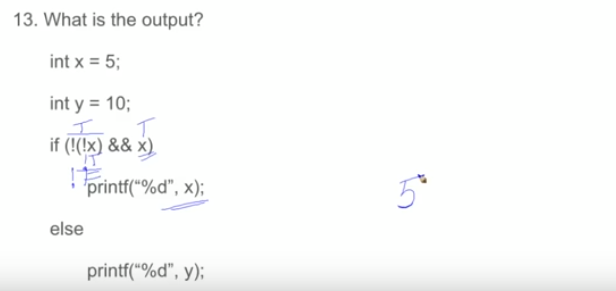
****

**Array[3]={5,0,0} So output like this: 5 0 0**

****

**11. First i=2 so goes to Case-2 i=4 then Case-4 i=5 Case-default i=8**

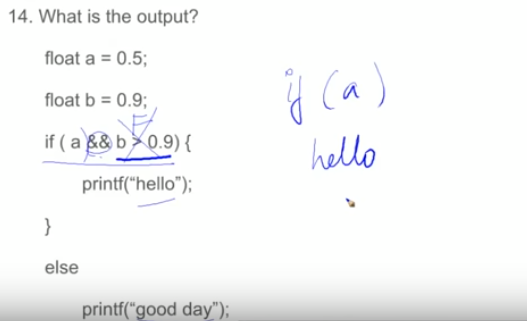
**Now in loop I=8<10 ,So case-8 not present in program so it goes to default-case and i=11 and finally loop terminate with i=11**

****

**13. First inner loop (!X) :- only X consider that is X=5 this is non-zero value so it true .then consider !X , so ‘not true’ i.e it will false .so (!X)=false.**

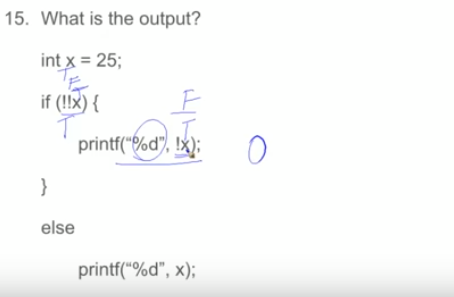
**Second !(!X)=!(false) becomes true**

**Finally True=True,so print X=5.**

****

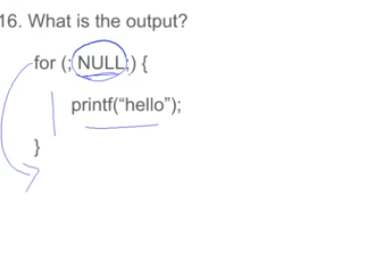
**Consider only if(a),so output will be ‘Hello’. means a is not zero, so it is true .**

**Next if(a && b>0.9 )=true=false means false so print Good Day.**

****

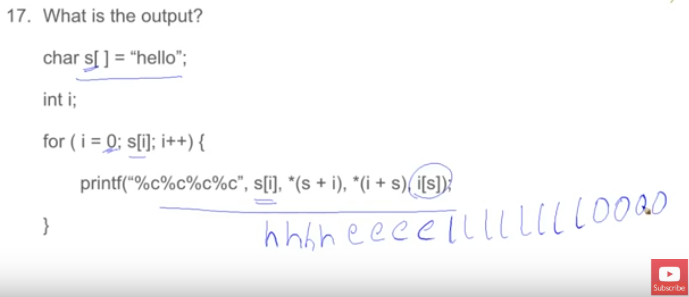
**15. if (!!X) return true (first x is true then !X return false then !!X not false means true).**

**Print(“%d”,!x) print false i.r ) zero in int type.**

****

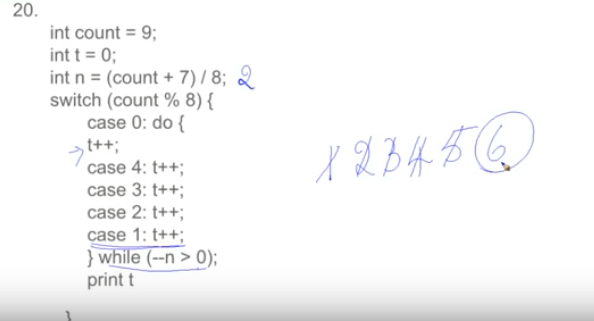
**NULL means ‘No condition’ so ‘No condition’ then ‘No Output’.**

**If for( ; ;) i.e it is infinite loop, but if NULL then ‘No condition’.**

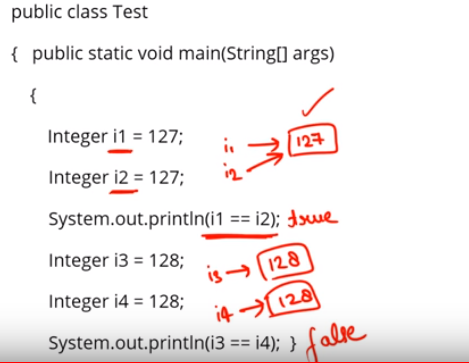
****

**17. s[i]==i[s] and \*(s+1)==\*(i+s)**

**Ouput:hhhheeeelllllllloooo.**

****

**20. output:- t=6**



23.First give true and give false.

Catche memory stored integer type (also byte ,short long) from -127 to 128.

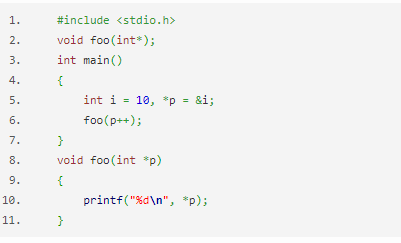
Character type from 0 to 128.

In first case i1 and i2 are object of class interger stored in catche or pool same memory location with value 127.so first give true.

But in second case whenever we assign a value beyond or above the 127 then new object is created ,so new references is created each time….so output is false in second case.

The above case is happen only with ‘integer’ and ‘character’ data type.

If we do that with float or double data type then each time new object is created, and new references is created.



24.Output: 10(in this we perform post increment operation with p,so first pass address then increment address, so previous value printed that pointed by pointer)

**See following Program of pre-increment of address**

#include <stdio.h>

void foo(int \*p);

int main()

{

int i=10;

int \*p;

p=&i;

foo(++p);

return 0;

}

foo(int \*p)

{

printf("%d",\*p);

}

Output:-13335622( Garbage value)

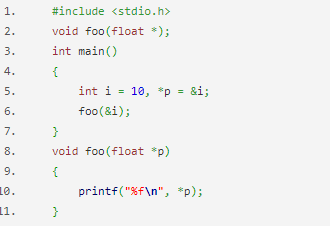
**So finally conclude that pointer ,used pointer with array to point next location.**

1. Perform this operation **if we have consecutive memory i.e Array**.
2. If we perform this operation in above example then we will get **garbage value** as we dont know the value stored at 3060.
3. Consider following example to get idea , how **\*++ptr is used in Array** –
4. #include<stdio.h>
5. **int** main()
6. {
7. **int** arr[5] = {10,20,30,40,50};
8. **int** \*ptr ;
9. ptr = &arr;
10. printf("%d",\*++ptr);
11. **return**(0);
12. }

**Output :**

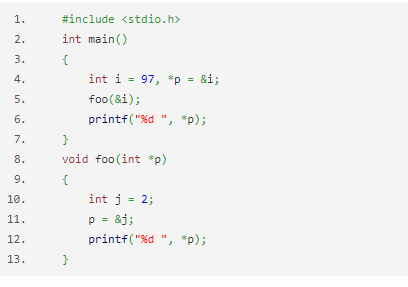
20

Because we increased address not value. Also see next example



Output:-0.0000

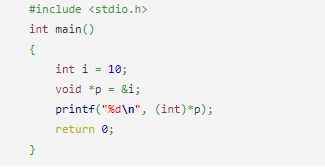
Because printf(“%f”,\*p): print floating value and already declare ‘int’ type of ‘ptr ’.so type casting is required like this: printf("%f\n",(float) \*p);



25.ouput: 2 97

Because: printf("%d ", \*p);//\* p represents value at ptr p has address of j. so \* p represents value at address of j i.e 2 so 2 will get printed.the pointer variable scope is local( it will not used outside the scope)

Then ptr p stored address of ‘I’ and print its value.



26.output:-error,we need type casting like this: printf("%d\n", \*(int\*)p);

27.

#include <stdio.h>

int main()

{

int var = 5;

printf("Value: %d\n", var);

printf("Address: %u", &var); //Notice, the ampersand(&) before var.

return 0;

}

**Output**

Value: 5

Address: 2686778