**Question 1**

#include <stdio.h>

int main()

{

    int i = 1024;

    for (; i; i >>= 1)

        printf("Quiz");

    return 0;

}

How many times will Quiz be printed in the above program?

1. 10
2. 11
3. Infinite

**Question 2**

Predict the output of the above program?

#include<stdio.h>

int main()

{

   int n;

   for (n = 9; n!=0; n--)

     printf("n = %d", n--);

   return 0;

}

}

1. 9 7 5 3 1
2. 9 8 7 6 5 4 3 2 1
3. Infinite Loop
4. 9 7 5 3

**Question 3**

Predict the output of the below program:

#include <stdio.h>

#define EVEN 0

#define ODD 1

int main()

{

    int i = 3;

    switch (i & 1)

    {

        case EVEN: printf("Even");

                break;

        case ODD: printf("Odd");

                break;

        default: printf("Default");

    }

    return 0;

}

1. Even
2. Odd
3. Default
4. Compile time error

**Question 4**

Predict the output of the below program:

int main()

{

    int i;

    if (printf("0"))

        i = 3;

    else

        i = 5;

    printf("%d", i);

    return 0;

}

1. 3
2. 5
3. 03
4. 05

**Question 5**

#include <stdio.h>

int i;

int main()

{

    if (i);

    else

        printf("Ëlse");

    return 0;

}

What is the output of the above program?

1. if block is executed.
2. else block is executed
3. It is unpredictable as i is not initialized
4. Error: misplaced else

**Question 6**

|  |
| --- |
| #include<stdio.h>  int main()  {     int n;     for (n = 9; n!=0; n--)       printf("n = %d", n--);     return 0;  } |

What is the output?

1. 9 7 5 3 1
2. 9 8 7 6 5 4 3 2 1
3. infinite Loop
4. 9 7 5 3

**Question 7**

In \_\_\_\_\_\_\_, the bodies of the two loops are merged together to form a single loop provided that they do not make any references to each other.

1. Loop unrolling
2. Strength reduction
3. Loop concatenation
4. Loop jamming

**Question 8**

In C++, which system - provided function is called when no handler is provided to deal with an exception?

1. terminate ( )
2. unexpected ( )
3. abort ( )
4. kill ( )

**Question 9**

Which of the following is correct with respect to “Jump Statements” in C?

1. goto
2. continue
3. break
4. All of the above.

**Question 10**

Consider the following program fragment i=6720; j=4; while (i%j)==0 { i=i/j; j=j+1; } On termination j will have the value

1. 4
2. 8
3. 9
4. 6720

ANSWERS  
Q1.A.  
In for loop, mentioning expression is optional. **>>=**is a composite operator. It shifts the binary representation of the value by 1 to the right and assigns the resulting value to the same variable. The for loop is executed until value of variable **i** doesn't drop to 0.

Q2.A.

The program goes in an infinite loop because n is never zero when loop condition (n != 0) is checked. n changes like 7 5 3 1 -1 -3 -5 -7 -9 ..

Q3.A.

The expression **i & 1** returns 1 if the rightmost bit is set and returns 0 if the rightmost bit is not set. As all odd integers have their rightmost bit set, the control goes to the block labeled ODD.

Q4.A

The control first goes to the if statement where **0** is printed. The **printf("0")** returns the number of characters being printed i.e. 1. The block under if statement gets executed and i is initialized with 3.

Q5.A.

Since i is defined globally, it is initialized with default value 0. The Else block is executed as the expression within if evaluates to FALSE. Please note that the empty block is equivalent to a semi-colon(;). So the statements **if (i);** and **if (i) {}** are equivalent.

Q6.A

The program goes in an infinite loop because n is never zero when loop condition (n != 0) is checked. n changes like 7 5 3 1 -1 -3 -5 -7 -9 ...

Q7.A

* In loop jamming, the bodies of the two loops are merged together to form a single loop provided that they do not make any references to each other.

* In loop concatenation, the bodies of the two loops are concatenated together to form a series of loop, loop concatenation, some times help to reduce complexity.

* In loop unrolling, we try to optimize a program's execution speed. It is also known as space–time tradeoff.

* In strength reduction compiler optimize expensive operations with equivalent but less expensive operations.

Q8.A

Currently there is no PrepInsta Explanation. Did you know?

You can also submit your own version of explanations under user explanation section and also view other submitted explanations by other users.

Q9.A

Currently there is no PrepInsta Explanation. Did you know?

You can also submit your own version of explanations under user explanation section and also view other submitted explanations by other users.

Q10.A  
Initially i= 6720, j=4  
  
Initially i= 6720, j=4

6720 % 4 == 0, true 6720/4=1680 5

1680 % 5 == 0, true 1680/5=336 6

336 % 6 == 0, true 336/6=56 7

56 % 7 == 0, true 56/7=8 8

8 % 8 == 0, true 8/8=1 9

1 % 9 == 0, false, condition terminated