1. For nested loops, which one is true?
2. We need to use at least one loop inside another loop.
3. We can use more than one loop inside another loop.
4. We can implement the concept of nested loops using the for,while and do while loop.
5. **All of the above.**

Explanation - All of the above things are correct.

2.

while(condition)

{

while(condition)

{

// inner loop statements.

}

// outer loop statements.

}

This is not a correct representation of a nested loop -

1. True
2. **False**

Explanation -We know that , When we use one loop inside another loop .. we call it a nested loop.

3.

#include<stdio.h>

int main()

{

int arr[3]= {1,2,3},i,j;

for(i=0; i<3; i++)

{

for(j=i; j<3; j++)

{

printf("%d ",arr[j]);

}

}

return 0;

}

What will be the output ?

**a) 1 2 3 2 3 3**

b) 1 2 3 1 2 3 1 2 3

c) 1 2 1 2 1 2

d) 1 2 1 2 1 2 1 2

Explanation - Here in the inner loop we start the loop from i ..That’s why the correct answer is a.

4.

int i,j;

for(i=0;i<=5;i++)

{

for(j=0;j<=5;j++)

{

}

}

How many times will the loop run ?

1. 16
2. 25
3. **36**
4. 49

Explanation - The loop will run 36 times because we loop through from 0 to 5 …total 6\*6=36 times.

5.

#include<stdio.h>

int main()

{

char alphabets= {'A','B','C'};

printf("%c\n",alphabets[1]);

return 0;

}

What will be the output ?

1. A
2. B
3. C
4. **Compilation error.**

Explanation - This will give us compilation error .Because here alphabets is a character variable not array of characters.

6.

int arr[5]= {0,-1+1,1+3,1,2};

What will be the value of arr[1] ?

1. **0**
2. 1
3. 2
4. 3

Explanation - answer is 0 . Because -1+1 =0

7.

#include<stdio.h>

int main()

{

char arr[2]= {'a','b'},i,j;

for(i=0; i<2; i++)

{

for(j=\_\_\_; j<2; j++)

{

printf("%c",arr[j]);

}

char ch=arr[0];

arr[\_\_\_]=arr[1];

arr[1]=ch;

}

return 0;

}

Output - abba

You want to print the above output . Now fill the blanks appropriately ?

In the option first value means the inner loop initialization value and the second value means arr[\_\_\_] this section value means after the inner loop section.

1. 1,0
2. 01
3. **0,0**
4. 1,1

Explanation - When the inner loop runs for the first time it will print ab.. And then after the first iteration we just swap the index value that’s why in the second iteration it will print ba. Ultimately the output will be abba

8.

int arr[5]={1,1,2,4,1};

Change the value of index 3 with 3 .

Now which one is correct ?

1. arr[3]=3
2. arr(3)=3;
3. arr{3}=3;
4. **arr[3]=3;**

Explanation - we use [] this to access the index of an array

9.

We can use an integer value and a character (Suppose - a) in the same array -

a)True.

**b)False.**

Explanation - We can not use the different data types in the same array.

10.

#include<stdio.h>

int main()

{

int arr[5]= {1,2,3,4,5},i=5,j;

while(i>=1)

{

arr[i-1]-=i;

while(1)

{

arr[i-1]+=i;

break;

}

i-=1;

}

for(j=0; j<5; j++)

{

printf("%d ",arr[j]);

}

return 0;

}

What will be the final value of the array ?

1. 0 0 0 0 0
2. **1 2 3 4 5**
3. 5 4 3 2 1
4. 1 2 3 4 0

Explanation - We firstly subtract the i from an index then we will again add the same value in the same index in the inner loop section. Ultimately the array remains the same.