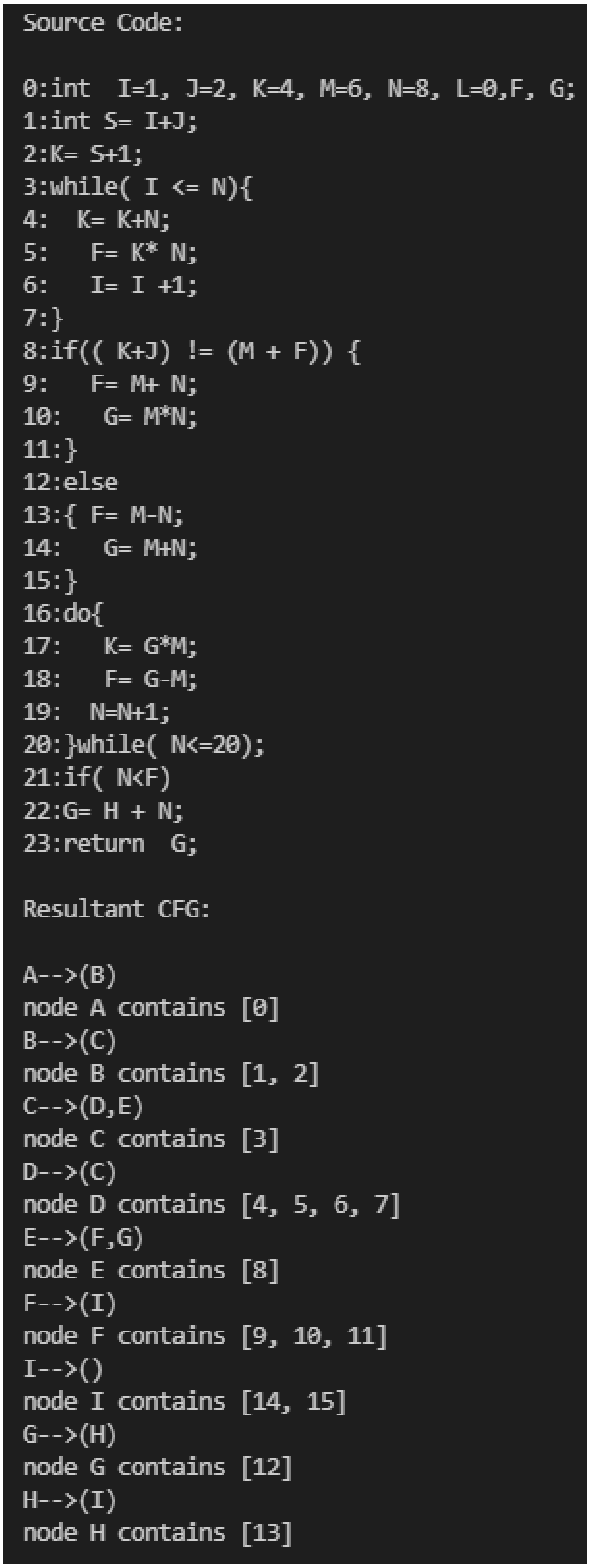
Input 1

```
int  I=1, J=2, K=4, M=6, N=8, L=0,F, G;

int S= I+J;

K= S+1;

while( I <= N){

  K= K+N;

   F= K* N;

   I= I +1;

}

if(( K+J) != (M + F)) {

   F= M+ N;

   G= M*N;

}

else

{ F= M-N;

   G= M+N;

}

do{

   K= G*M;

   F= G-M;

  N=N+1;

}while( N<=20);

if( N<F)

G= H + N;

return  G;
```

Output 1



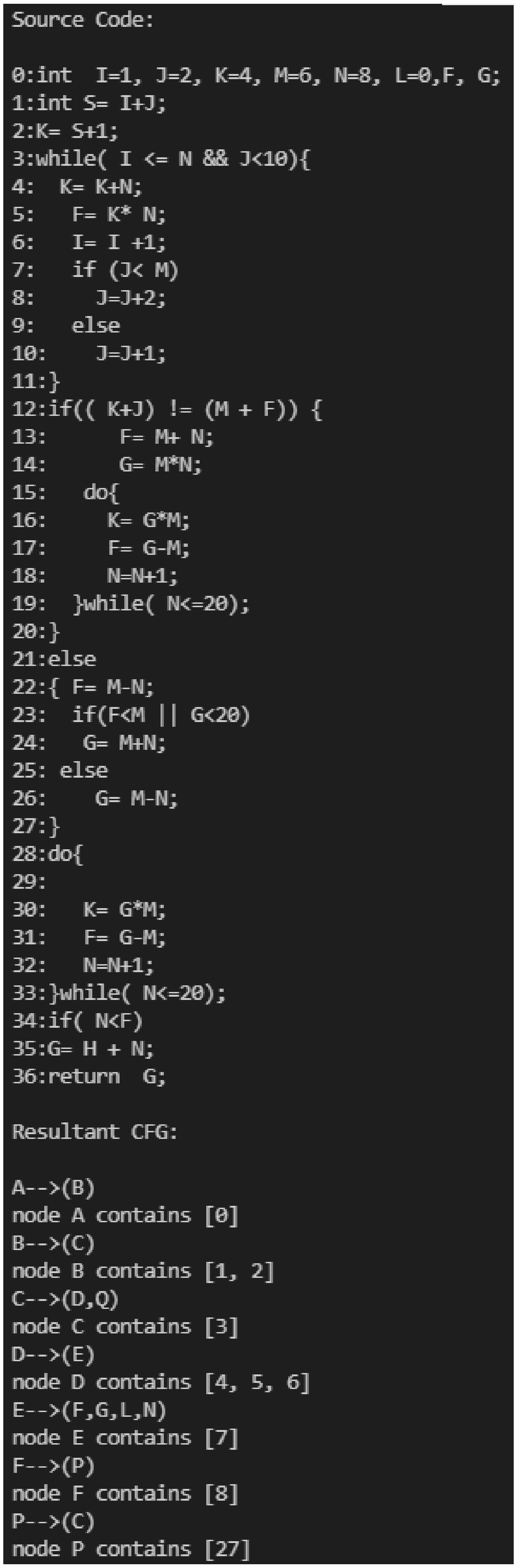
```
Source Code:

0:int  I=1, J=2, K=4, M=6, N=8, L=0,F, G;
1:int S= I+J;
2:K= S+1;
3:while( I <= N){
4:  K= K+N;
5:   F= K* N;
6:   I= I +1;
7:}
8:if(( K+J) != (M + F)) {
9:   F= M+ N;
10:   G= M*N;
11:}
12:else
13:{ F= M-N;
14:   G= M+N;
15:}
16:do{
17:   K= G*M;
18:   F= G-M;
19:  N=N+1;
20:}while( N<=20);
21:if( N<F)
22:G= H + N;
23:return  G;

Resultant CFG:

A-->(B)
node A contains [0]
B-->(C)
node B contains [1, 2]
C-->(D,E)
node C contains [3]
D-->(C)
node D contains [4, 5, 6, 7]
E-->(F,G)
node E contains [8]
F-->(I)
node F contains [9, 10, 11]
I-->()
node I contains [14, 15]
G-->(H)
node G contains [12]
H-->(I)
node H contains [13]
```

Input 2

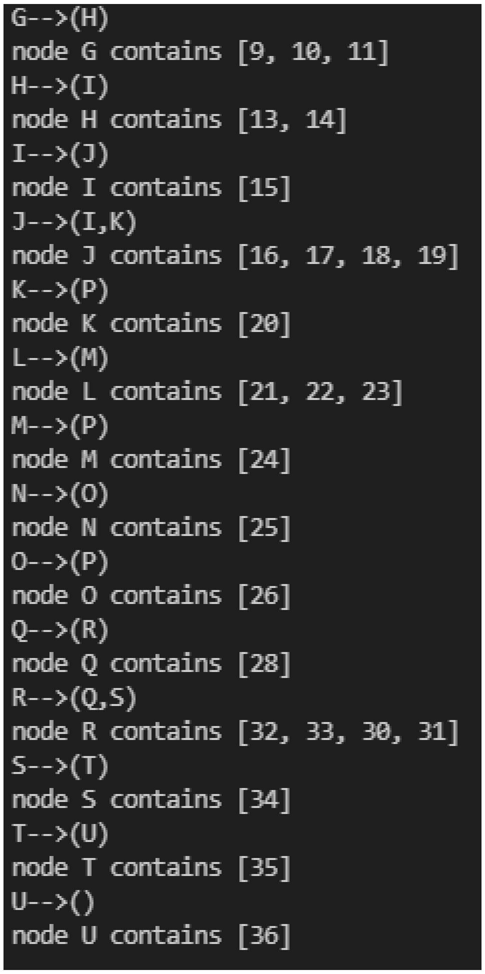
```
int  I=1, J=2, K=4, M=6, N=8, L=0,F, G;
int S= I+J;
K= S+1;
while( I <= N && J<10){
  K= K+N;
   F= K* N;
   I= I +1;
   if (J< M)
     J=J+2;
   else
    J=J+1;
}
if(( K+J) != (M + F)) {
      F= M+ N;
      G= M*N;
   do{
     K= G*M;
     F= G-M;
     N=N+1;
  }while( N<=20);
}
else
{ F= M-N;
  if(F<M || G<20)
   G= M+N;
 else
    G= M-N;
}
```

```
do{
   K= G*M;
   F= G-M;
   N=N+1;
}while( N<=20);
if( N<F)
G= H + N;
return  G;
```

Output 2



```
Source Code:

0:int  I=1, J=2, K=4, M=6, N=8, L=0,F, G;
1:int S= I+J;
2:K= S+1;
3:while( I <= N && J<10){
4:  K= K+N;
5:   F= K* N;
6:   I= I +1;
7:   if (J< M)
8:     J=J+2;
9:   else
10:    J=J+1;
11:}
12:if(( K+J) != (M + F)) {
13:      F= M+ N;
14:      G= M*N;
15:   do{
16:     K= G*M;
17:     F= G-M;
18:     N=N+1;
19:  }while( N<=20);
20:}
21:else
22:{ F= M-N;
23:  if(F<M || G<20)
24:   G= M+N;
25: else
26:    G= M-N;
27:}
28:do{
29:
30:   K= G*M;
31:   F= G-M;
32:   N=N+1;
33:}while( N<=20);
34:if( N<F)
35:G= H + N;
36:return  G;

Resultant CFG:

A-->(B)
node A contains [0]
B-->(C)
node B contains [1, 2]
C-->(D,Q)
node C contains [3]
D-->(E)
node D contains [4, 5, 6]
E-->(F,G,L,N)
node E contains [7]
F-->(P)
node F contains [8]
P-->(C)
node P contains [27]
```

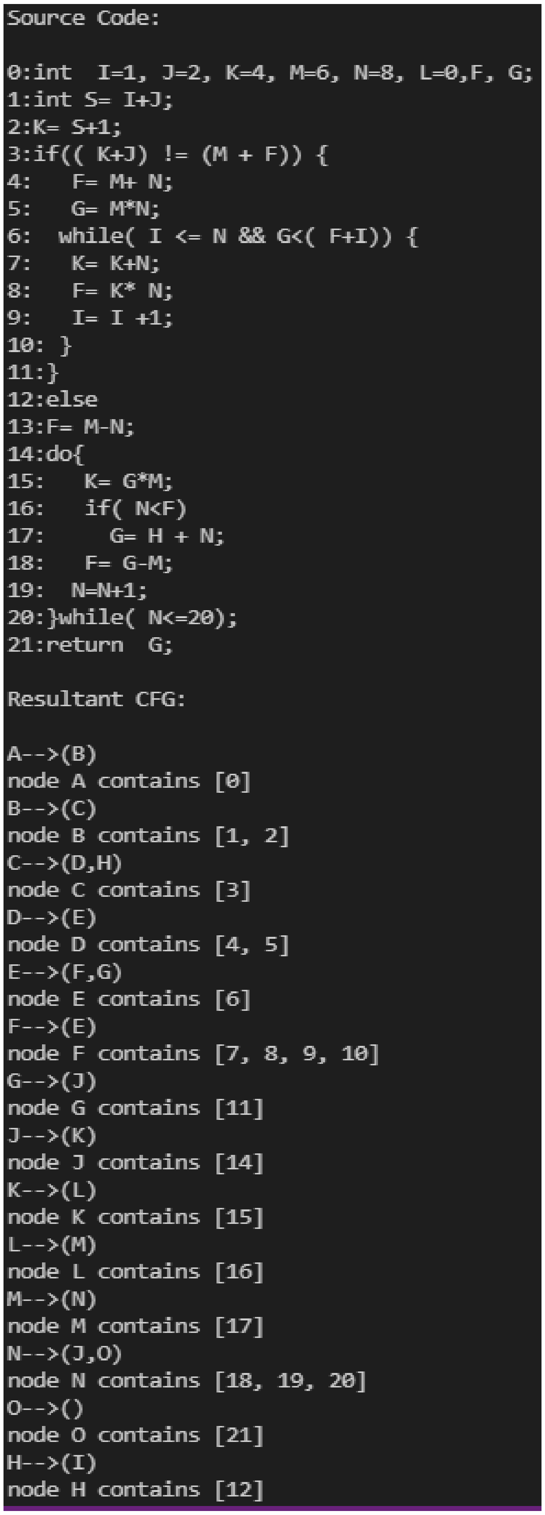


```
G-->(H)
node G contains [9, 10, 11]
H-->(I)
node H contains [13, 14]
I-->(J)
node I contains [15]
J-->(I,K)
node J contains [16, 17, 18, 19]
K-->(P)
node K contains [20]
L-->(M)
node L contains [21, 22, 23]
M-->(P)
node M contains [24]
N-->(O)
node N contains [25]
O-->(P)
node O contains [26]
Q-->(R)
node Q contains [28]
R-->(Q,S)
node R contains [32, 33, 30, 31]
S-->(T)
node S contains [34]
T-->(U)
node T contains [35]
U-->()
node U contains [36]
```

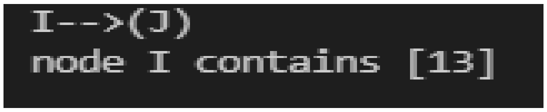
Input 3

```
int  I=1, J=2, K=4, M=6, N=8, L=0,F, G;

int S= I+J;

K= S+1;

if(( K+J) != (M + F)) {

   F= M+ N;

   G= M*N;

  while( I <= N && G<( F+I)) {

   K= K+N;

   F= K* N;

   I= I +1;

 }

}

else

F= M-N;

do{

   K= G*M;

   if( N<F)

     G= H + N;

   F= G-M;

  N=N+1;

}while( N<=20);

return  G;
```

Output 3



```
Source Code:

0:int  I=1, J=2, K=4, M=6, N=8, L=0,F, G;
1:int S= I+J;
2:K= S+1;
3:if(( K+J) != (M + F)) {
4:   F= M+ N;
5:   G= M*N;
6:  while( I <= N && G<( F+I)) {
7:   K= K+N;
8:   F= K* N;
9:   I= I +1;
10: }
11:}
12:else
13:F= M-N;
14:do{
15:   K= G*M;
16:   if( N<F)
17:     G= H + N;
18:   F= G-M;
19:  N=N+1;
20:}while( N<=20);
21:return  G;

Resultant CFG:

A-->(B)
node A contains [0]
B-->(C)
node B contains [1, 2]
C-->(D,H)
node C contains [3]
D-->(E)
node D contains [4, 5]
E-->(F,G)
node E contains [6]
F-->(E)
node F contains [7, 8, 9, 10]
G-->(J)
node G contains [11]
J-->(K)
node J contains [14]
K-->(L)
node K contains [15]
L-->(M)
node L contains [16]
M-->(N)
node M contains [17]
N-->(J,O)
node N contains [18, 19, 20]
O-->()
node O contains [21]
H-->(I)
node H contains [12]
```

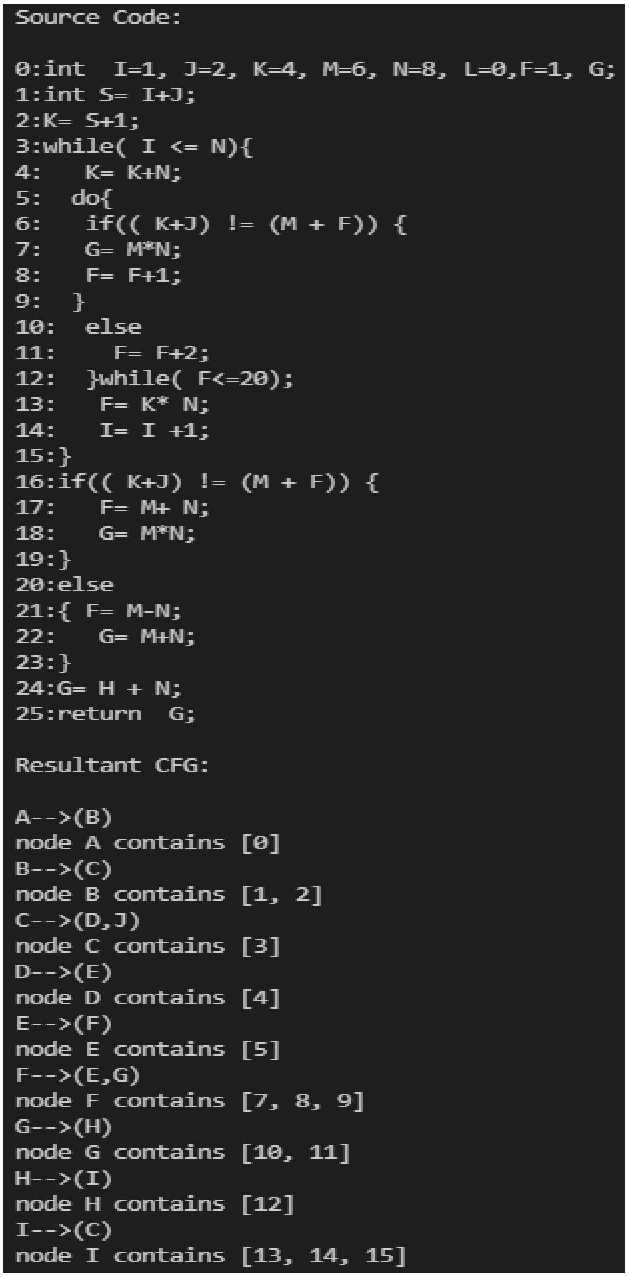


```
I-->(J)
node I contains [13]
```

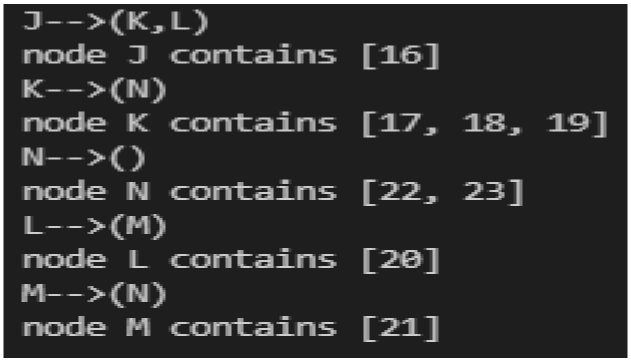
Input 4

```
int  I=1, J=2, K=4, M=6, N=8, L=0,F=1, G;
int S= I+J;
K= S+1;
while( I <= N){
   K= K+N;
  do{
   if(( K+J) != (M + F)) {
   G= M*N;
   F= F+1;
  }
  else
    F= F+2;
  }while( F<=20);
   F= K* N;
   I= I +1;
}
if(( K+J) != (M + F)) {
   F= M+ N;
   G= M*N;
}
else
{ F= M-N;
   G= M+N;
}
G= H + N;
return  G;
```

Output 4



```
Source Code:

0:int  I=1, J=2, K=4, M=6, N=8, L=0,F=1, G;
1:int S= I+J;
2:K= S+1;
3:while( I <= N){
4:   K= K+N;
5:  do{
6:   if(( K+J) != (M + F)) {
7:   G= M*N;
8:   F= F+1;
9:  }
10:  else
11:    F= F+2;
12:  }while( F<=20);
13:   F= K* N;
14:   I= I +1;
15:}
16:if(( K+J) != (M + F)) {
17:   F= M+ N;
18:   G= M*N;
19:}
20:else
21:{ F= M-N;
22:   G= M+N;
23:}
24:G= H + N;
25:return  G;

Resultant CFG:

A-->(B)
node A contains [0]
B-->(C)
node B contains [1, 2]
C-->(D,J)
node C contains [3]
D-->(E)
node D contains [4]
E-->(F)
node E contains [5]
F-->(E,G)
node F contains [7, 8, 9]
G-->(H)
node G contains [10, 11]
H-->(I)
node H contains [12]
I-->(C)
node I contains [13, 14, 15]
```



```
J-->(K,L)
node J contains [16]
K-->(N)
node K contains [17, 18, 19]
N-->()
node N contains [22, 23]
L-->(M)
node L contains [20]
M-->(N)
node M contains [21]
```