1. Ниже на пяти языках программирования записаны две рекурсивные функции: *F* и *G*.

Бейсик

```
DECLARE FUNCTION F(n)
DECLARE FUNCTION G(n)
FUNCTION F(n)
  IF n > 2 THEN
    F = F(n - 1) + G(n-2)
  ELSE
    F = 1
  END IF
END FUNCTION
FUNCTION G(n)
  IF n > 2 THEN
    G = G(n - 1) + F(n-2)
  ELSE
    G = 1
  END IF
END FUNCTION
```

Python

```
def F(n):
    if n > 2:
        return F(n-1)+ G(n-2)
    else: return 1
def G(n):
    if n > 2:
        return G(n-1) + F(n-2)
    else: return 1
```

Паскаль

```
function F(n: integer): integer;
begin
  if n > 2 then
    F := F(n - 1) + G(n - 2)
  else
    F := 1;
end;
function G(n: integer): integer;
begin
  if n > 2 then
    G := G(n - 1) + F(n - 2)
  else
    G := 1;
end;
```

Алгоритмический язык

```
алг цел F(цел n)
нач
  если n > 2
    то
      знач := F(n - 1) + G(n - 2)
    иначе
      знач := 1
  все
кон
алг цел G(цел n)
нач
  если n > 2
    то
      знач := G(n - 1) + F(n - 2)
    иначе
      знач := 1
  все
кон
```

Си

```
int F(int n)
{
  if (n > 2)
    return F(n-1) + G(n-2);
  else return 1;
}
int G(int n)
{
  if (n > 2)
    return G(n-1) + F(n-2);
  else return 1;
}
```

Чему будет равно значение, вычисленное при выполнении вызова *F*(7)?

2. Ниже на пяти языках программирования записаны две рекурсивные функции: *F* и *G*.

Бейсик

```
DECLARE FUNCTION F(n)
DECLARE FUNCTION G(n)
FUNCTION F(n)
  IF n > 2 THEN
    F = F(n - 1) + G(n-2)
  ELSE
    F = 1
  END IF
END FUNCTION
FUNCTION G(n)
  IF n > 2 THEN
    G = G(n - 1) + F(n-2)
  ELSE
    G = 1
  END IF
END FUNCTION
```

Python

```
def F(n):
    if n > 2:
        return F(n-1)+ G(n-2)
    else: return 1
def G(n):
    if n > 2:
        return G(n-1) + F(n-2)
    else: return 1
```

Паскаль

```
function F(n: integer): integer;
begin
  if n > 2 then
    F := F(n - 1) + G(n - 2)
  else
    F := 1;
end;
function G(n: integer): integer;
begin
  if n > 2 then
    G := G(n - 1) + F(n - 2)
  else
    G := 1;
end;
```

Алгоритмический язык

```
алг цел F(цел n)
нач
  если n > 2
    то
      знач := F(n - 1) + G(n - 2)
    иначе
      знач := 1
  все
кон
алг цел G(цел n)
нач
  если n > 2
    то
      знач := G(n - 1) + F(n - 2)
    иначе
      знач := 1
  все
кон
```

Си

```
int F(int n)
{
  if (n > 2)
    return F(n-1) + G(n-2);
  else return 1;
}
int G(int n)
{
  if (n > 2)
    return G(n-1) + F(n-2);
  else return 1;
}
```

Чему будет равно значение, вычисленное при выполнении вызова *F*(8)?

3. Ниже на пяти языках программирования записаны две рекурсивные функции: *F* и *G*.

Бейсик

```
FUNCTION F(n)
  IF n > 2 THEN
    F = F(n - 1) + G(n - 2)
  ELSE
    F = n
  END IF
END FUNCTION
FUNCTION G(n)
  IF n > 2 THEN
    G = G(n - 1) + F(n - 2)
  ELSE
    G = n + 1
  END IF
END FUNCTION
```

Python

```
def F(n):
  if n > 2:
    return F(n-1) + G(n-2)
  else: return n
def G(n):
  if n > 2:
    return G(n-1) + F(n-2)
  else: return n+1
```

Паскаль

```
function F(n: integer): integer;
begin
  if n > 2 then
    F := F(n - 1) + G(n - 2)
  else
    F := n;
end;
function G(n: integer): integer;
begin
  if n > 2 then
    G := G(n - 1) + F(n - 2)
  else
    G := n+1;
end;
```

Алгоритмический язык

```
алг цел F(цел n)
нач
  если n > 2
    то
      знач := F(n - 1)+G(n - 2)
    иначе
      знач := n
  все
кон
алг цел G(цел n)
нач
  если n > 2
    то
      знач := G(n - 1)+F(n - 2)
    иначе
      знач := n+1
  все
кон
```

Си

```
int F(int n)
{
  if (n > 2)
    return F(n-1) + G(n-2);
  else return n;
}
int G(int n)
{
  if (n > 2)
    return G(n-1) + F(n-2);
  else return n + 1;
}
```

Чему будет равно значение, вычисленное при выполнении вызова *F*(6)?

4. Ниже на пяти языках программирования записаны две рекурсивные функции: *F* и *G*.

Бейсик

```
FUNCTION F(n)
  IF n > 2 THEN
    F = F(n - 1) + G(n - 2)
  ELSE
    F = n
  END IF
END FUNCTION
FUNCTION G(n)
  IF n > 2 THEN
    G = G(n - 1) + F(n - 2)
  ELSE
    G = n + 1
  END IF
END FUNCTION
```

Python

```
def F(n):
  if n > 2:
    return F(n-1) + G(n-2)
  else: return n
def G(n):
  if n > 2:
    return G(n-1) + F(n-2)
  else: return n+1
```

Паскаль

```
function    G(n:integer):    integer;
forward;

function F(n: integer): integer;
begin
  if n > 2 then
    F := F(n - 1) + G(n - 2)
  else
    F := n;
end;
function G(n: integer): integer;
begin
  if n > 2 then
    G := G(n - 1) + F(n - 2)
  else
    G := n+1;
end;
```

Алгоритмический язык

```
алг цел F(цел n)
нач
  если n > 2
    то
      знач := F(n - 1)+G(n - 2)
    иначе
      знач := n
  все
кон
алг цел G(цел n)
нач
  если n > 2
    то
      знач := G(n - 1)+F(n - 2)
    иначе
      знач := n+1
  все
кон
```

Си

```
int F(int n)
{
  if (n > 2)
    return F(n-1) + G(n-2);
  else return n;
}
int G(int n)
{
  if (n > 2)
    return G(n-1) + F(n-2);
  else return n + 1;
}
```

Чему будет равно значение, вычисленное при выполнении вызова *G*(6)?

5. Ниже на пяти языках программирования записаны две рекурсивные функции: *F* и *G*.

Бейсик

```
FUNCTION F(n)
  IF n > 2 THEN
    F = F(n-1)+G(n-1)+F(n-2)
  ELSE
    F = n
  END IF
END FUNCTION

FUNCTION G(n)
  IF n > 2 THEN
    G = G(n-1)+F(n-1)+G(n-2)
  ELSE
    G = n+1
  END IF
END FUNCTION
```

Python

```
def F(n):
  if n > 2:
    return F(n-1)+G(n-1)+F(n-2)
  else: return n

def G(n):
  if n > 2:
    return G(n-1)+F(n-1)+G(n-2)
  else: return n+1
```

Паскаль

```
function F(n: integer):
integer;
begin
  if n > 2 then
    F := F(n-1)+G(n-1)+F(n-2)
  else
    F := n;
end;

function G(n: integer):
integer;
begin
  if n > 2 then
    G := G(n-1)+F(n-1)+G(n-2)
  else
    G := n+1;
end;
```

Алгоритмический язык

```
алг цел F(цел n)
нач
  если n > 2
    то
      знач := F(n-1)+G(n-1)+F(n-2)
    иначе
      знач := n
  все
кон

алг цел G(цел n)
нач
  если n > 2
  то
    знач := G(n-1)+F(n-1)+G(n-2)
  иначе
    знач := n+1
  все
кон
```

Си

```
int F(int n) {
  if (n > 2)
    return F(n-1)+G(n-1)+F(n-2);
  else return n;
}

int G(int n){
  if (n > 2)
    return G(n-1)+F(n-1)+G(n-2);
  else return n+1;
}
```

Чему будет равно значение, вычисленное при выполнении вызова *G*(5)?

6. Ниже на пяти языках программирования записаны две рекурсивные функции: *F* и *G*.

Бейсик

```
FUNCTION F(n)
  IF n > 2 THEN
    F = F(n-1)+G(n-1)+F(n-2)
  ELSE
    F = n
  END IF
END FUNCTION

FUNCTION G(n)
  IF n > 2 THEN
    G = G(n-1)+F(n-1)+G(n-2)
  ELSE
    G = n+1
  END IF
END FUNCTION
```

Python

```
def F(n):
  if n > 2:
    return F(n-1)+G(n-1)+F(n-2)
  else: return n

def G(n):
  if n > 2:
    return G(n-1)+F(n-1)+G(n-2)
  else: return n+1
```

Паскаль

```
function F(n: integer):
integer;
begin
  if n > 2 then
    F := F(n-1)+G(n-1)+F(n-2)
  else
    F := n;
end;

function G(n: integer):
integer;
begin
  if n > 2 then
    G := G(n-1)+F(n-1)+G(n-2)
  else
    G := n+1;
end;
```

Алгоритмический язык

```
алг цел F(цел n)
нач
  если n > 2
    то
      знач := F(n-1)+G(n-1)+F(n-2)
    иначе
      знач := n
  все
кон

алг цел G(цел n)
нач
  если n > 2
  то
    знач := G(n-1)+F(n-1)+G(n-2)
  иначе
    знач := n+1
  все
кон
```

Си

```
int F(int n) {
  if (n > 2)
    return F(n-1)+G(n-1)+F(n-2);
  else return n;
}

int G(int n){
  if (n > 2)
    return G(n-1)+F(n-1)+G(n-2);
  else return n+1;
}
```

Чему будет равно значение, вычисленное при выполнении вызова *F*(5)?

7. Ниже на пяти языках программирования записаны рекурсивные функции F и G.

Бейсик

```
FUNCTION F(n)
  IF n > 2 THEN
    F = F(n-1)+G(n-1)+F(n-2)
  ELSE
    F = n
  END IF
END FUNCTION
FUNCTION G(n)
  IF n > 2 THEN
    G = G(n-1)+F(n-1)+G(n-2)
  ELSE
    G = 3-n
  END IF
END FUNCTION
```

Python

```
def F(n):
    if n > 2:
        return F(n-1)+G(n-1)+F(n-2)
    else: return n
def G(n):
    if n > 2:
        return G(n-1)+F(n-1)+G(n-2)
    else: return 3-n
```

Алгоритмический язык

```
алг цел F(цел n)
нач
  если n > 2
    то
      знач := F(n-1)+G(n-1)+F(n-2)
    иначе
      знач := n
    все
кон
алг цел G(цел n)
нач
  если n > 2
    то
      знач := G(n-1)+F(n-1)+G(n-2)
    иначе
      знач := 3-n
  все
кон
```

Паскаль

```
function F(n: integer): integer;
begin
  if n > 2 then
    F := F(n-1)+G(n-1)+F(n-2)
  else
    F := n;
end;
function G(n: integer): integer;
begin
  if n > 2 then
    G := G(n-1)+F(n-1)+G(n-2)
  else
    G := 3-n;
end;
```

Си

```
int F(int n){
if (n > 2)
return F(n-1)+G(n-1)+F(n-2);
else return n;
}
int G(int n){
if (n > 2)
return G(n-1)+F(n-1)+G(n-2);
else return 3-n;
}
```

Чему будет равно значение, вычисленное при выполнении вызова G(5)?

8. Ниже на пяти языках программирования записаны рекурсивные функции F и G.

Бейсик

```
FUNCTION F(n)
  IF n > 2 THEN
    F = F(n-1)+G(n-1)+F(n-2)
  ELSE
    F = n
  END IF
END FUNCTION
FUNCTION G(n)
  IF n > 2 THEN
    G = G(n-1)+F(n-1)+G(n-2)
  ELSE
    G = 3-n
  END IF
END FUNCTION
```

Python

```
def F(n):
    if n > 2:
        return F(n-1)+G(n-1)+F(n-2)
    else: return n
def G(n):
    if n > 2:
        return G(n-1)+F(n-1)+G(n-2)
    else: return 3-n
```

Алгоритмический язык

```
алг цел F(цел n)
нач
  если n > 2
    то
      знач := F(n-1)+G(n-1)+F(n-2)
    иначе
      знач := n
    все
кон
алг цел G(цел n)
нач
  если n > 2
    то
      знач := G(n-1)+F(n-1)+G(n-2)
    иначе
      знач := 3-n
  все
кон
```

Паскаль

```
function F(n: integer): integer;
begin
  if n > 2 then
    F := F(n-1)+G(n-1)+F(n-2)
  else
    F := n;
end;
function G(n: integer): integer;
begin
  if n > 2 then
    G := G(n-1)+F(n-1)+G(n-2)
  else
    G := 3-n;
end;
```

Си

```
int F(int n){
if (n > 2)
return F(n-1)+G(n-1)+F(n-2);
else return n;
}
int G(int n){
if (n > 2)
return G(n-1)+F(n-1)+G(n-2);
else return 3-n;
}
```

Чему будет равно значение, вычисленное при выполнении вызова F(5)?

9. Ниже записаны две рекурсивные функции, F и G:

```
function F(n: integer): integer;
 begin
  if (n > 2) then F := F(n - 1) + G(n - 1) + F(n-2)
 else
    F := n;
 end;
function G(n: integer): integer;
 begin
  if (n > 2) then G := G(n - 1) + F(n - 1) + G(n-2)
 else
    G := n;
 end;
```

Чему будет равно значение, вычисленное при выполнении вызова F(5)?

10. Ниже на пяти языках программирования записаны две рекурсивные функции: F и G.

Бейсик

```
FUNCTION F(n)
 IF n > 2 THEN
   F = F(n - 1) +G(n - 2)
 ELSE
    F = 2
 END IF
END FUNCTION

FUNCTION G(n)
 IF n > 2 THEN
   G = G(n - 1) +F(n - 2)
 ELSE
    G = 2
 END IF
END FUNCTION
```

Python

```
def F(n):
    if n > 2:
      return F(n-1) + G(n-2)
    else: return 2
def G(n):
    if n > 2:
      return G(n-1) + F(n-2)
    else: return 2
```

Паскаль

```
function F(n : integer): integer;
 begin
  if n > 2 then
   F := F(n - 1) + G(n - 2)
  else
   F := 2;
 end;
function G(n : integer): integer;
 begin
  if n > 2 then
   G := G(n - 1) + F(n - 2)
  else
   G := 2;
 end;
```

Алгоритмический язык

```
алг цел F(цел n)
 нач
  если n > 2
  то
    знач:= F(n-1) + G(n-2)
  иначе
    знач:=2
  все
 кон
алг цел G(цел n)
 нач
  если n > 2
  то
    знач:= G(n-1) + F(n-2)
  иначе
    знач:=2
  все
 кон
```

Си

```
int F(int n) {
    if (n > 2)
     return F(n-1) + G(n-2);
    else
     return 2;
}
int G(int n) {
    if (n > 2)
     return G(n-1) + F(n-2);
    else
     return 2;
}
```

Чему будет равно значение, вычисленное при выполнении вызова F(6)?

11. Ниже на пяти языках программирования записаны две рекурсивные функции: F и G.

Бейсик

```
FUNCTION F(n)
 IF n > 2 THEN
   F = F(n - 1) +G(n - 2)
 ELSE
    F = 2
 END IF
END FUNCTION

FUNCTION G(n)
 IF n > 2 THEN
   G = G(n - 1) +F(n - 2)
 ELSE
    G = 2
 END IF
END FUNCTION
```

Python

```
def F(n):
    if n > 2:
      return F(n-1) + G(n-2)
    else: return 2
def G(n):
    if n > 2:
      return G(n-1) + F(n-2)
    else: return 2
```

Паскаль

```
function F(n : integer): integer;
 begin
  if n > 2 then
   F := F(n - 1) + G(n - 2)
  else
   F := 2;
 end;
function G(n : integer): integer;
 begin
  if n > 2 then
   G := G(n - 1) + F(n - 2)
  else
   G := 2;
 end;
```

Алгоритмический язык

```
алг цел F(цел n)
 нач
  если n > 2
  то
    знач:= F(n-1) + G(n-2)
  иначе
    знач:=2
  все
 кон
алг цел G(цел n)
 нач
  если n > 2
  то
    знач:= G(n-1) + F(n-2)
  иначе
    знач:=2
  все
 кон
```

Си

```
int F(int n) {
    if (n > 2)
     return F(n-1) + G(n-2);
    else
     return 2;
}
int G(int n) {
    if (n > 2)
     return G(n-1) + F(n-2);
    else
     return 2;
}
```

Чему будет равно значение, вычисленное при выполнении вызова G(6)?

12. Ниже на пяти языках программирования записаны две рекурсивные функции: F и G.

Бейсик

```
FUNCTION F(n)
 IF n > 1 THEN
   F = F(n - 1) +G(n - 1)
 ELSE
    F = n
 END IF
END FUNCTION

FUNCTION G(n)
 IF n > 1 THEN
   G = G(n - 1) +F(n)
 ELSE
    G = n
 END IF
END FUNCTION
```

Python

```
def F(n):
    if n > 1:
      return F(n-1) + G(n-1)
    else: return n
def G(n):
    if n > 1:
      return G(n-1) + F(n)
    else: return n
```

Паскаль

```
function F (n : integer) : integer;
 begin
  if n > 1 then
   F := F(n - 1) + G(n - 1)
  else
   F := n;
 end;
function G (n : integer) : integer;
 begin
  if n > 1 then
   G := G(n - 1) + F(n)
  else
   G := n;
 end;
```

Алгоритмический язык

```
алг цел F(цел n)
 нач
  если n > 1
  то
    знач:= F(n-1) + G(n-1)
  иначе
    знач:=n
  все
 кон
алг цел G(цел n)
 нач
  если n > 1
  то
    знач:= G(n-1) + F(n)
  иначе
    знач:=n
  все
 кон
```

Си

```
int F(int n) {
    if (n > 1)
     return F(n-1) + G(n-1);
    else
     return n;
}
int G(int n) {
    if (n > 1)
     return G(n-1) + F(n);
    else
     return n;
}
```

Чему будет равно значение, вычисленное при выполнении вызова F(5)?

13. Ниже на пяти языках программирования записаны две рекурсивные функции: F и G.

Бейсик

```
FUNCTION F(n)
 IF n > 1 THEN
   F = F(n - 1) +G(n - 1)
 ELSE
    F = n
 END IF
END FUNCTION

FUNCTION G(n)
 IF n > 1 THEN
   G = G(n - 1) +F(n)
 ELSE
    G = n
 END IF
END FUNCTION
```

Python

```
def F(n):
    if n > 1:
      return F(n-1) + G(n-1)
    else: return n
def G(n):
    if n > 1:
      return G(n-1) + F(n)
    else: return n
```

Паскаль

```
function F (n : integer) : integer;
 begin
  if n > 1 then
   F := F(n - 1) + G(n - 1)
  else
   F := n;
 end;
function G (n : integer) : integer;
 begin
  if n > 1 then
   G := G(n - 1) + F(n)
  else
   G := n;
 end;
```

Алгоритмический язык

```
алг цел F(цел n)
 нач
  если n > 1
  то
    знач:= F(n-1) + G(n-1)
  иначе
    знач:=n
  все
 кон
алг цел G(цел n)
 нач
  если n > 1
  то
    знач:= G(n-1) + F(n)
  иначе
    знач:=n
  все
 кон
```

Си

```
int F(int n) {
    if (n > 1)
     return F(n-1) + G(n-1);
    else
     return n;
}
int G(int n) {
    if (n > 1)
     return G(n-1) + F(n);
    else
     return n;
}
```

Чему будет равно значение, вычисленное при выполнении вызова G(5)?

14. Ниже на пяти языках программирования записаны две рекурсивные функции: F и G.

Бейсик

```
FUNCTION F(n)
  IF n > 2 THEN
    F = F(n-1) +G (n-2)
  ELSE
    F = n
  END IF
END FUNCTION

   FUNCTION G(n)
  IF n > 2 THEN
    G = G(n-1) + F(n-2)
  ELSE
    G = 3-n
  END IF
END FUNCTION
```

Python

```
def F(n):
    if n > 2:
        return F(n-1) + G(n-2)
    else: return n
def G(n):
    if n > 2:
        return G(n-1) + F(n-2)
    else: return 3-n
```

Паскаль

```
function F(n: integer): integer;
begin
  if n > 2 then
      F := F(n-1) + G(n-2)
  else
      F := n;
end;

function G(n: integer): integer;
begin
  if n > 2 then
    G := G(n-1) + F(n-2)
  else
      G := 3-n;
end;
```

Алгоритмический язык

```
алг цел F(цел n)
нач
  если n > 2
    то
      знач := F(n-1) + G(n-2)
    иначе
      знач := n
  все
кон

алг цел G(цел n)
нач
  если n > 2
    то
      знач := G(n-1) + F(n-2)
    иначе
      знач := 3-n
  все
кон
```

Си

```
int F(int n) {
    if (n > 2)
        return F(n-1) + G(n-2);
    else return n;
}

int G(int n) {
    if (n > 2)
        return G(n-1) + F(n-2);
    else return 3-n;
}
```

Чему будет равно значение, вычисленное при выполнении вызова G(6)?

15. Ниже на пяти языках программирования записаны две рекурсивные функции: F и G.

Бейсик

```
FUNCTION F(n)
  IF n > 2 THEN
    F = F(n-1) +G (n-2)
  ELSE
    F = n
  END IF
END FUNCTION

FUNCTION G(n)
  IF n > 2 THEN
    G = G(n-1) + F(n-2)
  ELSE
    G = 3-n
  END IF
END FUNCTION
```

Python

```
def F(n):
  if n > 2:
    return F(n-1) + G(n-2)
  else: return n

def G(n):
  if n > 2:
    return G(n-1) + F(n-2)
  else: return 3-n
```

Паскаль

```
function   F(n:   integer):   integer;
forward;
function   G(n:   integer):   integer;
forward;
function F(n: integer): integer;
begin
if n>2 then
F:=F(n-1)+G(n-2)
else
F:=n;
end;

function G(n:integer):integer;
begin
if n>2 then
G:=G(n-1)+F(n-2)
else
G:=3-n;
end;
```

Алгоритмический язык

```
алг цел F(цел n)
нач
  если n > 2
    то
      знач := F(n-1) + G(n-2)
    иначе
      знач := n
  все
кон

алг цел G(цел n)
нач
  если n > 2
    то
      знач := G(n-1) + F(n-2)
    иначе
      знач := 3-n
  все
кон
```

Си

```
int F(int n) {
  if (n > 2)
    return F(n-1) + G(n-2);
  else return n;
}

int G(int n) {
  if (n > 2)
    return G(n-1) + F(n-2);
  else return 3-n;
}
```

Чему будет равно значение, вычисленное при выполнении вызова F(6)?

16. Ниже на пяти языках программирования записаны две рекурсивные функции: F и G.

Бейсик

```
FUNCTION F(n)
    IF n > 2 THEN
        F = F(n - 1) + G(n - 2)
    ELSE
        F = n
    END IF
END FUNCTION

FUNCTION G(n)
    IF n > 2 THEN
        G = G(n - 1) + F(n - 2)
    ELSE
        G = n+1
    END IF
END FUNCTION
```

Python

```
def F(n):
    if n > 2:
        return F(n - 1)+ G(n - 2)
    else: return n

def G(n):
    if n > 2:
        return G(n - 1)+ F(n - 2)
    else: return n+1
```

Паскаль

```
function F(n: integer): integer;
begin
    if n > 2 then
        F := F(n - 1) + G(n - 2)
    else
        F := n;
end;

function G(n: integer): integer;
begin
    if n > 2 then
        G := G(n - 1) + F(n - 2)
    else
        G := n+1;
end;
```

Алгоритмический язык

```
алг цел F(цел n)
нач
    если n > 2
        то
            знач := F(n - 1)+G(n - 2)
        иначе
            знач := n
    все
кон

алг цел G(цел n)
нач
    если n > 2
        то
            знач := G(n - 1)+F(n - 2)
        иначе
            знач := n+1
    все
кон
```

Си

```
int F(int n)
{
if (n > 2)
return F(n - 1) + G(n - 2);
else return n;
}
int G(int n)
{
if (n > 2)
return G(n - 1) + F(n -2);
else return n+1;
}
```

Чему будет равно значение, вычисленное при выполнении вызова F(6)?

17. Ниже на пяти языках программирования записаны две рекурсивные функции: F и G.

Бейсик

```
FUNCTION F(n)
    IF n > 2 THEN
        F = F(n - 1) + G(n - 2)
    ELSE
        F = n
    END IF
END FUNCTION

FUNCTION G(n)
    IF n > 2 THEN
        G = G(n - 1) + F(n - 2)
    ELSE
        G = n+1
    END IF
END FUNCTION
```

Python

```
def F(n):
    if n > 2:
        return F(n - 1)+ G(n - 2)
    else: return n

def G(n):
    if n > 2:
        return G(n - 1)+ F(n - 2)
    else: return n+1
```

Паскаль

```
function F(n: integer): integer;
begin
    if n > 2 then
        F := F(n - 1) + G(n - 2)
    else
        F := n;
end;

function G(n: integer): integer;
begin
    if n > 2 then
        G := G(n - 1) + F(n - 2)
    else
        G := n+1;
end;
```

Алгоритмический язык

```
алг цел F(цел n)
нач
    если n > 2
        то
            знач := F(n - 1)+G(n - 2)
        иначе
            знач := n
    все
кон

алг цел G(цел n)
нач
    если n > 2
        то
            знач := G(n - 1)+F(n - 2)
        иначе
            знач := n+1
    все
кон
```

Си

```
int F(int n)
{
if (n > 2)
return F(n - 1) + G(n - 2);
else return n;
}
int G(int n)
{
if (n > 2)
return G(n - 1) + F(n -2);
else return n+1;
}
```

Чему будет равно значение, вычисленное при выполнении вызова G(6)?

18. Ниже на пяти языках программирования записаны рекурсивные функции F и G.

Бейсик

```
FUNCTION F(n)
    IF n > 2 THEN
        F = F(n - 1) + G(n - 2)
    ELSE
        F = n+1
    END IF
END FUNCTION

FUNCTION G(n)
    IF n > 2 THEN
        G = G(n - 1) + F(n - 2)
    ELSE
        G = n
    END IF
END FUNCTION
```

Python

```
def F(n):
    if n > 2:
        return F(n - 1)+ G(n - 2)
    else: return n+1

def G(n):
    if n > 2:
        return G(n - 1)+ F(n - 2)
    else: return n
```

Паскаль

```
function F(n: integer): integer;
begin
    if n > 2 then
        F := F(n - 1) + G(n - 2)
    else
        F := n+1;
end;

function G(n: integer): integer;
begin
    if n > 2 then
        G := G(n - 1) + F(n - 2)
    else
        G := n;
end;
```

Алгоритмический язык

```
алг цел F(цел n)
нач
    если n > 2
        то
            знач := F(n - 1)+G(n - 2)
        иначе
            знач := n+1
    все
кон

алг цел G(цел n)
нач
    если n > 2
        то
            знач := G(n - 1)+F(n - 2)
        иначе
            знач := n
    все
кон
```

Си

```
int F(int n)
{
if (n > 2)
return F(n - 1) + G(n - 2);
else return n+1;
}
int G(int n)
{
if (n > 2)
return G(n - 1) + F(n -2);
else return n;
}
```

Чему будет равно значение, вычисленное при выполнении вызова F(7)?

19. Ниже на пяти языках программирования записаны рекурсивные функции F и G.

Бейсик

```
FUNCTION F(n)
    IF n > 2 THEN
        F = F(n - 1) + G(n - 2)
    ELSE
        F = n+1
    END IF
END FUNCTION

FUNCTION G(n)
    IF n > 2 THEN
        G = G(n - 1) + F(n - 2)
    ELSE
        G = n
    END IF
END FUNCTION
```

Python

```
def F(n):
    if n > 2:
        return F(n - 1)+ G(n - 2)
    else: return n+1

def G(n):
    if n > 2:
        return G(n - 1)+ F(n - 2)
    else: return n
```

Паскаль

```
function F(n: integer): integer;
begin
    if n > 2 then
        F := F(n - 1) + G(n - 2)
    else
        F := n+1;
end;

function G(n: integer): integer;
begin
    if n > 2 then
        G := G(n - 1) + F(n - 2)
    else
        G := n;
end;
```

Алгоритмический язык

```
алг цел F(цел n)
нач
    если n > 2
        то
            знач := F(n - 1)+G(n - 2)
        иначе
            знач := n+1
    все
кон

алг цел G(цел n)
нач
    если n > 2
        то
            знач := G(n - 1)+F(n - 2)
        иначе
            знач := n
    все
кон
```

Си

```
int F(int n)
{
if (n > 2)
return F(n - 1) + G(n - 2);
else return n+1;
}
int G(int n)
{
if (n > 2)
return G(n - 1) + F(n -2);
else return n;
}
```

Чему будет равно значение, вычисленное при выполнении вызова G(7)?

20. Ниже на пяти языках программирования записана рекурсивная функция F.

Бейсик

```
FUNCTION F(n)
    IF n > 2 THEN
         F = F(n-2) + F(n\2)
     ELSE
         F = n
    END IF
END FUNCTION
```

Python

```
def F(n):
    if n > 2:
        return F(n-2) + F(n//2)
    else:
        return n
```

Паскаль

```
function F(n: integer): integer;
begin
    if n > 2 then
        F := F(n-2) + F(n div 2)
    else
        F := n
end;
```

Алгоритмический язык

```
алг цел F(цел n)
нач
    если n > 2
        то
         знач := F(n-2) + F(div(n,2))
        иначе
            знач := n
    все
кон
```

Си

```
int F(int n)
{
    if (n > 2)
        return F(n-2) + F(n/2);
    else
        return n;
}
```

Чему будет равно значение, вычисленное при выполнении вызова F(9)?