1. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    PRINT n
    IF n < 5 THEN
        F(n + 1)
        F(n + 3)
    END IF
END SUB
```

Python

```
def F(n):
    print(n)
    if n < 5:
        F(n + 1)
        F(n + 3)
```

Паскаль

```
procedure F(n: integer);
    begin
        writeln(n);
        if n < 5 then
            begin
                F(n + 1);
                F(n + 3)
            end
    end
```

Алгоритмический язык

```
алг F(цел n)
нач
вывод n, нс
если n < 5 то
    F(n + 1)
    F(n + 3)
все
кон
```

C++

```
void F(int n)
{
    cout << n << endl;
    if (n < 5) {
        F(n + 1);
        F(n + 3);
    }
}
```

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(1)?

2. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    PRINT n
    IF n > 0 THEN
        F(n - 1)
        F(n - 3)
     END IF
END SUB
```

Python

```
def F(n):
    print(n)
    if n > 0:
        F(n - 1)
        F(n - 3)
```

Паскаль

```
procedure F(n: integer);
    begin
        writeln(n);
        if n > 0 then
            begin
                F(n - 1);
                F(n - 3)
            end
end
```

Алгоритмический язык

```
алг F(цел n)
нач
вывод n, нс
если n > 0 то
    F(n - 1)
    F(n - 3)
все
кон
```

Си

```
void F(int n)
{
    cout << n;
    if (n > 0)
    {
        F(n - 1);
        F(n - 3);
    }
}
```

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(5)?

3. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    PRINT n
    IF n > 1 THEN
        F(n - 1)
        F(n - 3)
    END IF
END SUB
```

Python

```
def F(n):
    print(n)
    if n > 1:
        F(n - 1)
        F(n - 3)
```

Паскаль

```
procedure F(n: integer);
    begin
        writeln(n);
        if n > 1 then
            begin
                F(n - 1);
                F(n - 3)
            end
    end
```

Алгоритмический язык

```
алг F(цел n)
нач
вывод n, нс
если n > 1 то
    F(n - 1)
    F(n - 3)
все
кон
```

C++

```
void F(int n)
{
    cout << n;
    if (n > 1)
    {
        F(n - 1);
        F(n - 3);
    }
}
```

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(6)?

4. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    PRINT n
    IF n < 5 THEN
        F(n + 1)
        F(n + 2)
    END IF
END SUB
```

Python

```
def F(n):
    print(n)
    if n < 5:
        F(n + 1)
        F(n + 2)
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n, нс
    если n < 5 то
        F(n + 1)
        F(n + 2)
    все
кон
```

Паскаль

```
procedure F(n: integer);
begin
    writeln(n);
    if n < 5 then
    begin
        F(n + 1);
        F(n + 2)
    end
end
```

Си

```
void F(int n)
{
    cout << n;
    if (n < 5)
    {
        F(n + 1);
        F(n + 2);
    }
}
```

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(2)?

5. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    PRINT n
    IF n < 4 THEN
        F(n + 1)
        F(n + 3)
    END IF
END SUB
```

Python

```
def F(n):
    print(n)
    if n < 4:
        F(n + 1)
        F(n + 3)
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n, нс
    если n < 4 то
        F(n + 1)
        F(n + 3)
    все
кон
```

Паскаль

```
procedure F(n: integer);
begin
    writeln(n);
    if n < 4 then
    begin
        F(n + 1);
        F(n + 3)
    end
end
```

Си

```
void F(int n)
{
    cout << n;
    if (n < 4)
    {
        F(n + 1);
        F(n + 3);
    }
}
```

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(1)?

6. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
  IF n > 0 THEN
    F(n - 4)
    PRINT n
    F(n \ 3)
  END IF
END SUB
```

Паскаль

```
procedure F(n: integer);
begin
  if n > 0 then
  begin
    F(n - 4);
    writeln(n);
    F(n div 3)
  end
end;
```

Си

```
void F(int n)
{
if (n > 0)
  {
    F(n - 4);
    cout << n;
    F(n / 3);
  }
}
```

Алгоритмический язык

```
алг F(цел n)
нач
  если n > 0 то
   F(n - 4)
   вывод n, нс
   F(div(n, 3))
  все
кон
```

Python

```
def F(n):
    if n > 0:
        F(n - 4)
        print(n)
        F(n // 3)
```

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(9)?

7. Ниже на четырёх языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
  IF n > 0 THEN
    F(n - 4)
    F(n \ 3)
    PRINT n
  END IF
END SUB
```

Паскаль

```
procedure F(n: integer);
begin
  if n > 0 then
  begin
    F(n - 4);
    F(n div 3);
    writeln(n)
  end
end;
```

Си++

```
void F(int n)
{
if (n > 0)
  {
    F(n - 4);
    F(n / 3);
    cout << n << endl;
  }
}
```

Алгоритмический язык

```
алг F(цел n)
нач
  если n > 0 то
    F(n - 4)
    F(div(n, 3))
    вывод n, нс
  все
кон
```

Python

```
def F(n):
    if n > 0:
        F(n - 4)
        F(n // 3)
        print(n)
```

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(10)?

8. Ниже на пяти языках программирования записана рекурсивная функция (процедура) *F*.

Бейсик

```
SUB F(n)
    PRINT n,
    IF n > 2 THEN
      F(n – 3)
      F(n – 2)
      F(n – 1)
    END IF
END SUB
```

Python

```
def F(n):
    print (n, end='')
    if n > 2:
        F(n – 3)
        F(n – 2)
        F(n – 1)
```

Паскаль

```
procedure F(n: integer);
begin
    write(n);
    if n > 2 then
    begin
      F(n – 3);
      F(n – 2);
      F(n – 1)
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n
    если n > 2 то
      F(n – 3)
      F(n – 2)
      F(n – 1)
    все
кон
```

Си

```
void F(int n ){
    cout « n « endl;
    if (n > 2) {
        F(n - 3);
        F(n - 2);
        F(n - 1);
    }
}
```

Что выведет программа при вызове F(4)? В ответе запишите последовательность выведенных цифр слитно (без пробелов).

9. Ниже на пяти языках программирования записана рекурсивная функция (процедура) *F*.

Бейсик

```
SUB F(n)
    PRINT n,
    IF n > 2 THEN
      F(n – 1)
      F(n – 2)
      F(n – 3)
    END IF
END SUB
```

Python

```
def F(n):
    print (n)
    if n > 2:
        F(n – 1)
        F(n – 2)
        F(n – 3)
```

Паскаль

```
procedure F(n: integer);
begin
    write(n);
    if n > 2 then
    begin
      F(n – 1);
      F(n – 2);
      F(n – 3)
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n
    если n > 2 то
      F(n – 1)
      F(n – 2)
      F(n – 3)
    все
кон
```

Си

```
void F(int n ){
    cout « n « endl;
    if (n > 2) {
        F(n – 1);
        F(n – 2);
        F(n – 3);
    }
}
```

Что выведет программа при вызове F(4)? В ответе запишите последовательность выведенных цифр слитно (без пробелов).

10. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
  IF n > 0 THEN
    PRINT "*"
    F(n - 1)
    F(n \ 3)
  END IF
END SUB
```

Python

```
def F(n):
    if n > 0:
        print("*")
        F(n - 1)
        F(n // 3)
```

Алгоритмический язык

```
алг F(цел n)
нач
  если n > 0 то
    вывод "*"
    F(n - 1)
    F(div(n, 3))
  все
кон
```

Паскаль

```
procedure F(n: integer);
begin
  if n > 0 then
  begin
    writeln('*');
    F(n - 1);
    F(n div 3)
  end
end
```

Си

```
void F(int n)
{
  if (n > 0)
  {
    printf("*");
    F(n - 1);
    F(n / 3);
  }
}
```

Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(6)?

11. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
DECLARE SUB F(n)
SUB F(n)
    IF n > 2 THEN
        PRINT n
        F(n - 3)
        F(n – 4)
    END IF
END SUB
```

Python

```
def F(n):
    if n > 2:
        print(n)
        F(n - 3)
        F(n – 4)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 2 then begin
        writeln(n);
        F(n - 3);
        F(n – 4)
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 2 то
        вывод n, нс
        F(n - 3)
        F(n – 4)
    все
кон
```

Си

```
void F(int n) {
    if (n > 2) {
        cout << n;
        F(n - 3);
        F(n – 4);
    }
}
```

Чему равна сумма напечатанных на экране чисел при выполнении вызова *F*(10)?

12. Ниже на пяти языках программирования записана рекурсивная функция (процедура) F.

Бейсик

```
SUB F(n)
  PRINT n,
  IF n >= 3 THEN
    F(n - 1)
    F(n - 3)
  END IF
END SUB
```

Python

```
def F(n):
    print(n)
    if n >= 3:
        F(n - 1)
        F(n - 3)
```

Паскаль

```
procedure F(n: integer);
begin
  write(n);
  if n >= 3 then
  begin
    F(n - 1);
    F(n - 3)
  end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
  вывод n
  если n >= 3 то
    F(n - 1)
    F(n - 3)
  все
кон
```

Си

```
void F(int n) {
    cout « n « endl;
    if (n >= 3) {
        F(n - 1);
        F(n - 3);
    }
}
```

Что выведет программа при вызове F(5)? В ответе запишите последовательность выведенных цифр слитно (без пробелов).

13. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
DECLARE SUB F(n)
SUB F(n)
    IF n > 0 THEN
        PRINT n
        F(n - 3)
        F(n \ 3)
    END IF
END SUB
```

Python

```
def F(n):
    if n > 0:
        print(n)
        F(n - 3)
        F(n // 3)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 0 then begin
        writeln(n);
        F(n - 3);
        F(n div 3)
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 0 то
        вывод n, нс
        F(n - 3)
        F(div(n, 3))
    все
кон
```

Си++

```
void F(int n) {
    if (n > 0) {
        std: :cout << n;
        F(n - 3);
        F(n / 3);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

14. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    IF n > 0 THEN
         F(n - 3)
         F(n \ 3)
         PRINT N
    END IF
END SUB
```

Python

```
def F(n):
    if n > 0:
        F(n - 3)
        F(n // 3)
        print(n)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 0 then begin
        F(n – 3);
        F(n div 3);
        writeln(n);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 0 то
        F(n – 3)
        F(div(n,3))
        вывод n
    все
кон
```

C++

```
void F (int n)
{
     if (n > 0) {
        F (n - 3);
        F (n / 3);
        std::cout << n;
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

15. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    IF n > 0 THEN
         F(n \ 3)
         F(n – 3)
         PRINT N
    END IF
END SUB
```

Python

```
def F(n):
    if n > 0:
        F(n // 3)
        F(n – 3)
        print(n)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 0 then begin
        F(n div 3);
        F(n – 3);
        write(n)
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 0 то
        F(div(n,3))
        F(n – 3)
        вывод n
    все
кон
```

C++

```
void F (int n)
{
     if (n > 0) {
        F (n / 3);
        F (n – 3);
        std::cout << n;
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

16. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
DECLARE SUB F(n)
   SUB F (n)
     IF n > 0 THEN
         F(n \ 4)
         PRINT n
         F(n - 1)
     END IF
   END SUB
```

Python

```
def F(n):
    if n > 0:
        F(n // 4)
        print(n)
        F (n - 1)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 0 then
    begin
        F(n div 4);
        write(n);
        F(n - 1);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 0 то
        F(div(n, 4))
        вывод n
        F(n - 1)
    все
кон
```

Си++

```
void F(int n){
    if (n > 0){
        F(n / 4);
        std::cout << n;
        F(n - 1);
    }
}
```

В качестве ответа укажите последовательность цифр, которая будет напечатана на экране в результате вызова F(5).

17. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    IF n > 0 THEN
         F(n – 3)
         PRINT N
         F(n \ 3)
    END IF
END SUB
```

Python

```
def F(n):
    if n > 0:
        F(n – 3)
        print(n)
        F(n // 3)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 0 then begin
        F(n – 3);
        write(n);
        F(n div 3);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 0 то
        F(n – 3)
        вывод n
        F(div(n,3))
    все
кон
```

C++

```
void F (int n)
{
     if (n > 0) {
        F (n – 3);
        std::cout << n;
        F (n / 3);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

18. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    IF n > 0 THEN
         F(n \ 3)
         PRINT N
         F(n – 3)
    END IF
END SUB
```

Python

```
def F(n):
    if n > 0:
        F(n // 3)
        print(n)
        F(n – 3)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 0 then begin
        F(n div 3);
        write(n);
        F(n – 3);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 0 то
        F(div(n,3))
        вывод n
        F(n – 3)
    все
кон
```

C++

```
void F (int n)
{
     if (n > 0) {
        F (n / 3);
        std::cout << n;
        F (n – 3);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

19. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F (n)
  IF n > 0 THEN
      F(n – 1)
      PRINT n
      F(n \ 4)
  END IF
END SUB
```

Python

```
def F(n):
    if n > 0:
        F(n – 1)
        print(n)
        F (n // 4)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 0 then
    begin
        F(n – 1);
        write(n);
        F(n div 4);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 0 то
        F(n – 1)
        вывод n
        F(div(n, 4))
    все
кон
```

Си++

```
void F(int n){
    if (n > 0){
        F(n – 1)
        std::cout << n;
        F(n / 4);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(5). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

20. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    IF n > 0 THEN
         PRINT N
         F(n – 4)
         F(n \ 2)
    END IF
END SUB
```

Python

```
def F(n):
    if n > 0:
        print(n)
         F(n – 4)
         F(n // 2)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 0 then begin
        write(n);
        F(n – 4);
        F(n div 2);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 0 то
        вывод n
        F(n – 4)
        F(div(n,2))
    все
кон
```

C++

```
void F (int n)
{
     if (n > 0) {
        std::cout << n;
        F (n – 4);
        F (n / 2);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

21. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    IF n > 0 THEN
         PRINT N
         F(n \ 2)
         F(n - 4)
    END IF
END SUB
```

Python

```
def F(n):
    if n > 0:
        print(n)
        F(n // 2)
        F(n - 4)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 0 then begin
        write(n);
        F(n div 2);
        F(n - 4);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 0 то
        вывод n
        F(div(n,2))
        F(n - 4)
    все
кон
```

C++

```
void F (int n)
{
     if (n > 0) {
        std::cout << n;
        F (n / 2);
        F (n - 4);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

22. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    IF n > 2 THEN
         PRINT N
         F(n \ 2)
         F(n – 1)
    END IF
END SUB
```

Python

```
def F(n):
    if n > 2:
        print(n)
        F(n // 2)
        F(n – 1)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 2 then begin
        write(n);
        F(n div 2);
        F(n – 1);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 2 то
        вывод n
        F(div(n,2))
        F(n – 1)
    все
кон
```

C++

```
void F (int n)
{
     if (n > 2) {
        std::cout << n;
        F (n / 2);
        F (n – 1);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(7). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

23. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
DECLARE SUB F(n)
SUB F(n)
    IF n > 0 THEN
        F(n – 1)
        PRINT n
        F(n – 2)
    END IF
END SUB
```

Python

```
def F(n):
    if n > 0:
        F(n – 1)
        print(n)
        F(n – 2)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 0 then begin
        F(n – 1);
        writeln(n);
        F(n – 2)
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 0 то
        F(n – 1)
        вывод n
        F(n – 2)
    все
кон
```

Си++

```
void F(int n) {
    if (n > 0) {
        F(n – 1);
        std: :cout << n;
        F(n – 2);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(4). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

24. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    IF n < 8 THEN
         F(n + 3)
         PRINT N
         F(2 * n)
    END IF
END SUB
```

Python

```
def F(n):
    if n < 8:
        F(n + 3)
        print(n)
        F(2 * n)
```

Паскаль

```
procedure F(n: integer);
begin
    if n < 8 then begin
        F(n + 3);
        write(n);
        F(2 * n);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n < 8 то
        F(n + 3)
        вывод n
        F(2 * n)
    все
кон
```

C++

```
void F (int n)
{
     if (n < 8) {
        F (n + 3);
        std::cout << n;
        F (2 * n);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова F(1). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

25. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    IF n < 8 THEN
         F(2 * n)
         PRINT N
         F(n + 3)
    END IF
END SUB
```

Python

```
def F(n):
    if n < 8:
        F(2 * n)
        print(n)
        F(n + 3)
```

Паскаль

```
procedure F(n: integer);
begin
    if n < 8 then begin
        F(2 * n);
        write(n);
        F(n + 3);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n < 8 то
        F(2 * n)
        вывод n
        F(n + 3)
    все
кон
```

C++

```
void F (int n)
{
     if (n < 8) {
        F (2 * n);
        std::cout << n;
        F (n + 3);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова F(1). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

26. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    IF n < 8 THEN
         PRINT N
         F(2 * n)
         F(n + 3)
    END IF
END SUB
```

Python

```
def F(n):
    if n < 8:
        print(n)
         F(2 * n)
         F(n + 3)
```

Паскаль

```
procedure F(n: integer);
begin
    if n < 8 then begin
        write(n);
        F(2 * n);
        F(n + 3);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n < 8 то
        вывод n
        F(2 * n)
        F(n + 3)
    все
кон
```

C++

```
void F (int n)
{
     if (n < 8) {
        std::cout << n;
        F (2 * n);
        F (n + 3);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова F(1). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

27. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    IF n < 8 THEN
         F(n + 3)
         F(2 * n)
         PRINT N
    END IF
END SUB
```

Python

```
def F(n):
    if n < 8:
        F(n + 3)
        F(2 * n)
        print(n)
```

Паскаль

```
procedure F(n: integer);
begin
    if n < 8 then begin
        F(n + 3);
        F(2 * n);
        write(n);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n < 8 то
        F(n + 3)
        F(2 * n)
        вывод n
    все
кон
```

C++

```
void F (int n)
{
     if (n < 8) {
        F(n + 3);
        F(2 * n);
        std::cout << n;
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова F(1). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

28. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    IF n < 10 THEN
         F(n + 3)
         PRINT N
         F(3 * n)
    END IF
END SUB
```

Python

```
def F(n):
    if n < 10:
         F(n + 3)
        print(n)
         F(3 * n)
```

Паскаль

```
procedure F(n: integer);
begin
    if n < 10 then begin
        F(n + 3);
        write(n);
        F(3 * n);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n < 10 то
        F(n + 3)
        вывод n
        F(3 * n)
    все
кон
```

C++

```
void F (int n)
{
     if (n < 10) {
        F (n + 3);
        std::cout << n;
        F (3 * n);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова F(1). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

29. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    IF n < 10 THEN
         F(3 * n)
         PRINT N
         F(n + 3)
    END IF
END SUB
```

Python

```
def F(n):
    if n < 10:
         F(3 * n)
        print(n)
         F(n + 3)
```

Паскаль

```
procedure F(n: integer);
begin
    if n < 10 then begin
        F(3 * n);
        write(n);
        F(n + 3);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n < 10 то
        F(3 * n)
        вывод n
        F(n + 3)
    все
кон
```

C++

```
void F (int n)
{
     if (n < 10) {
        F (3 * n);
        std::cout << n;
        F (n + 3);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова F(1). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

30. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(N)
    PRINT N
    IF N >= 3 THEN
        F(n – 1)
        F(n – 2)
        F(n – 2)
    END IF
END SUB
```

Python

```
def F(n):
    print(n)
    if n >= 3:
        F(n – 1)
        F(n – 2)
        F(n – 2)
```

Паскаль

```
procedure F(n: integer);
begin
    write(n);
    if n >= 3 then begin
        F(n – 1);
        F(n – 2);
        F(n – 2)
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n
    если n >= 3 то
        F(n – 1)
        F(n – 2)
        F(n – 2)
    все
кон
```

C++

```
void F(int n)
{
    std::cout << n;
    if (n >= 3) {
        F(n – 1);
        F(n – 2);
        F(n – 2);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова F(4). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

31. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(N)
    PRINT N
    IF N >= 2 THEN
        F(n – 2)
        F(n – 1)
        F(n – 2)
    END IF
END SUB
```

Python

```
def F(n):
    print(n)
    if n >= 2:
        F(n – 2)
        F(n – 1)
        F(n – 2)
```

Паскаль

```
procedure F(n: integer);
begin
    write(n);
    if n >= 2 then begin
        F(n – 2);
        F(n – 1);
        F(n – 2)
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n
    если n >= 2 то
        F(n – 2)
        F(n – 1)
        F(n – 2)
    все
кон
```

C++

```
void F(int n)
{
    std::cout << n;
    if (n >= 2) {
        F(n – 2);
        F(n – 1);
        F(n – 2);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут выведены на экран при выполнении вызова F(3). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

32. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    IF n < 10 THEN
        F(3*n - 1)
        PRINT N
        F(2*n + 1)
    END IF
END SUB
```

Python

```
def F(n):
    if n < 10:
        F(3*n - 1)
        print(n)
        F(2*n + 1)
```

Паскаль

```
procedure F(n: integer);
begin
    if n < 10 then begin
        F(3*n - 1);
        write(n);
        F(2*n + 1)
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n < 10 то
        F(3*n - 1)
        вывод n
        F(2*n + 1)
    все
кон
```

C++

```
void F(int n)
{
    if (n < 10) {
        F(3*n - 1);
        std::cout << n;
        F(2*n + 1);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова F(1). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

33. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    PRINT N
    IF n < 5 THEN
        F(n + 3)
        F(2 * n)
    END IF
END SUB
```

Python

```
def F(n):
    print(n)
    if n < 5:
        F(n + 3)
        F(2 * n)
```

Паскаль

```
procedure F(n: integer);
begin
    write(n);
    if n < 5 then begin
        F(n + 3);
        F(2 * n)
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n
    если n < 5 то
        F(n + 3)
        F(2 * n)
    все
кон
```

C++

```
void F(int n)
{
    std::cout << n;
    if (n < 5) {
        F(n + 3);
        F(2 * n);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова F(1). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

34. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    PRINT N
    IF n < 5 THEN
        F(2 * n)
        F(n + 3)
    END IF
END SUB
```

Python

```
def F(n):
    print(n)
    if n < 5:
        F(2 * n)
        F(n + 3)
```

Паскаль

```
procedure F(n: integer);
begin
    write(n);
    if n < 5 then begin
        F(2 * n);
        F(n + 3)
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n
    если n < 5 то
        F(2 * n)
        F(n + 3)
    все
кон
```

C++

```
void F(int n)
{
    std::cout << n;
    if (n < 5) {
        F(2 * n);
        F(n + 3);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова F(1). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

35. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    PRINT N
    IF n >= 3 THEN
         F(n - 1)
         F(n \ 2)
    END IF
END SUB
```

Python

```
def F(n):
    print(n)
    if n >= 3:
         F(n - 1)
         F(n // 2)
```

Паскаль

```
procedure F(n: integer);
begin
    write(n);
    if n >= 3 then begin
        F(n - 1);
        F(n div 2);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n
    если n >= 3 то
        F(n - 1)
        F(div(n,2))
    все
кон
```

C++

```
void F (int n)
{
    std::cout << n;
     if (n >= 3) {
        F (n - 1);
        F (n / 2);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(5). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

36. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    PRINT N
    IF n >= 3 THEN
         F(n - 1)
         F(n - 1)
    END IF
END SUB
```

Python

```
def F(n):
    print(n)
    if n >= 3:
         F(n - 1)
         F(n - 1)
```

Паскаль

```
procedure F(n: integer);
begin
    write(n);
    if n >= 3 then begin
        F(n - 1);
        F(n - 1);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n
    если n >= 3 то
        F(n - 1)
        F(n - 1)
    все
кон
```

C++

```
void F (int n)
{
    std::cout << n;
     if (n >= 3) {
        F (n - 1);
        F (n - 1);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(4). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

37. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    PRINT N
    IF n >= 6 THEN
         F(n – 1)
         F(n – 3)
    END IF
END SUB
```

Python

```
def F(n):
    print(n)
    if n >= 6:
         F(n – 1)
         F(n – 3)
```

Паскаль

```
procedure F(n: integer);
begin
    write(n);
    if n >= 6 then begin
        F(n – 1);
        F(n – 3);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n
    если n >= 6 то
        F(n – 1)
        F(n – 3)
    все
кон
```

C++

```
void F (int n)
{
    std::cout << n;
     if (n >= 6) {
        F (n – 1);
        F (n – 3);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(8). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

38. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    PRINT n,
    IF n >= 3 THEN
        F(n \ 2)
        F(n - 1)
    END IF
END SUB
```

Python

```
def F(n):
    print(n, end='')
    if n >= 3:
        F(n // 2)
        F(n - 1)
```

Паскаль

```
procedure F(n: integer);
begin
    write(n);
    if n >= 3 then
    begin
        F(n div 2);
        F(n - 1)
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n
    если n >= 3 то
        F(div(n, 2))
        F(n - 1)
    все
кон
```

Си++

```
void F(int n) {
    std::cout << n;
    if (n >= 3) {
        F(n / 2);
        F(n - 1);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут выведены на экран при выполнении вызова F(5). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

39. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    IF n > 0 THEN
        PRINT n,
        F(n – 3)
        F(n \ 2)
    END IF
END SUB
```

Python

```
def F(n):
    if n > 0:
        print(n, end='')
        F(n – 3)
        F(n // 2)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 0 then
    begin
        write(n);
        F(n – 3)
        F(n div 2);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 0 то
        вывод n
        F(n – 3)
        F(div(n, 2))
    все
кон
```

Си++

```
void F(int n) {
    if (n > 0) {
        std::cout << n;
        F(n – 3);
        F(n / 2);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут выведены на экран при выполнении вызова F(7). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

40. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    IF n > 0 THEN
        F(n – 3)
        PRINT n
        F(n \ 2)
    END IF
END SUB
```

Python

```
def F(n):
    if n > 0:
        F(n – 3)
        print(n, end='')
        F(n // 2)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 0 then
    begin
        F(n – 3);
        write(n);
        F(n div 2);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 0 то
        F(n – 3)
        вывод n
        F(div(n, 2))
    все
кон
```

Си++

```
void F(int n) {
    if (n > 0) {
        F(n – 3);
        std::cout << n;
        F(n / 2);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут выведены на экран при выполнении вызова F(7). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

41. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    IF n > 0 THEN
         PRINT N
         F(n \ 3)
         F(n – 2)
    END IF
END SUB
```

Python

```
def F(n):
    if n > 0:
        print(n)
        F(n // 3)
        F(n – 2)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 0 then begin
        write(n);
        F(n div 3);
        F(n – 2);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 0 то
        вывод n
        F(div(n,3))
        F(n – 2)
    все
кон
```

C++

```
void F (int n)
{
     if (n > 0) {
        std::cout << n;
        F (n / 3);
        F (n – 2);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(7). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

42. Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    IF n > 2 THEN
         F(n \ 2)
         F(n – 1)
         PRINT N
    END IF
END SUB
```

Python

```
def F(n):
    if n > 2:
        F(n // 2)
        F(n – 1)
        print(n)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 2 then begin
        F(n div 2);
        F(n – 1);
        write(n);
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 2 то
        F(div(n,2))
        F(n – 1)
        вывод n
    все
кон
```

C++

```
void F (int n)
{
     if (n > 2) {
        F (n / 2);
        F (n – 1);
        std::cout << n;
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(7). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

43. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    IF n > 3 THEN
         F(n \ 2)
         F(n – 2)
    END IF
    PRINT N
END SUB
```

Python

```
def F(n):
    if n > 3:
        F(n // 2)
        F(n – 2)
    print(n)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 3 then begin
        F(n div 2);
        F(n – 2);
    end;
    write(n);
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 3 то
        F(div(n,2))
        F(n – 2)
    все
    вывод n
кон
```

C++

```
void F (int n)
{
     if (n > 3) {
        F (n / 2);
        F (n – 2);
    }
    std::cout << n;
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова *F*(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

44. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    IF n > 3 THEN
         F(n - 2)
         F(n \ 2)
    END IF
    PRINT N
END SUB
```

Python

```
def F(n):
    if n > 3:
        F(n - 2)
        F(n // 2)
    print(n)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 3 then begin
        F(n - 2);
        F(n div 2);
    end;
    write(n);
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 3 то
        F(n - 2)
        F(div(n,2))
    все
    вывод n
кон
```

C++

```
void F (int n)
{
     if (n > 3) {
        F (n - 2);
        F (n / 2);
    }
    std::cout << n;
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова *F*(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

45. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    IF n > 3 THEN
         F(n \ 2)
         F(n – 1)
    END IF
    PRINT N
END SUB
```

Python

```
def F(n):
    if n > 3:
        F(n // 2)
        F(n – 1)
    print(n)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 3 then begin
        F(n div 2);
        F(n – 1);
    end;
    write(n);
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 3 то
        F(div(n,2))
        F(n – 1)
    все
    вывод n
кон
```

C++

```
void F (int n)
{
     if (n > 3) {
        F (n / 2);
        F (n – 1);
    }
    std::cout << n;
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова *F*(7). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

46. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    IF n > 3 THEN
         F(n – 1)
         F(n \ 2)
    END IF
    PRINT N
END SUB
```

Python

```
def F(n):
    if n > 3:
        F(n – 1)
        F(n // 2)
    print(n)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 3 then begin
        F(n – 1);
        F(n div 2);
    end;
    write(n);
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 3 то
        F(n – 1)
        F(div(n,2))
    все
    вывод n
кон
```

C++

```
void F (int n)
{
     if (n > 3) {
        F (n – 1);
        F (n / 2);
    }
    std::cout << n;
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова *F*(7). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

47. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    PRINT n;
    IF n > 3 THEN
        F(n \ 2)
    END IF
    IF n >= 5 THEN
        F(n - 1)
    END IF
END SUB
```

Python

```
def F(n):
    print(n, end='')
    if n > 3:
        F(n // 2)
    if n >= 5:
        F(n - 1)
```

Паскаль

```
procedure F(n: integer);
begin
    write(n);
    if n > 3 then
        F(n div 2);
    if n >= 5 then
        F(n - 1)
    end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n
    если n > 3 то
        F(div(n,2))
    все
    если n >= 5 то
        F(n - 1)
    все
кон
```

C++

```
void F(int n)
{
    std::cout << n;
    if (n > 3)
        F(n / 2);
    if (n >= 5)
        F(n - 1);
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова *F*(8). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

48. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    IF n > 2 THEN
         F(n \ 2)
         PRINT n;
         F(n – 2)
    END IF
END SUB
```

Python

```
def F(n):
    if n > 2:
        F(n // 2)
        print(n, end='')
        F(n – 2)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 2 then begin
        F(n div 2);
        write(n);
        F(n – 2);
    end;
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 2 то
        F(div(n,2))
        вывод n
        F(n – 2)
    все
кон
```

C++

```
void F (int n)
{
     if (n > 2) {
        F (n / 2);
        std::cout << n;
        F (n – 2);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова F(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

49. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    PRINT n;
    IF n >= 5 THEN
        F(n - 1)
    END IF
    IF n > 3 THEN
        F(n \ 2)
    END IF
END SUB
```

Python

```
def F(n):
    print(n, end='')
    if n >= 5:
        F(n - 1)
    if n > 3:
        F(n // 2)
```

Паскаль

```
procedure F(n: integer);
begin
    write(n);
    if n >= 5 then
        F(n - 1)
    if n > 3 then
        F(n div 2);
    end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n
    если n >= 5 то
        F(n - 1)
    все
    если n > 3 то
        F(div(n,2))
    все
кон
```

C++

```
void F(int n)
{
    std::cout << n;
    if (n >= 5)
        F(n - 1);
    if (n > 3)
        F(n / 2);
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова *F*(8). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

50. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    PRINT n;
    IF n > 3 THEN
        F(n \ 2)
        F(n - 1)
    END IF
END SUB
```

Python

```
def F(n):
    print(n, end='')
    if n > 3:
        F(n // 2)
        F(n - 1)
```

Паскаль

```
procedure F(n: integer);
begin
    write(n);
    if n > 3 then begin
        F(n div 2);
        F(n - 1)
    end
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n
    если n > 3 то
        F(div(n,2))
        F(n - 1)
    все
кон
```

C++

```
void F(int n)
{
    std::cout << n;
    if (n > 3) {
        F(n / 2);
        F(n - 1);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова *F*(7). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

51. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    PRINT n;
    IF n > 3 THEN
        F(n - 1)
        F(n \ 2)
    END IF
END SUB
```

Python

```
def F(n):
    print(n, end='')
    if n > 3:
        F(n - 1)
        F(n // 2)
```

Паскаль

```
procedure F(n: integer);
begin
    write(n);
    if n > 3 then begin
        F(n - 1);
        F(n div 2)
    end;
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    вывод n
    если n > 3 то
        F(n - 1)
        F(div(n,2))
    все
кон
```

C++

```
void F(int n)
{
    std::cout << n;
    if (n > 3) {
        F(n - 1);
        F(n / 2);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова *F*(7). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

52. Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик

```
SUB F(n)
    IF n > 2 THEN
         F(n – 2)
         PRINT n;
         F(n \ 2)
    END IF
END SUB
```

Python

```
def F(n):
    if n > 2:
        F(n – 2)
        print(n, end='')
        F(n // 2)
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 2 then begin
        F(n – 2);
        write(n);
        F(n div 2);
    end;
end;
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 2 то
        F(n – 2)
        вывод n
        F(div(n,2))
    все
кон
```

C++

```
void F (int n)
{
     if (n > 2) {
        F (n – 2);
        std::cout << n;
        F (n / 2);
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут показаны на экране при выполнении вызова F(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.