Pseudocodes

1)
```
#include <iostream>
using namespace std;
int main()
{
int x=0, y=1;
for(; y; cout<<x++<<y++<<" ")
{
x=y++ <=15;
}
return 0;
}
```

Ans Infinite loop

2)

integer a, b, c, x, y, z, p, q, r;
Set x=5, y=10, z=15;
Set p=10, q=20, r=30;
a= x+p;
b= y*q;
c= r/z;
Print a b c

Any
a=5+10=15
b=10*20=200
c=30/15=2

15 200 2

3) Integer rak(Integer n);
if (n <-0)
return 0;
if(n < 1)
return 1;
Print n
rak(n-1);
Print n
End function rak

dry rak(int n) ⇒ rak(3)
↳ Print 3
↳ rak(2)
↳ Print 2
↳ rak(1)
↳ return 1
↳ Print 2
↳ Print 3

3 2 1 1 2 3

4) #include<stdio.h>
int main()
{
int x=4, y=5, z=6;
if(x+y>z)
{
if(z>y)
{
if(y>x)

```
{
    printf("%d", x);
}
}
else {
    printf("%d", y);
}
```

Ans $x+y=9$ & $z=6$.

```
if(9>6)
{
    if(z>y)
    {
        if(6>5)
        {
            if(5>4)
            {
                print(x)
            }
```

So, 4 will be printed.

5)
```
#Include<stdio.h>
int fun(int x, int y, int z)
{
    if(x>y && y>z)
    {
        return fun(y, x, z);
    }
    if(y>z && z>x)
    {
```

```
return fun(z,y,x);
}
if(z>y && z>x)
{
  return fun(x,y,z);
}
}
int main()
{
int a=10,b=15,c=20;
 a=a+b;
b=a+c;
c=b+c;
 fun(a,b,c);
}
```

a) 25 20 30 b) runtime error. c) 10 15 20
d) none of the above

The function fun will keep calling itself forever with the same arguments because there is no base case to stop recursion.
It is an runtime Error.

6) #include<stdio.h>
int fun(int i)
{
if (i%2==0) return i;
else return fun(i-1);
}
int main()
{
int a=11;
printf("%d", fun(a));
}

a) 0(zero) b) 10 c) 1 d) none of the above

The first if condition in fun(int h) will be false, as value of i is 11.

then

11%2==0 X

else

return fun(i-1) = 10

10%2==0 ✓

print (10)

Ans 10

7) #Include<stdio.h>
int fun(int i)
{
if(i==0)
return 0;
else if(i%2==0)
return fun(i-1);
else
return fun(i-1);
}
int main()
{
int a=11;
printf("%d", fun(a))
}

10

8) If LOC = -1 do ITEM NOT FOUND.
Do Something (DATA, N, ITEM, LOC)
initialize counter set LOC=0, LOW=0, HI=N-1
[Search for item]
Repeat while LOW ≤ HI
MID = (LOW+HI)/2
IF ITEM = DATA[MID] do
LOC = MID
Return LOC
IF ITEM = DATA[MID]
HI = MID-1
ELSE
LOW = MID+1

Ans The element in an array should be in the sorted form.

9) #include<stdio.h>
int main()
{
int x=12, y=10, z=13;
y=x+y>z ? printf("%d",z);

The ternary operator used in the code is not syntactically correct.

Condition? if true : if false

10) Set a=3; b=5; c=1;
a=a+b+c/2;
b=a+b/2;
if(a>b)
Print Prime Mock
else
Printf Prime Video

a+b+c/2 → 3+5+1/2 → 8+0.5 = 8
b = 8+5/2 = 10
if(8>10) ✗
Prime Video

11) main()
int num[ ] = {1,4,8,12,14};
int *a, *b;
int i;
a = num;
b = num+2;
i = *a++;
printf("%d, %d, %d\n", i, *a, *b);

a) 1, 4, 8 b) 4, 1, 8 c) 2, 1, 8 d) 1, 4, 4

When we have stored num in 'a' it means that we have stored num[0] ie 1 in a
and we have stored num+2 in 'b', we have stored num[2] = 8
Now in 'i', 1 will get stored
and then a will get incremented
Now 'a' = num+1 ie a++ so 4 will get stored in it and its previous value which is 1

i = 1
a = 4
b = 8

12)
```
#include<stdio.h>
int main()
{
int i=1,j;
for(;;)
{
i++;
if(i)
j= --i;
if (i<5)
printf("Royal Pass",i++);
else
break;
}
return 0;
}
```

a) will print Royal Pass 4 times b) No, compile error but it will run into an infinite loop printing Royal Pass

c) No, compile error but it'll not print Royal Pass.

d) compile-time error.

In every iteration the value of i is getting incremented

```
#include<iostream>
using namespace std;
int main()
{
int x=0,y=1;
for(;y;cout<<++x<<y+1<<" ")
{
x=y++<=15;
}
return 0;
}
```

Infinite loop

```
#include<stdio.h>
int main()
{
~~int i=1,j;~~ int x=4,y=0;
int z;
z=(x++ + ++y + y++, x++);
printf("%d\n",z);
return 0;
}
```

y

5

15)
```
#include<stdio.h>
int f(int n)
{
  if (n==0)
   return 1;
  else
   return f(n-1);
}
int main()
{
  printf("Result: %d", f(50));
  return 0;
}
```

The code is a basic application of recursion technique which prints a user required result, the function f(int n), will keep on calling itself until the value of n becomes 0, and then finally it will print 1.

16)
```
#include<stdio.h>
int f(int n)
{
  if(n==0)
   return 1;
  else
  return n+f(n-1);
}
int main()
{
  printf("%d", f(10));
  return 0;
}
```

a) compile time error
b) infinite loop
c) 56
d) 55

10+9+8+7+6+5+4+3+2+1+1 = 55