**Решение ЕГЭ информатика**

**1. За­да­ние.** Сколь­ко еди­ниц в дво­ич­ной за­пи­си ше­сна­дца­те­рич­но­го числа 12F016?

**По­яс­не­ние.**

Пе­ре­ве­дем число 12F016 в дво­ич­ную си­сте­му счис­ле­ния: 12F016 = 10010111100002.

Под­счи­та­ем ко­ли­че­ство еди­ниц: их 6.

Ответ: 6.

**2. За­да­ние** Ло­ги­че­ская функ­ция *F* задаётся вы­ра­же­ни­ем (¬*z*)∧*x* ∨ *x*∧*y*. Опре­де­ли­те, ка­ко­му столб­цу таб­ли­цы ис­тин­но­сти функ­ции *F* со­от­вет­ству­ет каж­дая из пе­ре­мен­ных *x, y, z.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Перем. 1** | **Перем. 2** | **Перем. 3** | **Функ­ция** |
| ??? | ??? | ??? | F |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

В от­ве­те на­пи­ши­те буквы *x, y, z* в том по­ряд­ке, в ко­то­ром идут со­от­вет­ству­ю­щие им столб­цы (сна­ча­ла – буква, со­от­вет­ству­ю­щая 1-му столб­цу; затем – буква, со­от­вет­ству­ю­щая 2-му столб­цу; затем – буква, со­от­вет­ству­ю­щая 3-му столб­цу). Буквы в от­ве­те пи­ши­те под­ряд, ни­ка­ких раз­де­ли­те­лей между бук­ва­ми ста­вить не нужно. При­мер. Пусть за­да­но вы­ра­же­ние *x* → *y*, за­ви­ся­щее от двух пе­ре­мен­ных *x* и*y*, и таб­ли­ца ис­тин­но­сти:

|  |  |  |
| --- | --- | --- |
| **Перем. 1** | **Перем. 2** | **Функ­ция** |
| ??? | ??? | F |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Тогда 1-му столб­цу со­от­вет­ству­ет пе­ре­мен­ная *y*, а 2-му столб­цу со­от­вет­ству­ет пе­ре­мен­ная *x*. В от­ве­те нужно на­пи­сать: *yx*.

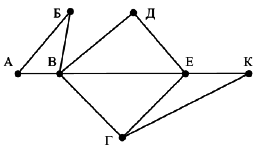
**По­яс­не­ние.**

Дан­ное вы­ра­же­ние яв­ля­ет­ся дизъ­юнк­ци­ей двух конъ­юнк­ций. Можем за­ме­тить, что в обоих сла­га­е­мых есть мно­жи­тель *x*. Т. е. при *x* = 0 сумма будет равна 0. Так, для пе­ре­мен­ной *x* под­хо­дит толь­ко тре­тий стол­бец.

В вось­мой стро­ке таб­ли­цы *x* = 1, а зна­че­ние функ­ции равно 0. Такое воз­мож­но толь­ко при *z* = 1, *у*= 0, т. е. пе­ре­мен­ная1 − *z*, а пе­ре­мен­ная2 − *y*.

Ответ: *zyx*.

**3. За­да­ние** На ри­сун­ке спра­ва схема дорог Н-ского рай­о­на изоб­ра­же­на в виде графа, в таб­ли­це со­дер­жат­ся све­де­ния о дли­нах этих дорог (в ки­ло­мет­рах).



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | П1 | П2 | П3 | П4 | П5 | П6 | П7 |
| П1 |  | 45 |  | 10 |  |  |  |
| П2 | 45 |  |  | 40 |  | 55 |  |
| П3 |  |  |  |  | 15 | 60 |  |
| П4 | 10 | 40 |  |  |  | 20 | 35 |
| П5 |  |  | 15 |  |  | 55 |  |
| П6 |  | 55 | 60 | 20 | 55 |  | 45 |
| П7 |  |  |  | 35 |  | 45 |  |

Так как таб­ли­цу и схему ри­со­ва­ли не­за­ви­си­мо друг от друга, то ну­ме­ра­ция населённых пунк­тов в таб­ли­це никак не свя­за­на с бук­вен­ны­ми обо­зна­че­ни­я­ми на графе. Опре­де­ли­те, ка­ко­ва длина до­ро­ги из пунк­та В в пункт Е. В от­ве­те за­пи­ши­те целое число – так, как оно ука­за­но в таб­ли­це.

**По­яс­не­ние.**

Пункт В − един­ствен­ный пункт с пятью до­ро­га­ми, зна­чит ему со­от­вет­ству­ет П6, а пункт Е − един­ствен­ный с че­тырь­мя до­ро­га­ми, зна­чит ему со­от­вет­ству­ет П4.

Длина до­ро­ги из П6 в П4 равна 20.

Ответ: 20.

**4. За­да­ние** В фраг­мен­те базы дан­ных пред­став­ле­ны све­де­ния о род­ствен­ных от­но­ше­ни­ях. На ос­но­ва­нии при­ведённых дан­ных опре­де­ли­те, сколь­ко пря­мых по­том­ков (т.е. детей и вну­ков) Пав­лен­ко А.К. упо­мя­ну­ты в таб­ли­це 1.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Таб­ли­ца 1** | | ID | Фа­ми­лия\_И.О. | Пол | | 2146 | Кри­вич Л. П. | Ж | | 2155 | Пав­лен­ко А. К. | М | | 2431 | Хит­рук П. А. | М | | 2480 | Кри­вич А. А. | М | | 2302 | Пав­лен­ко Е. А. | Ж | | 2500 | Сокол Н. А. | Ж | | 3002 | Пав­лен­ко И. А. | М | | 2523 | Пав­лен­ко Т. Х. | Ж | | 2529 | Хит­рук А. П. | М | | 2570 | Пав­лен­ко П. И. | М | | 2586 | Пав­лен­ко Т. И. | Ж | | 2933 | Си­мо­нян А. А. | Ж | | 2511 | Сокол В. А. | Ж | | 3193 | Биба С. А. | Ж | | ... | ... | ... | | |  | | --- | | **Таб­ли­ца 2** | | ID\_Ро­ди­те­ля | ID\_Ре­бен­ка | | 2146 | 2302 | | 2146 | 3002 | | 2155 | 2302 | | 2155 | 3002 | | 2302 | 2431 | | 2302 | 2511 | | 2302 | 3193 | | 3002 | 2586 | | 3002 | 2570 | | 2523 | 2586 | | 2523 | 2570 | | 2529 | 2431 | | 2529 | 2511 | | 2529 | 3193 | | ... | ... | |

ИЛИ

Для груп­по­вых опе­ра­ций с фай­ла­ми ис­поль­зу­ют­ся маски имён фай­лов. Маска пред­став­ля­ет собой по­сле­до­ва­тель­ность букв, цифр и про­чих до­пу­сти­мых в име­нах фай­лов сим­во­лов, в ко­то­рых также могут встре­чать­ся сле­ду­ю­щие сим­во­лы:

Сим­вол «?» (во­про­си­тель­ный знак) озна­ча­ет ровно один про­из­воль­ный сим­вол.

Сим­вол «\*» (звез­доч­ка) озна­ча­ет любую по­сле­до­ва­тель­ность сим­во­лов про­из­воль­ной длины, в том числе «\*» может за­да­вать и пу­стую по­сле­до­ва­тель­ность.

В ка­та­ло­ге на­хо­дит­ся 6 фай­лов:

maveric.map

maveric.mp3

taverna.mp4

revolver.mp4

vera.mp3

zveri.mp3

Ниже пред­став­ле­но во­семь масок. Сколь­ко из них таких, ко­то­рым со­от­вет­ству­ют ровно че­ты­ре файла из дан­но­го ка­та­ло­га?

|  |  |  |  |
| --- | --- | --- | --- |
| \*ver\*.mp\* | \*?ver?\*.mp? | ?\*ver\*.mp?\* | \*v\*r\*?.m?p\* |
| ???\*???.mp\* | ???\*???.m\* | \*a\*.\*a\* | \*a\*.\*p\* |

**По­яс­не­ние.**

Из таб­ли­цы 2 видим, что у Пав­лен­ко А. К.(ID 2155) два ре­бен­ка, их ID: 2302 и 3002.

У Пав­лен­ко Е. А.(ID 2302) трое детей, а у Пав­лен­ко И. А.(ID 3002) двое.

Таким об­ра­зом, у Пав­лен­ко А. К. се­ме­ро пря­мых по­том­ков: два ре­бен­ка и пять вну­ков.

Ответ: 7.

ИЛИ

Рас­смот­рим каж­дую маску:

1. По маске \*ver\*.mp\* будет ото­бра­но пять фай­лов:

maveric.mp3

taverna.mp4

revolver.mp4

vera.mp3

zveri.mp3

2. По маске \*?ver?\*.mp? будет ото­бра­но три файла:

maveric.mp3

taverna.mp4

zveri.mp3

3. По маске ?\*ver\*.mp?\* будет ото­бра­но че­ты­ре файла:

maveric.mp3

taverna.mp4

revolver.mp4

zveri.mp3

4. По маске \*v\*r\*?.m?p\* будет ото­бран один файл:

maveric.map

5. По маске ???\*???.mp\* будет ото­бра­но три файла:

maveric.mp3

taverna.mp4

revolver.mp4

6. По маске ???\*???.m\* будет ото­бра­но че­ты­ре файла:

maveric.map

maveric.mp3

taverna.mp4

revolver.mp4

7. По маске \*a\*.\*a\* будет ото­бран один файл:

maveric.map

8. По маске \*a\*.\*p\* будет ото­бра­но че­ты­ре файла:

maveric.map

maveric.mp3

taverna.mp4

vera.mp3

То есть три маски, ко­то­рым со­от­вет­ству­ют ровно че­ты­ре файла из дан­но­го ка­та­ло­га.

Ответ: 3.

Ответ: 7|3

**5. За­да­ние** По ка­на­лу связи пе­ре­да­ют­ся со­об­ще­ния, со­дер­жа­щие толь­ко че­ты­ре буквы: П, О, С, Т; для пе­ре­да­чи ис­поль­зу­ет­ся дво­ич­ный код, до­пус­ка­ю­щий од­но­знач­ное де­ко­ди­ро­ва­ние. Для букв Т, О, П ис­поль­зу­ют­ся такие ко­до­вые слова: Т: 111, О: 0, П: 100.

Ука­жи­те крат­чай­шее ко­до­вое слово для буквы С, при ко­то­ром код будет до­пус­кать од­но­знач­ное де­ко­ди­ро­ва­ние. Если таких кодов не­сколь­ко, ука­жи­те код с наи­мень­шим чис­ло­вым зна­че­ни­ем.

**По­яс­не­ние.**

Буква С не может ко­ди­ро­вать­ся как 0, так как 0 уже занят.

Буква С не может ко­ди­ро­вать­ся как 1, так как ко­ди­ро­ва­ние буквы Т на­чи­на­ет­ся с 1.

Буква С не может ко­ди­ро­вать­ся как 10, так как ко­ди­ро­ва­ние буквы П на­чи­на­ет­ся с 10.

Буква С не может ко­ди­ро­вать­ся как 11, так как ко­ди­ро­ва­ние буквы Т на­чи­на­ет­ся с 11.

Буква С может ко­ди­ро­вать­ся как 101 − это наи­мень­шее воз­мож­ное зна­че­ние.

Ответ: 101.

**6. За­да­ние** На вход ал­го­рит­ма подаётся на­ту­раль­ное число N. Ал­го­ритм стро­ит по нему новое число R сле­ду­ю­щим об­ра­зом.

1. Стро­ит­ся дво­ич­ная за­пись числа N.

2. К этой за­пи­си до­пи­сы­ва­ют­ся спра­ва ещё два раз­ря­да по сле­ду­ю­ще­му пра­ви­лу:

    а) скла­ды­ва­ют­ся все цифры дво­ич­ной за­пи­си, и оста­ток от де­ле­ния суммы на 2 до­пи­сы­ва­ет­ся в конец числа (спра­ва). На­при­мер, за­пись 11100 пре­об­ра­зу­ет­ся в за­пись 111001;

      б) над этой за­пи­сью про­из­во­дят­ся те же дей­ствия – спра­ва до­пи­сы­ва­ет­ся оста­ток от де­ле­ния суммы цифр на 2.

По­лу­чен­ная таким об­ра­зом за­пись (в ней на два раз­ря­да боль­ше, чем в за­пи­си ис­ход­но­го числа N) яв­ля­ет­ся дво­ич­ной за­пи­сью ис­ко­мо­го числа R.

Ука­жи­те такое наи­мень­шее число N, для ко­то­ро­го ре­зуль­тат ра­бо­ты ал­го­рит­ма боль­ше 125. В от­ве­те это число за­пи­ши­те в де­ся­тич­ной си­сте­ме счис­ле­ния.

ИЛИ

У ис­пол­ни­те­ля Каль­ку­ля­тор две ко­ман­ды, ко­то­рым при­сво­е­ны но­ме­ра:

**1. при­бавь 2,**

**2. умножь на 5.**

Вы­пол­няя первую из них, Каль­ку­ля­тор при­бав­ля­ет к числу на экра­не 2, а вы­пол­няя вто­рую, умно­жа­ет его на 5.

На­при­мер, про­грам­ма 2121 – это про­грам­ма

**умножь на 5,**

**при­бавь 2,**

**умножь на 5,**

**при­бавь 2,**

ко­то­рая пре­об­ра­зу­ет число 1 в число 37.

За­пи­ши­те по­ря­док ко­манд в про­грам­ме, ко­то­рая пре­об­ра­зу­ет число 2 в число 24 и со­дер­жит не более четырёх ко­манд. Ука­зы­вай­те лишь но­ме­ра ко­манд.

**По­яс­не­ние.**

Дан­ный ал­го­ритм при­пи­сы­ва­ет в конце числа или 10, если из­на­чаль­но в его дво­ич­ной за­пи­си было не­чет­ное ко­ли­че­ство еди­ниц, или 00 если чет­ное.

12610 = 11111102 может по­лу­чить­ся в ре­зуль­та­те ра­бо­ты ал­го­рит­ма из числа 111112.

111112 = 3110.

Ответ: 31.

ИЛИ

Решим за­да­чу от об­рат­но­го, а потом за­пи­шем по­лу­чен­ные ко­ман­ды спра­ва на­ле­во.

Если число не де­лит­ся на 5, тогда по­лу­че­но через ко­ман­ду 1, если де­лит­ся, то через ко­ман­ду 2.

22 + 2 = 24(ко­ман­да 1)

20 + 2 = 22(ко­ман­да 1)

4 \* 5 = 20(ко­ман­да 2)

2 + 2 = 4(ко­ман­да 1)

Ответ: 1211.

Ответ: 31|1211

**7. За­да­ние.** Дан фраг­мент элек­трон­ной таб­ли­цы. Из ячей­ки E4 в ячей­ку D3 была ско­пи­ро­ва­на фор­му­ла. При ко­пи­ро­ва­нии ад­ре­са ячеек в фор­му­ле ав­то­ма­ти­че­ски из­ме­ни­лись. Каким стало чис­ло­вое зна­че­ние фор­му­лы в ячей­ке D3?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** |
| 1 | 40 | 4 | 400 | 70 | 7 |
| 2 | 30 | 3 | 300 | 60 | 6 |
| 3 | 20 | 2 | 200 |  | 5 |
| 4 | 10 | 1 | 100 | 40 | =$B2 \* C$3 |

При­ме­ча­ние: знак $ обо­зна­ча­ет аб­со­лют­ную ад­ре­са­цию.

ИЛИ



Дан фраг­мент элек­трон­ной таб­ли­цы.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **A** | **B** | **C** |
| 1 |  | 6 | 10 |
| 2 | =(A1-3)/(B1-1) | =(A1-3)/(C1-5) | = C1/(A1 – 3) |

Какое целое число долж­но быть за­пи­са­но в ячей­ке A1, чтобы диа­грам­ма, по­стро­ен­ная по зна­че­ни­ям ячеек диа­па­зо­на A2:С2, со­от­вет­ство­ва­ла ри­сун­ку? Из­вест­но, что все зна­че­ния ячеек из рас­смат­ри­ва­е­мо­го диа­па­зо­на не­от­ри­ца­тель­ны.

**По­яс­не­ние.**

Фор­му­ла, при ко­пи­ро­ва­нии в ячей­ку D3 из­ме­ни­лась на =$B1 \* B$3.

B1 \* B3 = 4 \* 2 = 8.

Ответ: 8.

ИЛИ

Под­ста­вим зна­че­ния B1 и C1 в фор­му­лы A2:C2:

A2 = (A1-3)/5

B2 = (A1-3)/5

C2 = 10/(A1-3)

Так как A2 = B2, то С2 = 2 \* A2 = 2 \* B2

Под­ста­вим:

10/(A1-3) = 2\*(A1-3)/5

A1 - 3 = 5

A1 = 8.

Ответ: 8.

**8. За­да­ние** За­пи­ши­те число, ко­то­рое будет на­пе­ча­та­но в ре­зуль­та­те вы­пол­не­ния сле­ду­ю­щей про­грам­мы. Для Ва­ше­го удоб­ства про­грам­ма пред­став­ле­на на пяти язы­ках про­грам­ми­ро­ва­ния.

|  |  |
| --- | --- |
| **Бей­сик** | **Python** |
| DIM S, N AS INTEGER  S = 0  N = 0  WHILE S < 111  S = S + 8  N = N + 2  WEND  PRINT N | s = 0  n = 0  while s < 111:  s = s + 8  n = n + 2  print(n) |
| **Ал­го­рит­ми­че­ский язык** | **Пас­каль** |
| алг  нач  цел n, s  n := 0  s := 0  нц пока s < 111  s := s + 8  n := n + 2  кц  вывод n  кон | var s, n: integer;  begin  s := 0;  n := 0;  while s < 111 do  begin  s := s + 8;  n := n + 2  end;  writeln(n)  end. |
| **Си** | |
| #include  int main()  { int s = 0, n = 0;  while (s < 111) { s = s + 8; n = n + 2; }  printf("%d\n", n);  return 0;  } | |

**По­яс­не­ние.**

Цикл while вы­пол­ня­ет­ся до тех пор, пока ис­тин­но усло­вие s < 111, т. е. пе­ре­мен­ная s опре­де­ля­ет, сколь­ко раз вы­пол­нит­ся цикл. По­сколь­ку из­на­чаль­но s = 0, цикл вы­пол­нит­ся 14 раз, сле­до­ва­тель­но, n = 2 \* 14 = 28.

Ответ: 28.

**9. За­да­ние.** Какой ми­ни­маль­ный объём па­мя­ти (в Кбайт) нужно за­ре­зер­ви­ро­вать, чтобы можно было со­хра­нить любое раст­ро­вое изоб­ра­же­ние раз­ме­ром 64×64 пик­се­лов при усло­вии, что в изоб­ра­же­нии могут ис­поль­зо­вать­ся 256 раз­лич­ных цве­тов? В от­ве­те за­пи­ши­те толь­ко целое число, еди­ни­цу из­ме­ре­ния пи­сать не нужно.

ИЛИ

Му­зы­каль­ный фраг­мент был за­пи­сан в фор­ма­те моно, оциф­ро­ван и со­хранён в виде файла без ис­поль­зо­ва­ния сжа­тия дан­ных. Раз­мер по­лу­чен­но­го файла – 24 Мбайт. Затем тот же му­зы­каль­ный фраг­мент был за­пи­сан по­втор­но в фор­ма­те сте­рео (двух­ка­наль­ная за­пись) и оциф­ро­ван с раз­ре­ше­ни­ем в 4 раза выше и ча­сто­той дис­кре­ти­за­ции в 1,5 раза мень­ше, чем в пер­вый раз. Сжа­тие дан­ных не про­из­во­ди­лось. Ука­жи­те раз­мер файла в Мбайт, по­лу­чен­но­го при по­втор­ной за­пи­си. В от­ве­те за­пи­ши­те толь­ко целое число, еди­ни­цу из­ме­ре­ния пи­сать не нужно.

**По­яс­не­ние.**

Один пик­сель ко­ди­ру­ет­ся 8 би­та­ми па­мя­ти.

Всего 64 \* 64 = 212 пик­се­лей.

Объем па­мя­ти, за­ни­ма­е­мый изоб­ра­же­ни­ем 212 \* 8 = 215 бит = 212 байт = 4 Кбайт.

Ответ: 4.

ИЛИ

При за­пи­си того же файла в сте­рео фор­ма­те его объем уве­ли­чи­ва­ет­ся в 2 раза. 24 \* 2 = 48

При уве­ли­че­нии его раз­ре­ше­ния в 4 раза его объем также уве­ли­чи­ва­ет­ся в 4 раза. 48 \* 4 = 192

При умень­ше­нии ча­сто­ты дис­кре­ти­за­ции в 1,5 раза его объем умень­ша­ет­ся в 1,5 раза. 192 / 1,5 = 128.

Ответ: 128.

Ответ: 4|128

**10. За­да­ние** Игорь со­став­ля­ет таб­ли­цу ко­до­вых слов для пе­ре­да­чи со­об­ще­ний, каж­до­му со­об­ще­нию со­от­вет­ству­ет своё ко­до­вое слово. В ка­че­стве ко­до­вых слов Игорь ис­поль­зу­ет 5-бук­вен­ные слова, в ко­то­рых есть толь­ко буквы П, И, Р, причём буква П по­яв­ля­ет­ся ровно 1 раз. Каж­дая из дру­гих до­пу­сти­мых букв может встре­чать­ся в ко­до­вом слове любое ко­ли­че­ство раз или не встре­чать­ся со­всем. Сколь­ко раз­лич­ных ко­до­вых слов может ис­поль­зо­вать Игорь?

**По­яс­не­ние.**

Игорь может со­ста­вить 24 слов по­ста­вив букву П на пер­вое место. Ана­ло­гич­но можно по­ста­вить ее на вто­рое, тре­тье, чет­вер­тое и пятое место. По­лу­чим 5 \* 24 = 80 слов.

Ответ: 80.

**11. За­да­ние** Ниже на пяти язы­ках про­грам­ми­ро­ва­ния за­пи­са­ны две ре­кур­сив­ные функ­ции (про­це­ду­ры): F и G.

|  |  |
| --- | --- |
| **Бей­сик** | **Python** |
| DECLARE SUB F(n)  DECLARE SUB G(n)    SUB F(n)      IF n > 0 THEN G(n - 1)  END SUB    SUB G(n)      PRINT "\*"      IF n > 1 THEN F(n - 3)  END SUB | def F(n):      if n > 0:          G(n - 1)  def G(n):      print("\*")      if n > 1:          F(n - 3) |
| **Ал­го­рит­ми­че­ский язык** | **Пас­каль** |
| алг F(цел n)  нач      если n > 0 то          G(n - 1)      все  кон    алг G(цел n)  нач      вывод "\*"      если n > 1 то          F(n - 3)      все  кон | procedure F(n: integer); forward;  procedure G(n: integer); forward;    procedure F(n: integer);  begin      if n > 0 then          G(n - 1);  end;    procedure G(n: integer);  begin      writeln('\*');      if n > 1 then          F(n - 3);  end; |
| **Си** | |
| void F(int n);  void G(int n);    void F(int n){      if (n > 0)          G(n - 1);  }    void G(int n){      printf("\*");      if (n > 1)          F(n - 3);  } | |

Сколь­ко сим­во­лов «звёздоч­ка» будет на­пе­ча­та­но на экра­не при вы­пол­не­нии вы­зо­ва F(11)?

**По­яс­не­ние.**

Про­мо­де­ли­ру­ем ра­бо­ту про­грам­мы:

F(11)

G(10): \*

F(7)

G(6): \*

F(3)

G(2): \*

F(-1)

Ответ: 3.

**12. За­да­ние** В тер­ми­но­ло­гии сетей TCP/IP мас­кой сети на­зы­ва­ет­ся дво­ич­ное число, опре­де­ля­ю­щее, какая часть IP-ад­ре­са узла сети от­но­сит­ся к ад­ре­су сети, а какая – к ад­ре­су са­мо­го узла в этой сети. Обыч­но маска за­пи­сы­ва­ет­ся по тем же пра­ви­лам, что и IP-адрес, – в виде четырёх бай­тов, причём каж­дый байт за­пи­сы­ва­ет­ся в виде де­ся­тич­но­го числа. При этом в маске сна­ча­ла (в стар­ших раз­ря­дах) стоят еди­ни­цы, а затем с не­ко­то­ро­го раз­ря­да – нули. Адрес сети по­лу­ча­ет­ся в ре­зуль­та­те при­ме­не­ния по­раз­ряд­ной конъ­юнк­ции к за­дан­но­му IP-ад­ре­су узла и маске.

На­при­мер, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-ад­ре­сом 111.81.208.27 адрес сети равен 111.81.192.0. Чему равно наи­мень­шее воз­мож­ное зна­че­ние тре­тье­го слева байта маски? Ответ за­пи­ши­те в виде де­ся­тич­но­го числа.

**По­яс­не­ние.**

За­пи­шем тре­тий байт IP-ад­ре­са и ад­ре­са сети в дво­ич­ной си­сте­ме счис­ле­ния:

20810 = 110100002

19210 = 110000002

Видим, что два пер­вых слева бита маски − еди­ни­цы, зна­чит, чтобы зна­че­ние было наи­мень­шим, осталь­ные биты долж­ны быть ну­ля­ми. По­лу­ча­ем, что тре­тий слева байт маски равен 110000002 = 19210

Ответ: 192.

**13. За­да­ние** При ре­ги­стра­ции в ком­пью­тер­ной си­сте­ме каж­до­му поль­зо­ва­те­лю выдаётся па­роль, со­сто­я­щий из 15 сим­во­лов и со­дер­жа­щий толь­ко сим­во­лы из 12-сим­воль­но­го на­бо­ра: А, В, C, D, Е, F, G, H, K, L, M, N. В базе дан­ных для хра­не­ния све­де­ний о каж­дом поль­зо­ва­те­ле от­ве­де­но оди­на­ко­вое и ми­ни­маль­но воз­мож­ное целое число байт. При этом ис­поль­зу­ют по­сим­воль­ное ко­ди­ро­ва­ние па­ро­лей, все сим­во­лы ко­ди­ру­ют оди­на­ко­вым и ми­ни­маль­но воз­мож­ным ко­ли­че­ством бит. Кроме соб­ствен­но па­ро­ля, для каж­до­го поль­зо­ва­те­ля в си­сте­ме хра­нят­ся до­пол­ни­тель­ные све­де­ния, для чего вы­де­ле­но целое число байт; это число одно и то же для всех поль­зо­ва­те­лей. Для хра­не­ния све­де­ний о 20 поль­зо­ва­те­лях по­тре­бо­ва­лось 400 байт. Сколь­ко байт вы­де­ле­но для хра­не­ния до­пол­ни­тель­ных све­де­ний об одном поль­зо­ва­те­ле? В от­ве­те за­пи­ши­те толь­ко целое число – ко­ли­че­ство байт.

**По­яс­не­ние.**

Со­глас­но усло­вию, в но­ме­ре могут быть ис­поль­зо­ва­ны 12 букв. Из­вест­но, что с по­мо­щью N бит можно за­ко­ди­ро­вать 2N раз­лич­ных ва­ри­ан­тов. По­сколь­ку 23 < 12 < 24, то для за­пи­си каж­до­го из 12 сим­во­лов не­об­хо­ди­мо 4 бита.

Для хра­не­ния всех 15 сим­во­лов па­ро­ля нужно 4 · 15 = 60 бит, а т. к. для за­пи­си ис­поль­зу­ет­ся целое число байт, то берём бли­жай­шее не мень­шее зна­че­ние, крат­ное вось­ми, это число 64 = 8 · 8 бит (8 байт).

Пусть ко­ли­че­ство па­мя­ти, от­ве­ден­ное под до­пол­ни­тель­ные се­де­ния равно *x*, тогда:

20 \* (8+*x*) = 400

*x* = 12

Ответ: 12.

**14. За­да­ние** Ис­пол­ни­тель Ре­дак­тор по­лу­ча­ет на вход стро­ку цифр и пре­об­ра­зо­вы­ва­ет её. Ре­дак­тор может вы­пол­нять две ко­ман­ды, в обеих ко­ман­дах v и w обо­зна­ча­ют це­поч­ки цифр.

А) **за­ме­нить** (v, w).

Эта ко­ман­да за­ме­ня­ет в стро­ке пер­вое слева вхож­де­ние це­поч­ки v на це­поч­ку w. На­при­мер, вы­пол­не­ние ко­ман­ды

**за­ме­нить** (111, 27)

пре­об­ра­зу­ет стро­ку 05111150 в стро­ку 0527150. Если в стро­ке нет вхож­де­ний це­поч­ки v, то вы­пол­не­ние ко­ман­ды за­ме­нить (v, w) не ме­ня­ет эту стро­ку.

Б) **на­шлось** (v).

Эта ко­ман­да про­ве­ря­ет, встре­ча­ет­ся ли це­поч­ка v в стро­ке ис­пол­ни­те­ля Ре­дак­тор. Если она встре­ча­ет­ся, то ко­ман­да воз­вра­ща­ет ло­ги­че­ское зна­че­ние «ис­ти­на», в про­тив­ном слу­чае воз­вра­ща­ет зна­че­ние «ложь». Стро­ка

ис­пол­ни­те­ля при этом не из­ме­ня­ет­ся.

 Цикл

    ПОКА усло­вие

        по­сле­до­ва­тель­ность ко­манд

    КОНЕЦ ПОКА

 вы­пол­ня­ет­ся, пока усло­вие ис­тин­но.

 В кон­струк­ции

    ЕСЛИ усло­вие

        ТО ко­ман­да1

        ИНАЧЕ ко­ман­да2

    КОНЕЦ ЕСЛИ

 вы­пол­ня­ет­ся ко­ман­да1 (если усло­вие ис­тин­но) или ко­ман­да2 (если усло­вие ложно).

Какая стро­ка по­лу­чит­ся в ре­зуль­та­те при­ме­не­ния при­ведённой ниже

про­грам­мы к стро­ке, со­сто­я­щей из 68 иду­щих под­ряд цифр 8? В от­ве­те

за­пи­ши­те по­лу­чен­ную стро­ку.

 НА­ЧА­ЛО

 ПОКА **на­шлось** (222) ИЛИ **на­шлось** (888)

    ЕСЛИ **на­шлось** (222)

        ТО **за­ме­нить** (222, 8)

        ИНАЧЕ **за­ме­нить** (888, 2)

    КОНЕЦ ЕСЛИ

 КОНЕЦ ПОКА

 КОНЕЦ

**По­яс­не­ние.**

В 68 иду­щих под­ряд циф­рах 8 22 груп­пы по три вось­мер­ки, ко­то­рые за­ме­нят­ся на 22 двой­ки и оста­нут­ся две вось­мер­ки.

68(8) = 22(2) + 2(8)

22(2) + 2(8) = 1(2) + 9(8)

1(2) + 9(8) = 4(2)

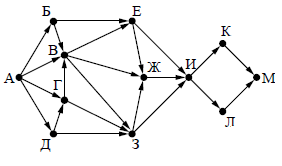
4(2) = 1(2) + 1(8) = 28

Ответ: 28.

**15. За­да­ние** ри­сун­ке пред­став­ле­на схема дорог, свя­зы­ва­ю­щих го­ро­да А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М.

По каж­дой до­ро­ге можно дви­гать­ся толь­ко в одном на­прав­ле­нии, ука­зан­ном стрел­кой.

Сколь­ко су­ще­ству­ет раз­лич­ных путей из го­ро­да А в город М?



**По­яс­не­ние.**

Нач­нем счи­тать ко­ли­че­ство путей с конца марш­ру­та — с го­ро­да М. Пусть NX — ко­ли­че­ство раз­лич­ных путей из го­ро­да А в город X, N — общее число путей. В город М можно при­е­хать из Л или K, по­это­му N = NМ = NЛ + NК. (\*)

Ана­ло­гич­но:

NК = NИ;

NЛ = NИ;

NИ = NЕ + NЖ + NЗ

NК = NЕ = 1.

До­ба­вим еще вер­ши­ны:

NБ = NA = 1;

NВ = NБ + NА + NГ = 1 + 1 + 1 = 3;

NЕ = NГ = 1;

NГ = NA = 1.

Под­ста­вим в фор­му­лу (\*): N = NM = 4 + 4 + 4 + 1 = 13.

Ответ: 13.

Ответ: 56

**16. За­да­ние** Зна­че­ние ариф­ме­ти­че­ско­го вы­ра­же­ния: 98 + 35 – 9 – за­пи­са­ли в си­стем счис­ле­ния с ос­но­ва­ни­ем 3. Сколь­ко цифр «2» со­дер­жит­ся в этой за­пи­си?

**По­яс­не­ние.**

Пре­об­ра­зу­ем вы­ра­же­ние:

(32)8 + 35 - 32

316 + 35 - 32

316 + 35 = 100...00100000

100...00100000 - 32 = 100...00022200

В по­лу­чен­ном числе три двой­ки.

Ответ: 3

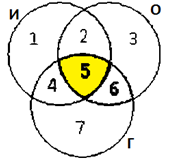
**17. За­да­ние** В языке за­про­сов по­ис­ко­во­го сер­ве­ра для обо­зна­че­ния ло­ги­че­ской опе­ра­ции «ИЛИ» ис­поль­зу­ет­ся сим­вол «|», а для обо­зна­че­ния ло­ги­че­ской опе­ра­ции «И» – сим­вол «&». В таб­ли­це при­ве­де­ны за­про­сы и ко­ли­че­ство най­ден­ных по ним стра­ниц не­ко­то­ро­го сег­мен­та сети Ин­тер­нет.

|  |  |
| --- | --- |
| **За­прос** | **Най­де­но стра­ниц (в ты­ся­чах)** |
| Гомер & Или­а­да | 200 |
| Гомер & (Одис­сея | Или­а­да) | 470 |
| Гомер & Одис­сея | 355 |

Какое ко­ли­че­ство стра­ниц (в ты­ся­чах) будет най­де­но по за­про­су *Гомер & Одис­сея & Или­а­да?* Счи­та­ет­ся, что все за­про­сы вы­пол­ня­лись прак­ти­че­ски од­но­вре­мен­но, так что набор стра­ниц, со­дер­жа­щих все ис­ко­мые слова, не из­ме­нял­ся за время

вы­пол­не­ния за­про­сов.

**По­яс­не­ние.**



Ко­ли­че­ство за­про­сов в дан­ной об­ла­сти будем обо­зна­чать Ni. Наша цель — N5.

Тогда из таб­ли­цы на­хо­дим, что:

N5 + N6 = 355,

N4 + N5 = 200,

N4 + N5 + N6 = 470.

Из пер­во­го и вто­ро­го урав­не­ния: N4 + 2N5 + N6 = 555.

Из по­след­не­го урав­не­ния: N5 = 85.

Ответ: 85

**18. За­да­ние** Обо­зна­чим через *m&n* по­раз­ряд­ную конъ­юнк­цию не­от­ри­ца­тель­ных целых чисел *m* и *n*. Так, на­при­мер, 14&5 = 11102&01012 = 01002 = 4.

Для ка­ко­го наи­мень­ше­го не­от­ри­ца­тель­но­го це­ло­го числа *А* фор­му­ла

*x&25 ≠ 0 → (x&17 = 0 → x&А ≠ 0)*

тож­де­ствен­но ис­тин­на (т.е. при­ни­ма­ет зна­че­ние 1 при любом не­от­ри­ца­тель­ном целом зна­че­нии пе­ре­мен­ной *х*)?

**По­яс­не­ние.**

Вве­дем обо­зна­че­ния:

(x ∈ А) ≡ A; (x ∈ P) ≡ P; (x ∈ Q) ≡ Q.

Пре­об­ра­зо­вав, по­лу­ча­ем:

¬P ∨ ¬(Q ∧ ¬A) ∨ ¬P = ¬P ∨ ¬Q ∨ A.

Ло­ги­че­ское ИЛИ ис­тин­но, если ис­тин­но хотя бы одно утвер­жде­ние. Усло­вию ¬P ∨ ¬Q = 1 удо­вле­тво­ряют лучи (−∞, 40) и (60, ∞). По­сколь­ку вы­ра­же­ние ¬P ∨ ¬Q ∨ A долж­но быть тож­де­ствен­но ис­тин­ным, вы­ра­же­ние A долж­но быть ис­тин­но на от­рез­ке [40, 60]. Его длина равна 20.

Ответ: 20.

Ответ: 8

**19. За­да­ние** В про­грам­ме ис­поль­зу­ет­ся од­но­мер­ный це­ло­чис­лен­ный мас­сив A с ин­дек­са­ми от 0 до 9. Зна­че­ния эле­мен­тов равны 4, 7, 3, 8, 5, 0, 1, 2, 9, 6 со­от­вет­ствен­но, т.е. A[0] = 4, A[1] = 7 и т.д.

Опре­де­ли­те зна­че­ние пе­ре­мен­ной *c* после вы­пол­не­ния сле­ду­ю­ще­го фраг­мен­та этой про­грам­мы *(за­пи­сан­но­го ниже на пяти язы­ках про­грам­ми­ро­ва­ния)*.

|  |  |
| --- | --- |
| **Бей­сик** | **Python** |
| c = 0   FOR i = 1 TO 9      IF A(i) < A(0) THEN          c = c + 1          t = A(i)          A(i) = A(0)          A(0) = t      ENDIF   NEXT i | c = 0   for i in range(1,10):      if A[i] < A[0]:          c = c + 1          t = A[i]          A[i] = A[0]          A[0] = t |
| **Ал­го­рит­ми­че­ский язык** | **Пас­каль** |
| c := 0   нц для i от 1 до 9      если A[i] < A[0] то          c := c + 1          t := A[i]          A[i] := A[0]          A[0] := t      все   кц | c := 0;   for i := 1 to 9 do      if A[i] < A[0] then      begin          c := c + 1;          t := A[i];          A[i] := A[0];          A[0] := t;      end; |
| **Си** | |
| c = 0;   for (i = 1;i < 10;i++)      if (A[i] < A[0])      {          c++;          t = A[i];          A[i] = A[0];          A[0] = t;      } | |

**По­яс­не­ние.**

Если A[i] эле­мент мас­си­ва мень­ше A[0], то про­грам­ма ме­ня­ет их ме­ста­ми и уве­ли­чи­ва­ет зна­че­ние пе­ре­мен­ной *c* на 1. Про­грам­ма вы­пол­нит­ся два­жды, пер­вый раз по­ме­няв ме­ста­ми A[0] и A[2], так как 3<4, и вто­рой раз по­ме­няв A[0] и A[5] (0<3). Таким об­ра­зом зна­че­ние пе­ре­мен­ной *с* ста­нет равно 2.

Ответ: 2.

**20. За­да­ние** Ниже на пяти язы­ках про­грам­ми­ро­ва­ния за­пи­сан ал­го­ритм. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет число *M*. Из­вест­но, что *x* > 100. Ука­жи­те наи­мень­шее такое (т.е. боль­шее 100) число *x*, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет 26.

|  |  |
| --- | --- |
| **Бей­сик** | **Python** |
| DIM X, L, M AS INTEGER  INPUT X  L = X  M = 65  IF L MOD 2 = 0 THEN      M = 52  ENDIF  WHILE L <> M  IF L > M THEN      L = L – M  ELSE      M = M – L  ENDIF  WEND  PRINT M | x = int(input())  L = x  M = 65  if L % 2 == 0:      M = 52  while L != M:      if L > M:          L = L - M      else:          M = M - L  print(M) |
| **Ал­го­рит­ми­че­ский язык** | **Пас­каль** |
| алг  нач      цел x, L, M      ввод x      L := x      M := 65      если mod(L,2)=0          то              M := 52      все      нц пока L <> M          если L > M              то                  L := L – M              иначе                  M := M – L          все      кц      вывод M  кон | var x, L, M: integer;  begin      readln(x);      L := x;      M := 65;      if L mod 2 = 0 then          M := 52;      while L <> M do          if L > M then              L := L - M          else              M := M – L;      writeln(M);  end. |
| **Си** | |
| #include  void main()  {      int x, L, M;      scanf("%d", &x);      L = x;      M = 65;      if (L % 2 == 0)          M = 52;      while (L != M){          if(L > M)              L = L - M;          else              M = M - L;      }      printf("%d", M);  } | |

**По­яс­не­ние.**

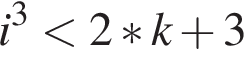
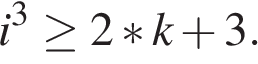
В теле цикла числа M и L умень­ша­ют­ся, пока не ста­нут рав­ны­ми. Чтобы в итоге было на­пе­ча­та­но 26, оба числа в какой-то мо­мент долж­ны быть равны 26. Пой­дем от конца к на­ча­лу: на преды­ду­щем шаге одно число было 26, а дру­гое 26 + 26 = 52. Еще на шаг рань­ше 52 + 26 = 78 и 52. До того 78 + 52 = 130 и 52. То есть наи­мень­шее воз­мож­ное число 130. А по­сколь­ку най­ден­ное число чет­ное, то M будет при­сво­е­но зна­че­ние 52, что и при­ве­дет к не­об­хо­ди­мо­му ре­зуль­та­ту.

Ответ: 130.

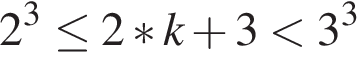
**21. За­да­ние** На­пи­ши­те в от­ве­те наи­мень­шее зна­че­ние вход­ной пе­ре­мен­ной *k*, при ко­то­ром про­грам­ма выдаёт тот же ответ, что и при вход­ном зна­че­нии *k* = 10. Для Ва­ше­го удоб­ства про­грам­ма при­ве­де­на на пяти язы­ках про­грам­ми­ро­ва­ния.

|  |  |
| --- | --- |
| **Бей­сик** | **Python** |
| DIM K, I AS LONG  INPUT K  I = 1  WHILE F(I) < G(K)      I = I + 1  WEND  PRINT I    FUNCTION F(N)      F = N \* N \* N  END FUNCTION    FUNCTION G(N)      G = 2\*N + 3  END FUNCTION | def f(n):      return n\*n\*n    def g(n):      return 2\*n+3    k = int(input())  i = 1  while f(i) < g(k):      i+=1  print (i) |
| **Ал­го­рит­ми­че­ский язык** | **Пас­каль** |
| алг  нач      цел i, k      ввод k      i := 1      нц пока f(i) < g(k)          i := i + 1      кц      вывод i  кон    алг цел f(цел n)  нач      знач := n \* n \* n  кон    алг цел g(цел n)  нач      знач := 2\*n + 3  кон | var      k, i : longint;    function f(n: longint): longint;  begin      f := n \* n \* n;  end;    function g(n: longint): longint;  begin      g := 2\*n + 3;  end;    begin      readln(k);      i := 1;      while f(i) < g(k) do          i := i+1;      writeln(i)  end. |
| **Си** | |
| #include  long f(long n) {      return n \* n \* n;  }    long g(long n) {      return 2\*n + 3;  }    int main()  {      long k, i;      scanf("%ld", &k);      i = 1;      while(f(i)<g(k))          i++;</g(k))      printf("%ld", i);      return 0;  } | |

**По­яс­не­ние.**

Дан­ная про­грам­ма срав­ни­ва­ет http://reshuege.ru/formula/c6/c61a8cd66b598b11464e3f22ca9a79e7p.png и http://reshuege.ru/formula/81/810670a23425a8f38928d01d2b38c34bp.png и при­бав­ля­ет к *i* еди­ни­цу до тех пор, пока . И вы­во­дит пер­вое зна­че­ние пе­ре­мен­ной *i* при ко­то­ром 

При k = 10, про­грам­ма вы­ве­дет число 3.

За­пи­шем не­ра­вен­ство:  от­сю­да по­лу­чим, что наи­мень­шее зна­че­ние *k* = 3.

Ответ: 3.

**22. За­да­ние** Ис­пол­ни­тель Май15 пре­об­ра­зу­ет число на экра­не. У ис­пол­ни­те­ля есть две ко­ман­ды, ко­то­рым при­сво­е­ны но­ме­ра:

**1. При­ба­вить 1**

**2. Умно­жить на 2**

Пер­вая ко­ман­да уве­ли­чи­ва­ет число на экра­не на 1, вто­рая умно­жа­ет его на 2. Про­грам­ма для ис­пол­ни­те­ля Май15 – это по­сле­до­ва­тель­ность ко­манд. Сколь­ко су­ще­ству­ет про­грамм, для ко­то­рых при ис­ход­ном числе 2 ре­зуль­та­том яв­ля­ет­ся число 29 и при этом тра­ек­то­рия вы­чис­ле­ний со­дер­жит число 14 и не со­дер­жит числа 25?

Тра­ек­то­рия вы­чис­ле­ний про­грам­мы – это по­сле­до­ва­тель­ность ре­зуль­та­тов

вы­пол­не­ния всех ко­манд про­грам­мы. На­при­мер, для про­грам­мы 121 при ис­ход­ном числе 7 тра­ек­то­рия будет со­сто­ять из чисел 8, 16, 17.

**По­яс­не­ние.**

Для сло­же­ния спра­вед­лив пе­ре­ме­сти­тель­ный (ком­му­та­тив­ный) закон, зна­чит, по­ря­док ко­манд в про­грам­ме не имеет зна­че­ния для ре­зуль­та­та.

Все ко­ман­ды уве­ли­чи­ва­ют ис­ход­ное число, по­это­му ко­ли­че­ство ко­манд не может пре­вос­хо­дить (30 − 21) = 9. При этом ми­ни­маль­ное ко­ли­че­ство ко­манд — 3.

Таким об­ра­зом, ко­манд может быть 3, 4, 5, 6, 7, 8 или 9. По­это­му по­ря­док ко­манд не имеет зна­че­ния, каж­до­му числу ко­манд со­от­вет­ству­ет один набор ко­манд, ко­то­рые можно рас­по­ло­жить в любом по­ряд­ке.

Рас­смот­рим все воз­мож­ные на­бо­ры и вы­чис­лим ко­ли­че­ство ва­ри­ан­тов рас­спо­ло­же­ния ко­манд в них. Набор 133 имеет 3 воз­мож­ных ва­ри­антов рас­по­ло­же­ния. На­бо­р 1223 — 12 воз­мож­ных ва­ри­ан­тов рас­по­ло­же­ния: это число пе­ре­ста­но­вок с по­вто­ре­ни­я­ми (1+2+1)!/(1! · 2! · 1!)). Набор 12222 — 5 ва­ри­ан­тов. Набор 111222 — 20 воз­мож­ных ва­ри­ан­тов. Набор 11123 — 20 ва­ри­ан­тов. Набор 111113 — 6 ва­ри­ан­тов, набор 1111122 — 21 ва­ри­ан­т, набор 11111112 — 8 ва­ри­ан­тов, набор 111111111 — один ва­ри­ант.

Всего имеем 3 + 12 + 5 + 20 + 20 + 6 + 21 + 8 + 1 = 96 про­грамм.

Ответ: 96.

Ответ: 96.

Ответ: 13

**23. За­да­ние** Сколь­ко су­ще­ству­ет раз­лич­ных на­бо­ров зна­че­ний ло­ги­че­ских пе­ре­мен­ных*x1, x2, ... x9, y1, y2, ... y9*, ко­то­рые удо­вле­тво­ря­ют всем пе­ре­чис­лен­ным ниже усло­ви­ям?

(¬ (*x*1 ≡ *y*1)) ≡ (*x*2 ≡ *y*2)

(¬ (*x*2 ≡ *y*2)) ≡ (*x*3 ≡ *y*3)

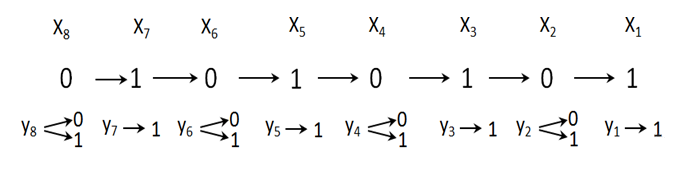
      …

(¬ (*x*8 ≡ *y*8)) ≡ (*x*9 ≡ *y*9)

В от­ве­те не нужно пе­ре­чис­лять все раз­лич­ные на­бо­ры зна­че­ний пе­ре­мен­ных *x1, x2, ... x9, y1, y2, ... y9*, при ко­то­рых вы­пол­не­на дан­ная си­сте­ма ра­венств. В ка­че­стве от­ве­та Вам нужно ука­зать ко­ли­че­ство таких на­бо­ров.

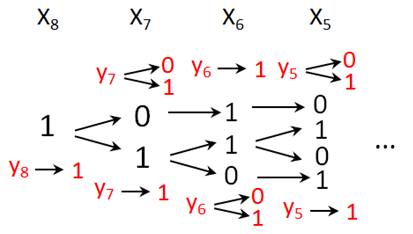
**По­яс­не­ние.**

Из по­след­не­го урав­не­ния на­хо­дим, что воз­мож­ны три ва­ри­ан­та зна­че­ний x8 и y8: 01, 00, 11. По­стро­им древо ва­ри­ан­тов для пер­вой и вто­рой пар зна­че­ний.



Таким об­ра­зом, имеем 16 на­бо­ров пе­ре­мен­ных.

Де­ре­во ва­ри­ан­тов для пары зна­че­ний 11:



По­лу­ча­ем 45 ва­ри­ан­тов. Таким об­ра­зом, си­сте­ма будет иметь 45 + 16 = 61 раз­лич­ных на­бо­ров ре­ше­ний.

Ответ: 61.

Ответ: 1024

**24. За­да­ние** На об­ра­бот­ку по­сту­па­ет по­ло­жи­тель­ное целое число, не пре­вы­ша­ю­щее 109. Нужно на­пи­сать про­грам­му, ко­то­рая вы­во­дит на экран сумму цифр этого числа, мень­ших 7. Если в числе нет цифр, мень­ших 7, тре­бу­ет­ся на экран вы­ве­сти 0. Про­грам­мист на­пи­сал про­грам­му не­пра­виль­но. Ниже эта про­грам­ма для Ва­ше­го удоб­ства при­ве­де­на на пяти язы­ках про­грам­ми­ро­ва­ния.

|  |  |
| --- | --- |
| **Бей­сик** | **Python** |
| DIM N, DIGIT, SUM AS LONG  INPUT N  SUM = 0  WHILE N > 0      DIGIT = N MOD 10      IF DIGIT < 7 THEN          SUM = SUM + 1      END IF      N = N \ 10  WEND  PRINT DIGIT | N = int(input())  sum = 0  while N > 0:      digit = N % 10      if digit < 7:          sum = sum + 1      N = N // 10  print(digit) |
| **Ал­го­рит­ми­че­ский язык** | **Пас­каль** |
| алг  нач      цел N, digit, sum      ввод N      sum := 0      нц пока N > 0          digit := mod(N,10)          если digit < 7 то              sum := sum + 1          все          N := div(N,10)      кц      вывод digit  кон | var N, digit, sum: longint;  begin      readln(N);      sum := 0;      while N > 0 do      begin          digit := N mod 10;          if digit < 7 then              sum := sum + 1;          N := N div 10;      end;      writeln(digit)  end. |
| **Си** | |
| #include  int main()  {      int N, digit, sum;      scanf("%d", &N);      sum = 0;      while (N > 0)      {          digit = N % 10;          if (digit < 7)              sum = sum + 1;          N = N / 10;      }      printf("%d",digit);      return0;  } | |

По­сле­до­ва­тель­но вы­пол­ни­те сле­ду­ю­щее.

1. На­пи­ши­те, что вы­ве­дет эта про­грам­ма при вводе числа 456.

2. При­ве­ди­те при­мер та­ко­го трёхзнач­но­го числа, при вводе ко­то­ро­го про­грам­ма выдаёт вер­ный ответ.

3. Най­ди­те все ошиб­ки в этой про­грам­ме (их может быть одна или не­сколь­ко). Из­вест­но, что каж­дая ошиб­ка за­тра­ги­ва­ет толь­ко одну стро­ку и может быть ис­прав­ле­на без из­ме­не­ния дру­гих строк. Для каж­дой ошиб­ки:

1) вы­пи­ши­те стро­ку, в ко­то­рой сде­ла­на ошиб­ка;

2) ука­жи­те, как ис­пра­вить ошиб­ку, т.е. при­ве­ди­те пра­виль­ный ва­ри­ант стро­ки.

До­ста­точ­но ука­зать ошиб­ки и спо­соб их ис­прав­ле­ния для од­но­го языка про­грам­ми­ро­ва­ния. Об­ра­ти­те вни­ма­ние, что тре­бу­ет­ся найти ошиб­ки в име­ю­щей­ся про­грам­ме, а не на­пи­сать свою, воз­мож­но, ис­поль­зу­ю­щую дру­гой ал­го­ритм ре­ше­ния. Ис­прав­ле­ние ошиб­ки долж­но за­тра­ги­вать толь­ко стро­ку, в ко­то­рой на­хо­дит­ся ошиб­ка.

**По­яс­не­ние.**

Ре­ше­ние ис­поль­зу­ет за­пись про­грам­мы на Пас­ка­ле. До­пус­ка­ет­ся ис­поль­зо­ва­ние про­грам­мы на любом из четырёх дру­гих язы­ков.

1. Про­грам­ма вы­ве­дет число 4.

2. При­мер числа, при вводе ко­то­ро­го про­грам­ма выдаёт вер­ный ответ: 835.

За­ме­ча­ние для про­ве­ря­ю­ще­го. Про­грам­ма ра­бо­та­ет не­пра­виль­но из-за не­вер­ной вы­во­ди­мой на экран пе­ре­мен­ной и не­вер­но­го уве­ли­че­ния суммы. Со­от­вет­ствен­но, про­грам­ма будет ра­бо­тать верно, если в числе стар­шая цифра (край­няя левая) равна сумме цифр, мень­ших 7.

3. В про­грам­ме есть две ошиб­ки.

Пер­вая ошиб­ка. Не­вер­ное уве­ли­че­ние суммы.

Стро­ка с ошиб­кой:

sum := sum + 1;

Вер­ное ис­прав­ле­ние:

sum := sum + digit;

Вто­рая ошиб­ка. Не­вер­ный вывод от­ве­та на экран.

Стро­ка с ошиб­кой:

writeln(digit)

Вер­ное ис­прав­ле­ние:

writeln(sum)

**25. За­да­ние** Дан це­ло­чис­лен­ный мас­сив из 20 эле­мен­тов. Эле­мен­ты мас­си­ва могут при­ни­мать целые зна­че­ния от –10 000 до 10 000 вклю­чи­тель­но. Опи­ши­те на есте­ствен­ном языке или на одном из язы­ков про­грам­ми­ро­ва­ния ал­го­ритм, поз­во­ля­ю­щий найти и вы­ве­сти ко­ли­че­ство пар эле­мен­тов мас­си­ва, в ко­то­рых хотя бы одно число де­лит­ся на 3. В дан­ной за­да­че под парой под­ра­зу­ме­ва­ет­ся два под­ряд иду­щих эле­мен­та мас­си­ва. На­при­мер, для мас­си­ва из пяти эле­мен­тов: 6; 2; 9; –3; 6 – ответ: 4.

Ис­ход­ные дан­ные объ­яв­ле­ны так, как по­ка­за­но ниже на при­ме­рах для не­ко­то­рых язы­ков про­грам­ми­ро­ва­ния и есте­ствен­но­го языка. За­пре­ща­ет­ся ис­поль­зо­вать пе­ре­мен­ные, не опи­сан­ные ниже, но раз­ре­ша­ет­ся не ис­поль­зо­вать не­ко­то­рые из опи­сан­ных пе­ре­мен­ных.

|  |  |
| --- | --- |
| **Бей­сик** | **Python** |
| CONST N AS INTEGER = 20  DIM A (1 TO N) AS INTEGER  DIM I AS INTEGER,         J AS INTEGER,         K AS INTEGER    FOR I = 1 TO N  INPUT A(I)  NEXT I  ...  END | # до­пус­ка­ет­ся также  # ис­поль­зо­вать две  # це­ло­чис­лен­ные пе­ре­мен­ные j и k  a = []  n = 20  for i in range(0, n):      a.append(int(input()))  ... |
| **Ал­го­рит­ми­че­ский язык** | **Пас­каль** |
| алг  нач      цел N = 20      цел­таб a[1:N]      цел i, j, k      нц для i от 1 до N          ввод a[i]      кц      ...  кон | const      N = 20;  var      a: array [1..N] of integer;      i, j, k: integer;  begin      for i := 1 to N do          readln(a[i]);      ...  end. |
| **Си** | **Есте­ствен­ный язык** |
| #include  #define N 20  int main() {      int a[N];      int i, j, k;      for (i = 0; i < N; i++)          scanf("%d", &a[i]);      ...      return 0;  } | Объ­яв­ля­ем мас­сив A из 20 эле­мен­тов.  Объ­яв­ля­ем це­ло­чис­лен­ные пе­ре­мен­ные I, J, K.  В цикле от 1 до 20 вво­дим эле­мен­ты мас­си­ва A с 1-го по 20-й.  … |

В ка­че­стве от­ве­та Вам не­об­хо­ди­мо при­ве­сти фраг­мент про­грам­мы (или опи­са­ние ал­го­рит­ма на есте­ствен­ном языке), 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 2.6) или в виде блок-схемы. В этом слу­чае Вы долж­ны ис­поль­зо­вать те же самые ис­ход­ные дан­ные и пе­ре­мен­ные, какие были пред­ло­же­ны в усло­вии (на­при­мер, в об­раз­це, за­пи­сан­ном на есте­ствен­ном языке).

**По­яс­не­ние.**

|  |
| --- |
| **Бей­сик** |
| K = 0  FOR I = 1 TO N-1      IF (A(I) MOD 3 = 0) OR (A(I + 1) MOD 3 = 0) THEN          K = K+1      END IF  NEXT I  PRINT K |
| **Python** |
| k = 0  for i in range(0, n – 1):      if (a[i] % 3 == 0 or a[i + 1] % 3 == 0):          k += 1  print(k) |
| **Ал­го­рит­ми­че­ский язык** |
| k := 0;  нц для i от 1 до N-1      если mod(a[i],3)=0 или mod(a[i+1],3)=0      то          k := k+1      все  кц  вывод k |
| **Пас­каль** |
| k := 0;  for i := 1 to N-1 do      if (a[i] mod 3=0) or (a[i+1] mod 3=0) then          inc(k);  writeln(k); |
| **Си** |
| k = 0;  for (i = 0; i      if (a[i]%3 == 0 || a[i+1]%3 == 0)          k++;  printf("%d", k); |
| **Есте­ствен­ный язык** |
| За­пи­сы­ва­ем в пе­ре­мен­ную K на­чаль­ное зна­че­ние, рав­ное 0. В цикле от пер­во­го эле­мен­та до пред­по­след­не­го на­хо­дим оста­ток от де­ле­ния те­ку­ще­го и сле­ду­ю­ще­го эле­мен­та мас­си­ва на 3. Если пер­вый или вто­рой из по­лу­чен­ных остат­ков равен 0, уве­ли­чи­ва­ем пе­ре­мен­ную K на еди­ни­цу. После за­вер­ше­ния цикла вы­во­дим зна­че­ние пе­ре­мен­ной K |

**26. За­да­ние** Два иг­ро­ка, Петя и Ваня, иг­ра­ют в сле­ду­ю­щую игру. Перед иг­ро­ка­ми лежат две кучи кам­ней. Иг­ро­ки ходят по оче­ре­ди, пер­вый ход де­ла­ет Петя. За один ход игрок может до­ба­вить в одну из куч (по сво­е­му вы­бо­ру) один ка­мень или уве­ли­чить ко­ли­че­ство кам­ней в куче в два раза. На­при­мер, пусть в одной куче 10 кам­ней, а в дру­гой 7 кам­ней; такую по­зи­цию в игре будем обо­зна­чать (10, 7). Тогда за один ход можно по­лу­чить любую из четырёх по­зи­ций: (11, 7), (20, 7), (10, 8), (10, 14). Для того чтобы де­лать ходы, у каж­до­го иг­ро­ка есть не­огра­ни­чен­ное ко­ли­че­ство кам­ней.

Игра за­вер­ша­ет­ся в тот мо­мент, когда сум­мар­ное ко­ли­че­ство кам­ней в кучах ста­но­вит­ся не менее 73. По­бе­ди­те­лем счи­та­ет­ся игрок, сде­лав­ший по­след­ний ход, т.е. пер­вым по­лу­чив­ший такую по­зи­цию, что в кучах всего будет 73 камня или боль­ше.

Будем го­во­рить, 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. На­при­мер, при на­чаль­ных по­зи­ци­ях (6, 34), (7, 33), (9, 32) вы­иг­рыш­ная стра­те­гия есть у Пети. Чтобы вы­иг­рать, ему до­ста­точ­но удво­ить ко­ли­че­ство кам­ней во вто­рой куче.

**За­да­ние 1.** Для каж­дой из на­чаль­ных по­зи­ций (6, 33), (8, 32) ука­жи­те, кто из иг­ро­ков имеет вы­иг­рыш­ную стра­те­гию. В каж­дом слу­чае опи­ши­те вы­иг­рыш­ную стра­те­гию; объ­яс­ни­те, по­че­му эта стра­те­гия ведёт к вы­иг­ры­шу, и ука­жи­те, какое наи­боль­шее ко­ли­че­ство ходов может по­тре­бо­вать­ся по­бе­ди­те­лю для вы­иг­ры­ша при этой стра­те­гии.

**За­да­ние 2.** Для каж­дой из на­чаль­ных по­зи­ций (6, 32), (7, 32), (8, 31) ука­жи­те, кто из иг­ро­ков имеет вы­иг­рыш­ную стра­те­гию. В каж­дом слу­чае опи­ши­те вы­иг­рыш­ную стра­те­гию; объ­яс­ни­те, по­че­му эта стра­те­гия ведёт к вы­иг­ры­шу, и ука­жи­те, какое наи­боль­шее ко­ли­че­ство ходов может по­тре­бо­вать­ся по­бе­ди­те­лю для вы­иг­ры­ша при этой стра­те­гии.

**За­да­ние 3.** Для на­чаль­ной по­зи­ции (7, 31) ука­жи­те, кто из иг­ро­ков имеет вы­иг­рыш­ную стра­те­гию. Опи­ши­те вы­иг­рыш­ную стра­те­гию; объ­яс­ни­те, по­че­му эта стра­те­гия ведёт к вы­иг­ры­шу, и ука­жи­те, какое наи­боль­шее ко­ли­че­ство ходов может по­тре­бо­вать­ся по­бе­ди­те­лю для вы­иг­ры­ша при этой стра­те­гии. По­строй­те де­ре­во всех пар­тий, воз­мож­ных при ука­зан­ной Вами вы­иг­рыш­ной стра­те­гии. Пред­ставь­те де­ре­во в виде ри­сун­ка или таб­ли­цы.

**По­яс­не­ние.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Положение после очередных ходов** | | | |
| **Исходное положение** | 1-й ход Пети (разобраны все ходы, указанна полученная позиция) | 1-й ход Вани (только ход по стратегии, указана полученная позиция) | 2-й ход Пети (разобраны все ходы, указана полученная позиция) | 2-й ход Вани (только ход по стратегии, указана полученная позиция) |
| (7,31)  Всего 38 | (7,31+1)=(7,32)  Всего 39 | (7+1,32)=(8,32)  Всего 40 | (8+1,32)=(9,32)  Всего 41 | (9,32\*2)=(9,64)  Всего 73 |
| (8,32+1)=(8,33)  Всего 41 | (8,33\*2)=(8,66)  Всего 74 |
| (8\*2,32)=(16,32)  Всего 48 | (16,32\*2)=(16,64)  Всего80 |
| (8,32\*2)=(8,64)  Всего 72 | (8,64\*2)=(8,128)  Всего 136 |
| (7+1,31)=(8,31)  Всего 39 | (8,31+1)=(8,32)  Всего 40 | (8+1,32)=(9,32)  Всего 41 | (9,32\*2)=(9,64)  Всего 73 |
| (8,32+1)=(8,33)  Всего41 | (8,33\*2)=(8,66)  Всего 74 |
| (8\*2,32)=(16,32)  Всего 48 | (16,32\*2)=(16,64)  Всего 80 |
| (8,32\*2)=(8,64)  Всего 72 | (8,64\*2)=(8,128)  Всего 136 |
| (7\*2,31)=(14,31)  Всего 45 | (14,31\*2)=(14,62)  Всего 76 |  |  |
| (7,31\*2)=(7,62)  Всего 69 | (7,62\*2)=(7,124)  Всего 131 |  |  |

**За­да­ние 1.** В на­чаль­ных по­зи­ци­ях (6, 33), (8, 32) вы­иг­рыш­ная стра­те­гия есть у Вани. При на­чаль­ной по­зи­ции (6, 33) после пер­во­го хода Пети может по­лу­чить­ся одна из сле­ду­ю­щих четырёх по­зи­ций: (7, 33), (12, 33), (6, 34), (6, 66). Каж­дая из этих по­зи­ций со­дер­жит менее 73 кам­ней. При этом из любой из этих по­зи­ций Ваня может по­лу­чить по­зи­цию, со­дер­жа­щую не менее 73 кам­ней, удво­ив ко­ли­че­ство кам­ней во вто­рой куче. Для по­зи­ции (8, 32) после пер­во­го хода Пети может по­лу­чить­ся одна из сле­ду­ю­щих четырёх по­зи­ций: (9, 32), (16, 32), (8, 33), (8, 64). Каж­дая из этих по­зи­ций со­дер­жит менее 73 кам­ней. При этом из любой из этих по­зи­ций Ваня может по­лу­чить по­зи­цию, со­дер­жа­щую не менее 73 кам­ней, удво­ив ко­ли­че­ство кам­ней во вто­рой куче. Таким об­ра­зом, Ваня при любом ходе Пети

вы­иг­ры­ва­ет своим пер­вым ходом.

**За­да­ние 2.** В на­чаль­ных по­зи­ци­ях (6, 32), (7, 32) и (8, 31) вы­иг­рыш­ная стра­те­гия есть у Пети. При на­чаль­ной по­зи­ции (6, 32) он дол­жен пер­вым ходом по­лу­чить по­зи­цию (6, 33), из на­чаль­ных по­зи­ций (7, 32) и (8, 31). Петя после пер­во­го хода дол­жен по­лу­чить по­зи­цию (8, 32). По­зи­ции (6, 33) и (8, 32) рас­смот­ре­ны при раз­бо­ре за­да­ния 1. В этих по­зи­ци­ях вы­иг­рыш­ная стра­те­гия есть у иг­ро­ка, ко­то­рый будет хо­дить вто­рым (те­перь это Петя). Эта стра­те­гия опи­са­на при раз­бо­ре за­да­ния 1. Таким об­ра­зом, Петя при любой игре Вани вы­иг­ры­ва­ет своим вто­рым ходом.

**За­да­ние 3.** В на­чаль­ной по­зи­ции (7, 31) вы­иг­рыш­ная стра­те­гия есть у Вани. После пер­во­го хода Пети может воз­ник­нуть одна из четырёх по­зи­ций: (8, 31), (7, 32), (14, 31) и (7, 62). В по­зи­ци­ях (14, 31) и (7, 62) Ваня может вы­иг­рать одним ходом, удво­ив ко­ли­че­ство кам­ней во вто­рой куче. По­зи­ции (8, 31) и (7, 32) были рас­смот­ре­ны при раз­бо­ре за­да­ния 2. В этих по­зи­ци­ях у иг­ро­ка, ко­то­рый дол­жен сде­лать ход (те­перь это Ваня), есть вы­иг­рыш­ная стра­те­гия. Эта стра­те­гия опи­са­на при раз­бо­ре за­да­ния 2. Таким об­ра­зом, в за­ви­си­мо­сти от игры Пети Ваня вы­иг­ры­ва­ет на пер­вом или вто­ром ходу.

**27. За­да­ние** В фи­зи­че­ской ла­бо­ра­то­рии про­во­дит­ся дол­го­вре­мен­ный экс­пе­ри­мент по изу­че­нию гра­ви­та­ци­он­но­го поля Земли. По ка­на­лу связи каж­дую ми­ну­ту в ла­бо­ра­то­рию пе­ре­даётся по­ло­жи­тель­ное целое число – те­ку­щее по­ка­за­ние при­бо­ра «Сигма 2015». Ко­ли­че­ство пе­ре­да­ва­е­мых чисел в серии из­вест­но и не пре­вы­ша­ет 10 000. Все числа не пре­вы­ша­ют 1000. Вре­ме­нем, в те­че­ние ко­то­ро­го про­ис­хо­дит пе­ре­да­ча, можно пре­не­бречь.

Не­об­хо­ди­мо вы­чис­лить «бета-зна­че­ние» серии по­ка­за­ний при­бо­ра – ми­ни­маль­ное чётное про­из­ве­де­ние двух по­ка­за­ний, между мо­мен­та­ми пе­ре­да­чи ко­то­рых про­шло не менее 6 минут. Если по­лу­чить такое про­из­ве­де­ние не удаётся, ответ счи­та­ет­ся рав­ным –1.

*Вам пред­ла­га­ет­ся два за­да­ния, свя­зан­ных с этой за­да­чей: за­да­ние А и за­да­ние Б. Вы мо­же­те ре­шать оба за­да­ния или одно из них по сво­е­му вы­бо­ру. Ито­го­вая оцен­ка вы­став­ля­ет­ся как мак­си­маль­ная из оце­нок за за­да­ния А и Б. Если ре­ше­ние од­но­го из за­да­ний не пред­став­ле­но, то счи­та­ет­ся, что оцен­ка за это за­да­ние – 0 бал­лов. За­да­ние Б яв­ля­ет­ся усложнённым ва­ри­ан­том за­да­ния А, оно со­дер­жит до­пол­ни­тель­ные тре­бо­ва­ния к про­грам­ме.*

А. На­пи­ши­те на любом языке про­грам­ми­ро­ва­ния про­грам­му для ре­ше­ния по­став­лен­ной за­да­чи, в ко­то­рой вход­ные дан­ные будут за­по­ми­нать­ся в мас­си­ве, после чего будут про­ве­ре­ны все воз­мож­ные пары эле­мен­тов. Перед про­грам­мой ука­жи­те вер­сию языка про­грам­ми­ро­ва­ния.

ОБЯ­ЗА­ТЕЛЬ­НО ука­жи­те, что про­грам­ма яв­ля­ет­ся ре­ше­ни­ем ЗА­ДА­НИЯ А.

Мак­си­маль­ная оцен­ка за вы­пол­не­ние за­да­ния А – 2 балла.

Б. На­пи­ши­те про­грам­му для ре­ше­ния по­став­лен­ной за­да­чи, ко­то­рая будет эф­фек­тив­на как по вре­ме­ни, так и по па­мя­ти (или хотя бы по одной из этих ха­рак­те­ри­стик).

Про­грам­ма счи­та­ет­ся эф­фек­тив­ной по вре­ме­ни, если время ра­бо­ты

про­грам­мы про­пор­ци­о­наль­но ко­ли­че­ству по­лу­чен­ных по­ка­за­ний при­бо­ра N, т.е. при уве­ли­че­нии N в k раз время ра­бо­ты про­грам­мы долж­но уве­ли­чи­вать­ся не более чем в k раз.

Про­грам­ма счи­та­ет­ся эф­фек­тив­ной по па­мя­ти, если раз­мер па­мя­ти, ис­поль­зо­ван­ной в про­грам­ме для хра­не­ния дан­ных, не за­ви­сит от числа N и не пре­вы­ша­ет 1 ки­ло­бай­та.

Перед про­грам­мой ука­жи­те вер­сию языка про­грам­ми­ро­ва­ния и крат­ко опи­ши­те ис­поль­зо­ван­ный ал­го­ритм.

ОБЯ­ЗА­ТЕЛЬ­НО ука­жи­те, что про­грам­ма яв­ля­ет­ся ре­ше­ни­ем ЗА­ДА­НИЯ Б.

Мак­си­маль­ная оцен­ка за пра­виль­ную про­грам­му, эф­фек­тив­ную по вре­ме­ни и по па­мя­ти, – 4 балла.

Мак­си­маль­ная оцен­ка за пра­виль­ную про­грам­му, эф­фек­тив­ную по вре­ме­ни, но не­эф­фек­тив­ную по па­мя­ти, – 3 балла. НА­ПО­МИ­НА­ЕМ! Не за­будь­те ука­зать, к ка­ко­му за­да­нию от­но­сит­ся каж­дая из пред­став­лен­ных Вами про­грамм.

Вход­ные дан­ные пред­став­ле­ны сле­ду­ю­щим об­ра­зом. В пер­вой стро­ке задаётся число N – общее ко­ли­че­ство по­ка­за­ний при­бо­ра. Га­ран­ти­ру­ет­ся, что N > 6. В каж­дой из сле­ду­ю­щих N строк задаётся одно по­ло­жи­тель­ное целое число – оче­ред­ное по­ка­за­ние при­бо­ра.

*При­мер вход­ных дан­ных:*

*11*

*12*

*45*

*5*

*3*

*17*

*23*

*21*

*20*

*19*

*18*

*17*

Про­грам­ма долж­на вы­ве­сти одно число – опи­сан­ное в усло­вии про­из­ве­де­ние либо –1, если по­лу­чить такое про­из­ве­де­ние не удаётся.

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

*54*

**По­яс­не­ние.**

За­да­ние Б (ре­ше­ние для за­да­ния А при­ве­де­но ниже, см. про­грам­му 4). Чтобы про­из­ве­де­ние было чётным, хотя бы один со­мно­жи­тель дол­жен быть чётным, по­это­му при по­ис­ке под­хо­дя­щих про­из­ве­де­ний чётные по­ка­за­ния при­бо­ра можно рас­смат­ри­вать в паре с лю­бы­ми дру­ги­ми, а нечётные – толь­ко с чётными.

Для каж­до­го по­ка­за­ния с но­ме­ром k, на­чи­ная с k = 7, рас­смот­рим все до­пу­сти­мые по усло­ви­ям за­да­чи пары, в ко­то­рых дан­ное по­ка­за­ние по­лу­че­но вто­рым. Ми­ни­маль­ное про­из­ве­де­ние из всех этих пар будет по­лу­че­но, если пер­вым в паре будет взято ми­ни­маль­ное под­хо­дя­щее по­ка­за­ние среди всех, по­лу­чен­ных от на­ча­ла приёма и до по­ка­за­ния с но­ме­ром k – 6. Если оче­ред­ное по­ка­за­ние чётное, ми­ни­маль­ное среди преды­ду­щих может быть любым, если нечётное – толь­ко чётным.

Для по­лу­че­ния эф­фек­тив­но­го по вре­ме­ни ре­ше­ния нужно по мере ввода дан­ных пом­нить аб­со­лют­ное ми­ни­маль­ное и ми­ни­маль­ное чётное по­ка­за­ние на каж­дый мо­мент вре­ме­ни, каж­дое вновь по­лу­чен­ное по­ка­за­ние умно­жать на со­от­вет­ству­ю­щий ему ми­ни­мум, имев­ший­ся на 6 эле­мен­тов ранее, и вы­брать ми­ни­маль­ное из всех таких про­из­ве­де­ний.

По­сколь­ку каж­дое те­ку­щее ми­ни­маль­ное по­ка­за­ние ис­поль­зу­ет­ся после ввода ещё 6 эле­мен­тов и после этого ста­но­вит­ся не­нуж­ным, до­ста­точ­но хра­нить толь­ко 6 по­след­них ми­ни­му­мов. Для этого можно ис­поль­зо­вать мас­сив из 6 эле­мен­тов и цик­ли­че­ски за­пол­нять его по мере ввода дан­ных. Раз­мер этого мас­си­ва не за­ви­сит от об­ще­го ко­ли­че­ства введённых по­ка­за­ний, по­это­му такое ре­ше­ние будет эф­фек­тив­ным не толь­ко по вре­ме­ни, но и по па­мя­ти. Чтобы хра­нить аб­со­лют­ный и чётный ми­ни­му­мы, нужно ис­поль­зо­вать два таких мас­си­ва. Ниже при­во­дит­ся при­мер такой про­грам­мы, на­пи­сан­ной на ал­го­рит­ми­че­ском языке.

При­мер 1. При­мер пра­виль­ной про­грам­мы на ал­го­рит­ми­че­ском языке. Про­грам­ма эф­фек­тив­на и по вре­ме­ни, и по па­мя­ти.

алг

нач

    цел s = 6 | тре­бу­е­мое рас­сто­я­ние между по­ка­за­ни­я­ми

    цел amax = 1001 | боль­ше мак­си­маль­но воз­мож­но­го по­ка­за­ния

    цел N

    ввод N

    цел a | оче­ред­ное по­ка­за­ние при­бо­ра

    цел­таб мини[0:s-1] | те­ку­щие ми­ни­му­мы по­след­них s эле­мен­тов

    цел­таб ми­ни­чет[0:s-1] | чётные ми­ни­му­мы по­след­них s эле­мен­тов

    цел i

    | вво­дим пер­вые s по­ка­за­ний, фик­си­ру­ем ми­ни­му­мы

    цел ма; ма := amax | ми­ни­маль­ное по­ка­за­ние

    цел мчет; мчет := amax | ми­ни­маль­ное чётное по­ка­за­ние

    нц для i от 1 до s

        ввод а

        ма := imin(ма, a)

        если mod(a,2) = 0 то мчет := imin(мчет,a) все

        мини[mod(i, s)] := ма

        ми­ни­чет[mod(i, s)] := мчет

    кц

    цел мп = amax\*amax | ми­ни­маль­ное зна­че­ние про­из­ве­де­ния

    цел п

    нц для i от s+1 до N

        ввод а

        если mod(a,2)=0

            то п := a \* мини[mod(i, s)]

            иначе если мчет < amax

                то п := a \* ми­ни­чет[mod(i, s)]

                иначе п := amax\*amax;

            все

        все

        мп := imin(мп, п)

        ма := imin(ма, a)

        если mod(a,2) = 0 то мчет := imin(мчет,a) все

        мини[mod(i, s)] := ма

        ми­ни­чет[mod(i, s)] := мчет

    кц

    если мп = amax\*amax то мп:=-1 все

    вывод мп

кон

Воз­мож­ны и дру­гие ре­а­ли­за­ции. На­при­мер, вме­сто цик­ли­че­ско­го за­пол­не­ния мас­си­ва можно каж­дый раз сдви­гать его эле­мен­ты. В при­ведённом ниже при­ме­ре хра­нят­ся и сдви­га­ют­ся не ми­ни­му­мы, а ис­ход­ные зна­че­ния. Это тре­бу­ет чуть мень­ше па­мя­ти (до­ста­точ­но од­но­го мас­си­ва вме­сто двух), но по вре­ме­ни ре­ше­ние со сдви­га­ми менее эф­фек­тив­но, чем с цик­ли­че­ским за­пол­не­ни­ем. Од­на­ко время ра­бо­ты остаётся про­пор­ци­о­наль­ным N, по­это­му мак­си­маль­ная оцен­ка за такое ре­ше­ние тоже со­став­ля­ет 4 балла.

Про­грам­ма 2. При­мер пра­виль­ной про­грам­мы на языке Пас­каль.

Про­грам­ма ис­поль­зу­ет сдви­ги, но эф­фек­тив­на по вре­ме­ни и по па­мя­ти

const s = 6; {тре­бу­е­мое рас­сто­я­ние между по­ка­за­ни­я­ми}

        amax = 1001; {боль­ше мак­си­маль­но воз­мож­но­го по­ка­за­ния}

var

    N: integer;

    a: array[1..s] of integer; {хра­не­ние s по­ка­за­ний при­бо­ра}

    a\_: integer; {ввод оче­ред­но­го по­ка­за­ния}

    ma: integer; {ми­ни­маль­ное число без s по­след­них}

    me: integer; {ми­ни­маль­ное чётное число без s по­след­них}

    mp: integer; {ми­ни­маль­ное зна­че­ние про­из­ве­де­ния}

    p: integer;

    i, j: integer;

begin

    readln(N);

    {Ввод пер­вых s чисел}

    for i:=1 to s do readln(a[i]);

    {Ввод осталь­ных зна­че­ний, поиск ми­ни­маль­но­го про­из­ве­де­ния}

    ma := amax; me := amax;

    mp :=amax\*amax;

    for i := s + 1 to N do begin

        readln(a\_);

        if a[1] < ma then ma := a[1];

        if (a[1] mod 2 = 0) and (a[1] < me) then me := a[1];

        if a\_ mod 2 = 0 then p := a\_ \* ma

        else if me < amax then p := a\_ \* me

        else p := amax\* amax;

        if (p < mp) then mp := p;

        {сдви­га­ем эле­мен­ты вспо­мо­га­тель­но­го мас­си­ва влево}

        for j := 1 to s - 1 do

            a[j] := a[j + 1];

        a[s] := a\_

    end;

    if mp = amax\*amax then mp:=-1;

    writeln(mp)

end.

Если вме­сто не­боль­шо­го мас­си­ва фик­си­ро­ван­но­го раз­ме­ра (цик­ли­че­ско­го или со сдви­га­ми) хра­нят­ся все ис­ход­ные дан­ные (или все те­ку­щие ми­ни­му­мы), про­грам­ма со­хра­ня­ет эф­фек­тив­ность по вре­ме­ни, но ста­но­вит­ся не­эф­фек­тив­ной по па­мя­ти, так как тре­бу­е­мая па­мять растёт про­пор­ци­о­наль­но N. Ниже при­во­дит­ся при­мер такой про­грам­мы на языке Пас­каль. По­доб­ные (и ана­ло­гич­ные по сути) про­грам­мы оце­ни­ва­ют­ся не выше 3 бал­лов.

Про­грам­ма 3. При­мер пра­виль­ной про­грам­мы на языке Пас­каль. Про­грам­ма эф­фек­тив­на по вре­ме­ни, но не­эф­фек­тив­на по па­мя­ти

const s = 6; {тре­бу­е­мое рас­сто­я­ние между по­ка­за­ни­я­ми}

        amax = 1001; {боль­ше мак­си­маль­но воз­мож­но­го по­ка­за­ния}

var

    N, p, i: integer;

    a: array[1..10000] of integer; {все по­ка­за­ния при­бо­ра}

    ma: integer; {ми­ни­маль­ное число без s по­след­них}

    me: integer; {ми­ни­маль­ное чётное число без s по­след­них}

    mp: integer; {ми­ни­маль­ное зна­че­ние про­из­ве­де­ния}

begin

    readln(N);

    {Ввод всех по­ка­за­ний при­бо­ра}

    for i:=1 to N do readln(a[i]);

    ma := amax;

    me := amax;

    mp := amax\*amax;

    for i := s + 1 to N do

    begin

        if a[i-s] < ma then ma := a[i-s];

        if (a[i-s] mod 2 = 0) and (a[i-s] < me) then

            me := a[i-s];

        if a[i] mod 2 = 0 then p := a[i] \* ma

        else if me < amax then p := a[i] \* me

        else p := amax \* amax;

        if (p < mp) then mp := p

    end;

    if mp = amax\*amax then mp := -1;

    writeln(mp)

end.

Воз­мож­но также пе­ре­бор­ное ре­ше­ние, в ко­то­ром на­хо­дят­ся про­из­ве­де­ния всех воз­мож­ных пар и из них вы­би­ра­ет­ся ми­ни­маль­ное. Ниже (см. про­грам­му 4) при­ведён при­мер по­доб­но­го ре­ше­ния. Это (и ана­ло­гич­ные ему) ре­ше­ние не­эф­фек­тив­но ни по вре­ме­ни, ни по па­мя­ти. Оно яв­ля­ет­ся ре­ше­ни­ем за­да­ния А, но не яв­ля­ет­ся ре­ше­ни­ем за­да­ния Б. Оцен­ка за такое ре­ше­ние – 2 балла.

Про­грам­ма 4. При­мер пра­виль­ной про­грам­мы на языке Пас­каль. Про­грам­ма не­эф­фек­тив­на ни по вре­ме­ни, ни по па­мя­ти

const s = 6; {тре­бу­е­мое рас­сто­я­ние между по­ка­за­ни­я­ми}

var

    N: integer;

    a: array[1..10000] of integer; {все по­ка­за­ния при­бо­ра}

    mp: integer; {ми­ни­маль­ное зна­че­ние про­из­ве­де­ния}

    i, j: integer;

begin

    readln(N);

    {Ввод зна­че­ний при­бо­ра}

    for i:=1 to N do

        readln(a[i]);

    mp := 1000 \* 1000 + 1;

    for i := 1 to N-s do begin

        for j := i+s to N do begin

            if (a[i]\*a[j] mod 2 = 0) and (a[i]\*a[j] < mp)

                then mp := a[i]\*a[j]

        end;

    end;

    if mp = 1000 \* 1000 + 1 then mp := -1;

    writeln(mp)

end.