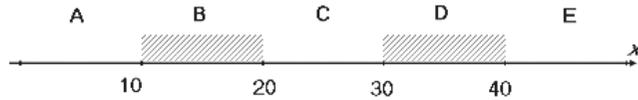


**Критерии оценивания заданий с развёрнутым ответом**

**C1** Требовалось написать программу, при выполнении которой с клавиатуры считывается координата точки на прямой ( $x$  – действительное число) и определяется принадлежность этой точки заданной области (включая границы). Программист торопился и написал программу неправильно.

**Бейсик**

```
INPUT x
IF x<=30 THEN
  IF x<=20 THEN
    IF x<=10 THEN
      PRINT "не принадлежит"
    ELSE
      PRINT "принадлежит"
    END IF
  END IF
END IF
END IF
END
```

**Паскаль**

```
var x: real;
begin
  readln(x);
  if x<=30 then
    if x<=20 then
      if x<=10 then
        write('не принадлежит')
      else
        write('принадлежит')
      end if
    end if
  end if
end.
```

**Си**

```
#include <stdio.h>
void main()
{
  float x;
  scanf("%f", &x);
  if (x<=30)
    if (x<=20)
      if (x<=10)
        printf("не принадлежит");
      else
        printf("принадлежит");
    }
}
```

**Алгоритмический язык**

```
алг
нач
  веш x
  ввод x
  если x<=30 то
    если x<=20 то
      если x<=10 то
        вывод 'не принадлежит'
      иначе
        вывод 'принадлежит'
      все
    все
  все
кон
```

Последовательно выполните следующее.

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D и E). Точки, лежащие на границах областей (то есть, точки 10, 20, 30 и 40) при заполнении таблицы не рассматриваются.

Область	Условие 1 ( $x \leq 30$ )	Условие 2 ( $x \leq 20$ )	Условие 3 ( $x \leq 10$ )	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					

В столбцах условий укажите «да», если условие выполнится, «нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце «Программа выведет» укажите, что программа выведет на экран. Если программа ничего не выводит, поставьте «—» (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите «да» или «нет».

2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

**Содержание верного ответа и указания по оцениванию**

(допускаются иные формулировки ответа, не искажающие его смысла)

Элементы ответа:

1.

Область	Условие 1 ( $x \leq 30$ )	Условие 2 ( $x \leq 20$ )	Условие 3 ( $x \leq 10$ )	Программа выведет	Область обрабатывается верно
A	да	да	да	не принадлежит	Да
B	да	да	нет	принадлежит	Да
C	да	нет	—	—	Нет
D	нет	—	—	—	Нет
E	нет	—	—	—	Нет

2. Возможная доработка (Паскаль):

```
if (x>=10) and (x<=20) or (x>=30) and (x<=40) then
  write('принадлежит')
else
  write('не принадлежит')
```

Возможны и другие способы доработки.

Например:

```
if x>=10 then
  if x<=20 then
    write('принадлежит')
  else
    if x>=30 then
      if x<=40 then
        write('принадлежит')
      else
        write('не принадлежит')
    else
      write('не принадлежит')
else
  write('не принадлежит')
```

Другой пример:

```
if abs(abs(x-25)-10)<=5 then
  write('принадлежит')
else
  write('не принадлежит')
```

**Обратите внимание!** В задаче требовалось выполнить **три** действия: указать для каждой области, как будет работать программа, что она выведет на экран и правильно ли это (в виде таблицы) и исправить две ошибки.

Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.

1. Верное заполнение предложенной таблицы.

2. Неправильное использование условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего

(отсутствуют случаи ELSE). Исправлением этой ошибки может быть либо добавление случая ELSE к каждому условию IF, либо объединение всех условий IF в одно при помощи конъюнкции.

В сложных случаях это действие считается выполненным, если программа выдаёт одно из двух сообщений «принадлежит» или «не принадлежит» для любых чисел  $x$ , при этом программа не стала работать хуже, чем раньше, то есть для всех точек, для которых программа ранее выдавала верный ответ, доработанная программа также должна выдавать верный ответ.

3. Приведённых трёх ограничений недостаточно для описания двух областей (потеряно условие  $x \leq 40$ ). Кроме того, необходимо учесть, что области не соединены. Исправлением этой ошибки может быть разбиение области на две части и использование дизъюнкции, либо использование сложной (для выведения) математической конструкции ( $||x-25|-10| \leq 5$ ), либо использование сложной комбинации каскадных условий.

В сложных случаях это действие считается выполненным, если верно определены заштрихованные области, то есть программа выводит сообщение «принадлежит» для всех точек закрашенных областей, и только для них, для точек вне заштрихованных областей программа выводит «не принадлежит» или не выводит ничего.

Указания по оцениванию	Баллы
Правильно выполнены оба пункта задания. Исправлены две ошибки. Программа для всех чисел $x$ верно определяет принадлежность точки заштрихованной области. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения	3
1. Правильно выполнены два действия из трёх (исправлены обе ошибки, но в первом пункте задания не приведена таблица (либо таблица содержит ошибки более чем в одной строке), либо приведена таблица (которая содержит ошибки не более чем в одной строке), но исправлена только одна ошибка программы). При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо « $x \leq 40$ » используется « $x < 40$ ».	2
2. Или выполнены все три действия, но при этом в логическом выражении неверно учтены приоритеты логических операций (не расставлены или неправильно расставлены скобки в выражениях)	1
Правильно выполнено только одно действие из трёх, то есть либо только приведена таблица, которая содержит ошибки в не более чем двух строках, либо таблица не приведена (или приведена и содержит ошибки более чем в двух строках), но исправлена одна ошибка программы. При оценивании этого задания на 1 балл допускается не учитывать корректность работы программ на точках границ областей (вместо нестрогих неравенств в решении были использованы строгие неравенства или наоборот)	1
Все пункты задания выполнены неверно (таблица анализа правильности алгоритма не приведена либо содержит три и более строк с ошибками, программа не приведена либо ни одна из двух ошибок не исправлена)	0
<i>Максимальный балл</i>	
	3

**C2** Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 1000. Элемент массива называется хорошим, если это двузначное число, причём цифра в разряде десятков больше, чем цифра в разряде единиц. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести сумму всех хороших элементов массива.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

#### Паскаль

```
const
  N=30;
var
  a: array [1..N] of integer;
  i, j, s: integer;
begin
  for i:=1 to N do
    readln(a[i]);
  ...
end.
```

#### Бейсик

```
N=30
DIM A(N) AS INTEGER
DIM I, J, S AS INTEGER
FOR I = 1 TO N
  INPUT A(I)
NEXT I
...
END
```

#### Си

```
#include <stdio.h>
#define N 30
void main(){
  int a[N];
  int i, j, s;
  for (i=0; i<N; i++)
    scanf("%d", &a[i]);
  ...
}
```

#### Алгоритмический язык

```
алг
нач
  цел N=30
  целтаб a [1:N]
  цел i, j, s
  нц для i от 1 до N
    ввод a[i]
  кц
  ...
кон
```

В качестве ответа Вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например, Free Pascal 2.4) или в виде блок-схемы. В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

**Содержание верного ответа и указания по оцениванию**  
(допускаются иные формулировки ответа, не искажающие его смысла)

#### Пример программы на языке Паскаль

```
s:=0;
for i:=1 to N do begin
  if (10<=a[i]) and (a[i]<=99) and
    ((a[i] div 10) > (a[i] mod 10) ) then s:=s+a[i];
end;
writeln(s);
```

#### Пример программы на языке Бейсик

```
S = 0
FOR I = 1 TO N
  IF 10 <=A(I) AND A(I)<=99 AND A(I)\10 > A(I) MOD 10 THEN
    S = S + A(I)
  END IF
NEXT I
PRINT S
```

#### Пример программы на языке Си

```
s=0;
for (i=0; i<N; i++) {
  if (10<=a[i] && a[i]<=99 && a[i]/10 > a[i]%10)
    s = s +a[i];
}
printf("%d", s);
```

**Пример программы на алгоритмическом языке**

```

s:=0
нц для i от 1 до N
  если 10<=a[i]<=99 и div(a[i],10) > mod(a[i],10)
    то s:=s+a[i]
  все
кц
вывод s

```

Указания по оцениванию	Баллы
Предложен правильный алгоритм, выдающий верное значение. Допускается запись алгоритма на другом языке, использующая аналогичные переменные. В случае, если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных в тексте задания. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается языком программирования, при этом количество переменных и их идентификаторы должны соответствовать условию задачи. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы	2
В любом варианте решения может присутствовать не более одной ошибки из числа следующих: 1) Не инициализируется или неверно инициализируется переменная S. 2) Неверно осуществляется проверка того, что элемент массива – хороший. 3) Вместо проверки того, что элемент хороший, аналогичная проверка выполняется для индекса элемента. 4) Неверно осуществляется накопление суммы в цикле (например, s:=a[i];). 5) Отсутствует вывод ответа. 6) Используется переменная, не объявленная в разделе описания переменных. 7) Не указано или неверно указано условие завершения цикла. 8) Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно. 9) Неверно расставлены операторные скобки	1
Ошибок, перечисленных в п. 1–9, две или больше, или алгоритм сформулирован неверно	0
<i>Максимальный балл</i>	2

**С3** Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить в кучу один камень** или **возвести количество камней в квадрат**. Например, имея кучу из 7 камней, за один ход можно получить кучу из 8 или 49 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится 100 или более. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 100 или больше камней. В начальный момент в куче было  $S$  камней,  $1 \leq S \leq 99$ .

Говорят, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) При каких значениях числа  $S$  Петя может выиграть первым ходом? Укажите все такие значения и выигрышающий ход Пети.

б) Укажите такое значение  $S$ , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите два значения  $S$ , при которых у Пети есть выигрышная стратегия, причем (а) Петя не может выиграть первым ходом, но (б) Петя может выиграть своим вторым ходом, независимо от того, как будет ходить Ваня.

Для указанных значений  $S$  опишите выигрышную стратегию Пети.

3. Укажите такое значение  $S$ , при котором – у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и при этом – у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения  $S$  опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции.

**Содержание верного ответа и указания по оцениванию**

(допускаются иные формулировки ответа, не искажающие его смысла)

1. а) Петя может выиграть, если  $S = 10, \dots, 99$ . Пете достаточно возвести количество камней в квадрат. При  $S < 10$  получить за один ход 100 или больше камней невозможно.

б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет  $S = 9$  камней. Тогда после первого хода Пети в куче будет 10 камней или 81 камень. В обоих случаях Ваня возводит количество камней в квадрат и выигрывает в один ход.

2. Возможные значения  $S$ : 3, 8. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 9 камней (при  $S=3$  он возводит количество камней в квадрат; при  $S=8$  – добавляет 1 камень). Эта позиция разобрана в п. 1 б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть Петя) следующим ходом выигрывает.

3. Возможное значение  $S$ : 7. После первого хода Пети в куче будет 8 или 49 камней. Если в куче станет 49 камней, Ваня возводит количество камней в квадрат и выигрывает своим первым ходом. Ситуация, когда в куче 8 камней разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня) выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Исх. положение	Положения после очередных ходов			
	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)
7	$7+1=8$	$8+1=9$	$9+1=10$	$10*10=100$
			$9*9=81$	$81*81=6561$
	$7*7=49$	$49*49=2401$		

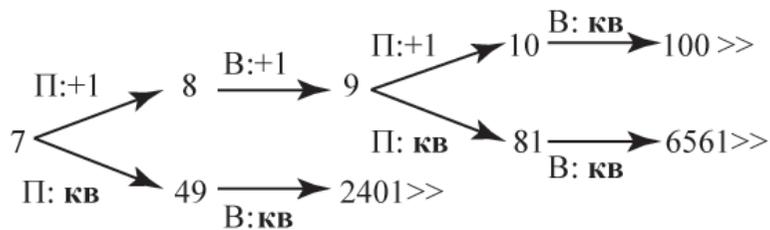


Рис.1. Дерево всех партий, возможных при Ваниной стратегии. Знаком >>> обозначены позиции, в которых партия заканчивается

В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).

Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, например, арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.

Первое задание считается выполненным полностью, если выполнены полностью оба пункта а) и б). Пункт а) считается выполненным полностью, если правильно указаны все позиции, в которых Петя выигрывает первым ходом, и указано, каким должен быть первый ход. Пункт б) считается выполненным, если правильно указана позиция, в которой Ваня выигрывает первым ходом, и описана стратегия Вани, т. е. показано, как Ваня может получить кучу, в которой содержится нужное количество камней, при любом ходе Пети.

Первое задание считается выполненным частично, если (а) правильно указаны все позиции, в которых Петя выигрывает первым ходом, (б) правильно указана позиция, в которой Ваня выигрывает первым ходом, и явно сказано, что при любом ходе Пети Ваня может получить кучу, которая содержит нужное для выигрыша количество камней. Отличие от полного решения в том, что явно не указаны ходы, ведущие к выигрышу.

Второе задание выполнено, если правильно указаны обе позиции, выигрышные для Пети, и описана соответствующая стратегия Пети – так, как это написано в примере решения, или другим способом, например, с помощью дерева всех партий, возможных при выбранной стратегии Пети.

Третье задание выполнено, если правильно указана позиция, выигрышная для Вани, и построено дерево всех партий, возможных при Ваниной стратегии. Должно быть явно сказано, что в этом дереве в каждой позиции, где должен ходить Петя, разобраны все возможные ходы, а для позиций, где должен ходить Ваня, – только ход, соответствующий стратегии, которую выбрал Ваня.

Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом.

Указания по оцениванию	Баллы
Выполнены второе и третье задания. Первое задание выполнено полностью или частично. Здесь и далее в решениях допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу.	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий. Задание 3 выполнено полностью. Первое и второе задания выполнены полностью. Первое задание выполнено полностью или частично; для заданий 2 и 3 указаны правильные значения $S$ .	2
Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий. Первое задание выполнено полностью. Во втором задании правильно указано одно из двух возможных значений $S$ , и для этого значения указана и обоснована выигрышная стратегия Пети. Первое задание выполнено частично, и для одного из остальных заданий правильно указано значение $S$ . Для второго и третьего заданий правильно указаны значения $S$ .	1
Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл	0
<i>Максимальный балл</i>	3

**C4** По каналу связи передаётся последовательность положительных целых чисел  $X_1, X_2, \dots$ . Все числа не превышают 1000, их количество заранее неизвестно. Каждое число передаётся в виде отдельной текстовой строки, содержащей десятичную запись числа. Признаком конца передаваемой последовательности является число 0.

Участок последовательности от элемента  $X_T$  до элемента  $X_{T+N}$  называется подъёмом, если на этом участке каждое следующее число больше или равно предыдущему, причем участок нельзя расширить, т.е.

1)  $T = 1$  или  $X_{T-1} > X_T$

2)  $X_{T+N}$  – последний элемент последовательности или  $X_{T+N} > X_{T+N+1}$ .

Высотой подъёма называется разность  $X_{T+N} - X_T$ . Подъём считается значительным, если высота подъёма больше величины минимального элемента этого подъёма.

Напишите эффективную программу, которая вычисляет количество значительных подъёмов в заданной последовательности.

Программа должна вывести результаты в следующей форме:

Получено чисел: ...

Найдено значительных подъёмов: ...

Размер памяти, которую использует программа, не должен зависеть от длины переданной последовательности чисел.

Перед текстом программы кратко опишите алгоритм решения задачи.

*Пример входных данных:*

144  
17  
21  
27  
3  
7  
9  
11  
25  
0

*Пример выходных данных для приведённого выше примера входных данных:*

Получено чисел: 9

Найдено значительных подъёмов: 1

### Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Программа читает все входные данные один раз, не запоминая все входные данные в массиве. Во время чтения программа помнит число **up** – текущее количество значительных подъёмов, а также необходимые сведения о текущем участке неубывания, например, число **start** – значение первого (и, значит, минимального) элемента участка неубывания и последнее прочитанное число **last** (это число – наибольшее из чисел текущего участка неубывания). Прочитав очередное число **x**, программа сравнивает его с числом **last**. Если  $x < last$ , то фиксируется конец участка неубывания и начало нового участка. Если при этом выполнено условие  $last > 2 * start$ , нужно увеличить количество **up**. При обнаружении конца массива следует таким же способом проверить, является ли последний подъём значительным. Если нужно, следует увеличить значение **up**.

Ниже приведены примеры решения задания на языке Паскаль, на алгоритмическом языке и на языке Бейсик. Допускаются решения, записанные на других языках программирования.

**Пример правильной и эффективной программы на языке Паскаль:**

```
program c4_1;
var
  n, up, x, start, last : integer;
begin
  n:=0;
  up:=0;
  start:=1001;
  last:=1001;
  repeat
    readln(x);
    n:=n+1;
    if x < last then begin
      if last > 2*start then up:=up+1;
      start:=x;
    end;
    last:=x;
  until x = 0;
  writeln('Получено чисел: ', n);
  writeln('Найдено значительных подъёмов: ', up);
end.
```

**Пример правильной и эффективной программы на алгоритмическом языке:**

```
алг C4_1
нач
  цел n, up, x, start, last
  n:=0
  up:=0
  start:=1001
  last:=1001
  нц
    ввод x
    n:=n+1
    если x < last то
      если last > 2*start то up:=up+1 все
      start:=x;
    все
    last:=x;
  кц при x=0
  вывод "Получено чисел: ", n, нс
  вывод "Найдено значительных подъёмов: ", up, нс
кон
```

**Пример правильной и эффективной программы на языке Бейсик:**

```

DIM n, up, x, start, last AS INTEGER
n=0
up=0
start=1001
last=1001
DO
    INPUT x
    n = n + 1
    IF x < last THEN
        IF last > 2 * start THEN up = up + 1
        start = x;
    END IF
    last = x
LOOP UNTIL x = 0
PRINT "Получено чисел: "; n
PRINT "Найдено значительных подъемов: "; up

```

Указания по оцениванию	Баллы
Программа правильно работает для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве. Допускается наличие в тексте программы одной синтаксической ошибки: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку).	4
Программа работает верно, но размер используемой памяти зависит от длины используемой последовательности. Например, входные данные запоминаются в массиве или другой структуре данных (например, контейнер <code>priority_queue</code> , <code>set</code> или <code>map</code> в C++), размер которого соответствует количеству прочитанных чисел. Допускается наличие от одной до трёх синтаксических ошибок. Возможно, в принципиально верно организованном вводе данных есть ошибка.	3
Программа работает в целом верно, эффективно или нет, но в реализации алгоритма содержатся ошибки при инициализации цикла анализа массива данных или обработке конца массива. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.	2
В программе есть блок выделения очередного участка возрастания, однако этот блок написан с ошибками. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше	1
Прочее	0
<i>Максимальный балл</i>	4

**Ответы к заданиям с выбором ответа**

№ задания	Ответ
A1	4
A2	3
A3	2
A4	3
A5	3
A6	2
A7	1

№ задания	Ответ
A8	2
A9	4
A10	4
A11	3
A12	3
A13	1

**Ответы к заданиям с кратким ответом**

№ задания	Ответ
B1	21221
B2	5
B3	4
B4	288
B5	150
B6	31
B7	46
B8	1300

№ задания	Ответ
B9	37
B10	4
B11	СЕСН
B12	1400
B13	44
B14	28
B15	43

**Ответы к заданиям с выбором ответа**

№ задания	Ответ
A1	2
A2	4
A3	3
A4	4
A5	1
A6	1
A7	4

№ задания	Ответ
A8	2
A9	3
A10	2
A11	2
A12	4
A13	2

**Ответы к заданиям с кратким ответом**

№ задания	Ответ
B1	12211
B2	19
B3	3
B4	576
B5	120
B6	63
B7	34
B8	1700

№ задания	Ответ
B9	37
B10	3
B11	DFDH
B12	900
B13	32
B14	32
B15	31