Week 1 Questions

Sr. no.	Question	Options	Answer & Explanation

1	Find the output:
	#include<stdio.h>
	int main()
	{
	int x=10; int y;
	{
	y=x++;
	}
	printf("%d",x);
	}
		A. 10
		B. 20
		C. 21
		D. 11
			Answer: D
			Explanation:
			x++ increments the value to 11. So printf statement uses x=11.

2	Find the output:
	#include<stdio.h>
	int main()
	{
	char c='a';
	switch(c){
	case 97:
	printf("97");
	break;
	case 98:
	printf("98");
	break;
	case 99:
	printf("99");
	break;
	default:
	printf("default");
	}
	}
		A. 97
		B. 98
		C. 99
		D. ‘a’
			Answer: A
			Explanation:
			The ASCII value of ‘a’ is 97 which gets switched while switching the char c

3	Find the output:
	#include<stdio.h>
	int main()
	{
	int x=10;
	int a=1,b=2,c=3,d=4;
	x+=a=b*c+d-a;
	printf("%d,%d",a,x);
	}
		A. 1,10
		B. 9,11
		C. 9,19
		D. 1,19
			Answer: C
			
			Explanation: The expression b*c+d-a gets evaluated first, which then sets the value of a to 9. This value is used for incrementing x.

4	Find the output:
	#include<stdio.h>
	int main()
	{
	float a=1.1;
	int b=1.1;
	if(a==b)
	printf("YES");
	else
	printf("NO");
	}
		A. YES
		B. NO
		C. Error
			Answer: B
			
			Explanation:
			int b = 1.1 assigns 1 to b. Comparing a==b compares 1 with 1.1 which is FALSE.

5	Find the output:
	#include<stdio.h>
	int main()
	{
	int i,x=10;
	for(i=0;i<2;i++);
	{
	x++;
	}
	printf("%d",x);
	}
		11	**Answer: B**
			
			Explanation:
			The semicolon after the for loop terminates it and the brackets are treated as a single block, thus x++ is executed only once.

6	Find the output:
	#include<stdio.h>
	int main()
	{
	int i=-10;
	if(i){
	printf("1");
	}
	i=0;
	if(i){
	printf("2");
	}
	i=5;
	if(i){
	printf("3");
	}
	}
		13	**Answer: A**
			
			Explanation:
			Condition is evaluated to true for any non-zero value. So, the first and last IF conditions are evaluated as TRUE.

7	Find the output:
	#include<stdio.h>
	int main()
	{
	int x=10;
	do{
	x++;
	}while(x++>12);
	printf("%d",x);
	}
		A. 10
		B. 11
		C. 12
		D. 13
		E. The program runs for more than 10 iterations.
			Answer: C
			
			Explanation:
			The do loop executes once and goes to the check. In the check, there is one more increment. So the final answer is 12.

9	Find the output: void main(){	A. 2 B. 3	**Answer: C** Explanation:

	int a=22;
	a=a>>4;
	printf("%d",a);
	}
		C. 1
		D. 5
			Right shift 4 times of 22 is 22/16 = 1 (integer division)

10	Find the output:
	void main(){
	int a,b;
	a=3,1;
	b=(5,4);
	printf("%d",a+b);
	}
		A. 7
		B. 8
		C. 6
		D. Error
			Answer: A
			Explanation:
			, can also be used as an operator and it has the least precedence. So in the first statement assignment is done first. In the second, assignment is done later.

11	Which of the following condition checks will print 10 twice?
	
	void main(){
	int a=10;
	if(________)
	printf("\n%d",a);
	else
	printf("\n%d %d",a,a);
	}
		A. a==10
		B. a <= 10
		C. printf("\n %d",a)
		D. None of the above
			Answer : C
			Explanation : printf returns number of characters printed. So the first 10 can be printed using the condition. Since the statement prints more than 0 characters, the IF condition is true and hence the IF part will print one more 10.

12	#include<stdio.h>
	main ( )
	{
	int a=5;
	int b = a % (a−a /2) ∗ (a − 3) + a ;
	printf(”%d”,b);
	return 0 ;
	}
		9	**Answer : B**
			Answer :
			5 % (5 − 5/2) ∗ (5 −3)+5
			= 5 % (5 − 2) ∗ (5 − 3) + 5
			= 5 % 3 ∗ 2 + 5
			= 2 ∗ 2 + 5
			= 9

14	#include<stdio.h>
	
	int main(){
	int i=1;
	i += i*i>>2+3;
		a. 1
		b. 2
		c. 4
		d. 7
			a
			The precedence order is *, +, >> and lastly +=
			Therefore Ans = 1 + ((1 *1)>>(2+3)) = 1 + ((1)>>(5)) =

	printf("%d",i);
	return 0;
	}
	Give the result of compiling and running the above program?
			1+0 = 1

15	#include<stdio.h>
	
	int main(){
	int i=1;
	i += i*i<<2+3;
	printf("%d",i);
	return 0;
	}
	Give the result of compiling and running the above program?
		a. 15
		b. 24
		c. 33
		d. 34
			c.
			
			The precedence order is *, +, >> and lastly +=
			Therefore Ans = 1 + ((1 *1)<<(2+3)) = 1 + ((1)<<(5)) = 1+32 = 3**3**

16	char a,b,c;
	a = 'b';
	b = 'c';
	c = 'A';
	b=c;c=b;
	a=c;c=a;
	printf("A%cB%cC%c",a,b,c);
	What will be the output printed by the printf statement?
		a. AABbCc
		b. AABbCc
		c. AABACA
		d. AcBbCA
			c.
			b = c => b = 'A'
			c = b => c = 'A'
			a = c => a = 'A'
			c = a => c = 'A'

Programming Question #1

Write a program to find the number of perfect squares between given two numbers A and B (both inclusive). A number is called a perfect square if it can be written as x*x for some integer x.

Constraints:
Both A and B are positive. They both are less than 100,000.

Input: Two numbers A and B separated by a space
Output: Count of the number of perfect squares

Example 1:

Input: 3 10

Output: 2

Example 2:

Input: 16 70

Output: 5

Public Test cases:

Number	Input	Output
1.	3 10	2
2.	16 70	5
3.	11 99	6

Private Test cases:

Number	Input	Output
1.	1 100	10
2.	17 124	7
3.	1 2	1
4.	46 47	0
5.	10000 20000	42

Solution:

```
#include<stdio.h>

int main(){
        int x,y,i,a;
        int count=0,flag=0;

        scanf("%d %d",&x,&y);

        for(i=x;i<=y;i++){
                for(a = 0; a <= i; a++)
                {
                        if (i == a * a)
```

```
            {
                flag=1;
            }
        }
        if(flag==1)
            count++;
        flag=0;
    }
    printf("%d",count);
    return 1;
}
```

Programming Question #2

Write a program to find whether a given number (say x) is a “perfect number” or not.

Definition of Perfect number:
A perfect number is a positive integer that is equal to the sum of its proper positive divisors.

Explanation:
Take number 6.
Proper positive divisors of 6 is 1,2,3 and their sum is 1+2+3=6.
So, 6 is a perfect number.

Constraint:
x>=1

Input: A single number x
Output:
yes if given number x is a perfect number
no if given number x is not a perfect number

Example 1:

Input: 6

Output: yes

Example 2:

Input: 7

Output: no

Public Test cases:

Number	Input	Output
1.	6	yes
2.	10	no
3.	28	yes

Private Test cases:

Number	Input	Output
1.	1	no
2.	496	yes
3.	8128	yes
4.	1000	no
5.	4042	no
6.	33550336	yes

Solution:

```
#include<stdio.h>

int main(){
        int x,i,sum=0;

        scanf("%d",&x);

        for(i=1;i<x;i++){
                if(x%i==0){
                        sum+=i;
                }
        }
        if(sum==x){
                printf("yes");
        }else{
                printf("no");
        }
        return 1;
}
```

Programming Qn #3

Write a C program that takes a positive number N and produces an output that is the product of its digits.

Explanation:

Take number 657.

Answer expected : 6 * 5 * 7 = 210

Constraint:

1<=N<=999999

Input: A single number

Output:

The value

Example 1:

Input: 657

Output: 210

Example 2:

Input: 7

Output: 7

<u>Public Test cases:</u>

Number	Input	Output
1.	657	210
2.	41	4
3.	931	27

<u>Private Test cases:</u>

Number	Input	Output
1.	1	1
2.	201	0
3.	1234	24

4.	5555	625
5.	10156	0
6.	999999	531441
7.	124356	720
8.	111111	1

Programming Qn #4

Given three points (x1, y1), (x2, y2) and (x3, y3) , write a program to check if all the three points fall on one straight line.

INPUT:
Six integers x1, y1, x2, y2, x3, y3 separated by whitespace.

OUTPUT:
Print “Yes” if all the points fall on straight line, “No” otherwise.

CONSTRAINTS:
-1000 <= x1, y1, x2, y2, x3, y3 <= 1000

Public Test cases:

Number	**Input**	**Output**
1.	1 0 0 0 3 0	Yes
2.	-2 0 -2 1 -2 2	Yes
3.	-62 14 -18 -23 -6 23	No