1. #include<stdio.h>

```
main()
{
        float m=1.1;
        double x=1.1;
        if(m==x)
        printf("India");
        else
        printf("AP");
}
```

output:AP

2. **#include<stdio.h>**

```
main()
{
        int i=-1,j=-1,k=0,l=2,m;


        printf("%d%d%d%d%d",i,j,k,l,m);
} 0utput:0 0 1 3 1
```

3. **#include<stdio.h>**

```
main()
{
```

```
                printf("\nab");
                printf("\bsi");
                printf("\rha");
    }  output : hai
4. #include<stdio.h>
    #define a 10
    main()
    {
                #define c 50
                printf("%d",c);

          } 50
5. #include<stdio.h>
    main()
    {

                printf("%p",main);
          }hexadecimal numbers
6. #include<stdio.h>
    main()
    {
                int i=400,j=300;
                printf("%d..%d");
          }garbage collection
7. #include<stdio.h>
```

```
main()
{
        int i,j;

                printf("%d",scanf("%d%d",&i,&j));
} 2
```

8.
```
#include <stdio.h>
int i;
int main()
{
   if (i);
   else
      printf("Ëlse");
  return 0;
}
```
else block is executed

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

10.
```
      #include <stdio.h>
int main()
```

```
{
 int c = 5, no = 10;
    do {
        no /= c;
   } while(c--);

    printf ("%dn", no);
    return 0;
}
```

11.

```
#include <stdio.h>
int main()
{
 int i;
  for (i = 1; i != 10; i += 1)
   printf(" Computer");
 return 0;
}   infinite times
```

12.

```
#include <stdio.h>
int main()
{
    int i = 3;
    switch(i)
    {
        printf("shsdskd");
        case 1: printf("com");
```

```
            break;
        case 2: printf("puter");
            break;
        default: printf("computer");
    }
    return 0;
} computer
```

13.
```
#include <stdio.h>
int main()
{
    X check;
    switch (check)
  {
        // Some case labels
    }
    return 0;
}
```
Which of the following cannot represent *X*?

14.
```
#include <stdio.h>
int main()
{
    int check = 20, arr[] = {10, 20, 30};
    switch (check)
   {
        case arr[0]: printf("Geeks ");
        case arr[1]: printf("Quiz ");
```

```
        case arr[2]: printf("GeeksQuiz");
    }
    return 0;
} Compiletime error
```

Important points:

- 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.

```
#include <stdio.h>
int main()
{
int index;
for(index=1; index<=5; index++)
{
printf("%d", index);
if (index==3)
continue;
}
} 12345
```