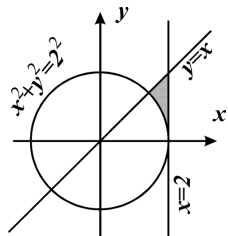


C1



Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно.

| ПРОГРАММА НА ПАСКАЛЕ | ПРОГРАММА НА БЕЙСИКЕ | ПРОГРАММА НА СИ |
|--|---|--|
| <pre>var x,y: real; begin readln(x,y); if x*x+y*y>=4 then if x<=2 then if y<=x then write('принадлежит') else write('не принадлежит'); end.</pre> | <pre>INPUT x, y IF x*x+y*y>=4 THEN IF x<=2 THEN IF y<=x THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF END</pre> | <pre>void main(void) { float x,y; scanf ("%f%f", &x, &y); if (x*x+y*y>=4) if (x<=2) if (y<=x) printf("принадлежит"); else printf("не принадлежит"); }</pre> |

Последовательно выполните следующее:

- 1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу.
- 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

Ответ:

| Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла) |
|--|
| <p>Элементы ответа:</p> <p>1) Пример: $x=1, y=-3$ (Любая пара (x, y), для которой выполняется: $x^2+y^2 < 4$ или $x > 2$ или $(y < 0$ и $y <= x)$ и $x^2+y^2 >= 4$)</p> <p>2) Возможная доработка (Паскаль): <pre>if (x*x+y*y>=4) and (x<=2) and (y<=x) and (y>=0) then write('принадлежит') else write('не принадлежит');</pre> (могут быть и другие способы доработки).</p> |

| Указания по оцениванию | Баллы |
|--|-------|
| <p>Обратите внимание! В задаче требовалось выполнить три действия: указать пример входных данных, при которых программа работает неверно, и исправить две ошибки:</p> <p>1. Неправильное использование условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего (отсутствуют случаи ELSE).</p> <p>2. Приведенным трем ограничениям удовлетворяют также те точки плоскости, у которых $(x^2+y^2) >= 4$ и $(y <= x)$ и $(x <= 2)$ и $(y < 0)$.</p> | |
| <p>Правильно выполнены оба пункта задания. Исправлены две ошибки.</p> <p>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения</p> | 3 |
| <p>Правильно выполнены 2 действия из трех (исправлены обе ошибки, но не указан/неправильно указан пример требуемых входных данных, либо правильно указан пример входных данных, программа правильно работает при большем числе случаев, чем исходная, но не при всех).</p> <p>Например, выдает "принадлежит" для точек, у которых $(x^2+y^2) >= 4$ и $(y <= x)$ и $(x <= 2)$ и $(y < 0)$.</p> <p>Допускается, например, такое решение:</p> <pre>if x*x+y*y>=4 then if x<=2 then if y<=x then write('принадлежит') else write('не принадлежит') else write('не принадлежит')</pre> | 2 |
| <p>Правильно выполнено только одно действие из трех.</p> <p>То есть только приведен пример входных данных, либо он не приведен, но имеется программа, корректно работающая при большем количестве входных данных, чем исходная, но не при всех.</p> | 1 |
| <p>Все пункты задания выполнены неверно (пример входных данных не указан или указан неверно, программа не приведена, либо приведенная программа корректно работает в не большем количестве случаев, чем исходная)</p> | 0 |
| <i>Максимальный балл</i> | |
| | 3 |

C2 Дан целочисленный массив из 30 элементов. Элементы массива могут принимать значения от 150 до 200 – рост учащихся выпускного класса. В команду по автогонкам входят все учащиеся, чей рост не более 175 см. Гарантируется, что такие учащиеся в классе есть. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит рост самого высокого участника гоночной команды. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

| Паскаль | Бейсик |
|---|---|
| <pre>const N=30; var a: array [1..N] of integer; i, j, max: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre> | <pre>N=30 DIM A(N) AS INTEGER DIM I, J, MAX AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre> |
| СИ | Естественный язык |
| <pre>#include <stdio.h> #define N 30 void main(void) {int a[N]; int i, j, max; for (i=0; i<N; i++) scanf("%d", &a[i]); ... }</pre> | <p>Объявляем массив A из 30 элементов. Объявляем целочисленные переменные I, J, MAX. В цикле от 1 до 30 вводим элементы массива A с 1-го по 30-й. ...</p> |

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, Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать переменные, аналогичные переменным, используемым в алгоритме, записанном на естественном языке, с учетом синтаксиса и особенностей используемого вами языка программирования.

Ответ:

| Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла) | |
|---|--|
| <p>На языке Паскаль</p> <pre>max:=150; for i:=1 to N do if (a[i]<=175) and (a[i]>max) then max:=a[i]; writeln(max);</pre> | <p>На языке Бейсик</p> <pre>MAX = 150 FOR I = 1 TO N IF A(I) <= 175 AND A(I) > MAX THEN MAX = A(I) ENDIF NEXT I PRINT MAX</pre> |
| <p>На языке СИ</p> <pre>max=150; for (i=0; i<N; i++) if (a[i]<=175 && a[i]>max) max=a[i]; printf("%d", max);</pre> | <p>На естественном языке</p> <p>Записываем в переменную MAX начальное значение, равное 150. В цикле от первого элемента до тридцатого сравниваем элементы исходного массива со 175. Если текущий элемент меньше или равен 175, то сравниваем значение текущего элемента массива со значением переменной MAX. Если текущий элемент массива больше MAX, то записываем в MAX значение этого элемента массива. Переходим к следующему элементу. После завершения цикла выводим значение переменной MAX.</p> |
| Указания по оцениванию | |
| <p>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 на естественном языке. 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.</p> | |
| Баллы | |
| 2 | |

| | |
|---|---|
| В любом варианте решения может присутствовать не более одной ошибки из числа следующих: 1) Не инициализируется или неверно инициализируется переменная MAX (например, ей присваивается значение a[1] или число большее 150) 2) В сравнении со 175 вместо знака «меньше или равно» используется знак «меньше». 3) Отсутствует вывод ответа 4) Используется переменная, не объявленная в разделе описания переменных. 5) Не указано или неверно указано условие завершения цикла 6) Индексная переменная в цикле не меняется (например, в цикле while). 7) Неверно расставлены операторные скобки. | 1 |
| Ошибок, перечисленных в п. 1-7, две или больше, или алгоритм сформулирован неверно. | 0 |
| <i>Максимальный балл</i> | 2 |

С3 Два игрока играют в следующую игру. На координатной плоскости стоит фишка. Игроки ходят по очереди. В начале игры фишка находится в точке с координатами (1,-1). Ход состоит в том, что игрок перемещает фишку из точки с координатами (x,y) в одну из трех точек: или в точку с координатами (x+3,y), или в точку с координатами (x,y+4), или в точку с координатами (x+2,y+2). Выигрывает игрок, после хода которого расстояние по прямой от фишки до точки с координатами (0,0) больше 9 единиц. Кто выигрывает при безошибочной игре обоих игроков – игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

Ответ:

| |
|---|
| Содержание верного ответа и указания к оцениванию (допускаются иные формулировки ответа, не искажающие его смысла) |
| Выигрывает второй игрок. Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны координаты фишки на каждом этапе игры. |

| | 1 ход | 2 ход | 3 ход | 4 ход |
|-------------------|-------------------------------|---|-------------------------------|--|
| Стартовая позиция | I-й игрок (все варианты хода) | II-й игрок (выигрышный ход) | I-й игрок (все варианты хода) | II-й игрок (выигрышный ход, один из вариантов) |
| 1,-1 | 3,1 | <u>5,3</u> | 8,3 | <u>11,3</u> |
| | | | 5,7 | <u>8,7</u> |
| | | | 7,5 | <u>10,5</u> |
| | 1,3 | <u>4,3</u> | 7,3 | <u>10,3</u> |
| | | | 4,7 | <u>7,7</u> |
| | | | 6,5 | <u>9,5</u> |
| 4,-1 | <u>4,3</u> | Те же варианты третьего-четвертого ходов. | | |

Таблица содержит **все возможные** варианты ходов первого игрока. Из неё видно, что при любом ходе первого игрока у второго имеется ход, приводящий к победе.

| Указания по оцениванию | Баллы |
|---|-------|
| Правильное указание выигрывающего игрока и его ходов со строгим доказательством правильности (с помощью или без помощи дерева игры). | 3 |
| Правильное указание выигрывающего игрока, стратегии игры, приводящей к победе, но при отсутствии доказательства ее правильности. | 2 |
| При наличии в представленном решении одного из пунктов: 1. Правильно указаны все варианты хода первого игрока и возможные ответы второго игрока (в том числе и все выигрышные), но неверно определены дальнейшие действия и неправильно указан победитель. 2. Правильно указан выигрывающий игрок, но описание выигрышной стратегии неполно и рассмотрены несколько (больше одного, но не все) вариантов хода первого игрока и частные случаи ответов второго игрока. | 1 |
| Задание не выполнено или в представленном решении полностью отсутствует описание элементов выигрышной стратегии, и отсутствует анализ вариантов первого-второго ходов играющих (даже при наличии правильного указания выигрывающего игрока). | 0 |
| <i>Максимальный балл</i> | 3 |

С4 Региональный этап олимпиады по праву проводился для учеников 9-11-х классов, участвующих в общем конкурсе. Каждый участник олимпиады мог набрать от 0 до 60 баллов. Для определения победителей и призеров сначала отбираются 25% участников, показавших лучшие результаты.

По положению, в случае, когда у последнего участника, входящего в 25%, оказывается количество баллов такое же, как и у следующих за ним в итоговой таблице, решение по данному участнику и всем участникам, имеющим с ним равное количество баллов, определяется следующим образом: все участники признаются призерами, если набранные ими баллы больше половины максимально возможных;

все участники не признаются призерами, если набранные ими баллы не превышают половины максимально возможных.

Напишите эффективную по времени работы и по используемой памяти программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая по результатам олимпиады будет определять, какой минимальный балл нужно было набрать, чтобы стать победителем или призером олимпиады.

На вход программе сначала подается число участников олимпиады N . В каждой из следующих N строк находится результат одного из участников олимпиады в следующем формате:

<Фамилия> <Имя> <класс> <баллы>

где <Фамилия> – строка, состоящая не более, чем из 20 символов, <Имя> – строка, состоящая не более, чем из 15 символов, <класс> – число от 9 до 11, <баллы> – целое число от 0 до 60 набранных участником баллов. <Фамилия> и <Имя>, <Имя> и <класс>, а также <класс> и <баллы> разделены одним пробелом. Пример входной строки:

Иванов Петр 10 17

Программа должна выводить минимальный балл призера. Гарантируется, что хотя бы одного призера по указанным правилам определить можно.

Ответ:

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

Программа читает все входные данные один раз, сразу подсчитывая в массиве с индексами от 0 до 60 количество участников, набравших тот или иной балл. Путем просмотра этого массива с конца (от 60 баллов) определяется число участников, заведомо попадающих в число 25% лучших (добавление всех участников, набравших следующий балл приводит к выходу за 25%). Последний балл, который набрали не менее 1 участника, запоминается. Если хотя бы один из следующих участников также попадает в 25%, то проверяется, что он и следующие набрали более половины баллов, в этом случае они все добавляются к числу победителей и призеров и их балл

является искомым.

Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая (например, когда все участники набрали различные баллы, каждый балл кто-то набрал и 25% от числа участников – целая величина).

Ниже приведены примеры решения задания на языках Бейсик и Паскаль. Допускаются решения, записанные на других языках программирования. При оценивании решений на других языках программирования необходимо учитывать особенности этих языков программирования.

Пример правильной и эффективной программы на языке Паскаль:

```
var
  cnt: array[0..60] of integer;
  c: char;
  i, k, N, b, S, minb: integer;
begin
  for i:=0 to 60 do
    cnt[i]:=0;
  readln(N);
  for i:=1 to N do
    begin
      repeat
        read(c);
      until c=' '; {считана фамилия}
      repeat
        read(c);
      until c=' '; {считано имя}
      readln(k,b);
      cnt[b]:=cnt[b]+1;
    end;
  S:=0;
  b:=60;
  while (S + cnt[b])*100<=N*25 do
    begin
      S:=S+cnt[b];
      if cnt[b]>0 then minb:=b;
      b:=b-1
    end; {определены те, кто наверняка стал призером и пропущены баллы
  которве никто не набрал}
  if (S+1)*100<=N*25 then
  {если еще хотя бы один участник попадает в 25%,
  то проверяется, какой балл набрали он и следующие участники}
  begin
    if b>30 then minb:=b
  end;
  writeln(minb);
end.
```

Пример правильной и эффективной программы на языке Бейсик:

```
DIM cnt(60) AS INTEGER
DIM ss AS STRING
FOR i = 0 TO 60
  cnt(i) = 0
NEXT i
INPUT N
FOR j = 1 TO N
  LINE INPUT ss
```

```

c$ = MID$(ss, 1, 1)
i = 1
WHILE NOT (c$ = " ")
  i = i + 1
  c$ = MID$(ss, i, 1)
WEND
i = i + 1
c$ = MID$(ss, i, 1)
WHILE NOT (c$ = " ")
  i = i + 1
  c$ = MID$(ss, i, 1)
WEND
i = i + 1
c$ = MID$(ss, i, 1)
IF c$ = "1" THEN
  i = i + 1
ENDIF
b = VAL(MID$(ss, i + 2))
cnt(b) = cnt(b) + 1
NEXT j
s = 0
b = 60
WHILE (s + cnt(b)) * 100 <= N * 25
  s = s + cnt(b)
  IF cnt(b) > 0 THEN minb = b
  b = b - 1
WEND
IF (s + 1) * 100 <= N * 25 THEN
  IF b > 30 THEN minb = b
ENDIF
PRINT minb
END
    
```

| Указания по оцениванию | Баллы |
|--|-------|
| Программа работает верно для любых входных данных произвольного размера и находит ответ, не сохраняя все входные данные или баллы участников в массиве, размер которого равен числу участников. Программа просматривает входные данные один раз, заполняя во время считывания массив размерностью от 0 до 60 для хранения количества участников, набравших то или иное количество баллов. Искомые величины находятся путем однократного просмотра этого массива. При определении минимального балла рассматриваются только баллы, которые кто-то набрал. Допускается наличие в тексте программы одной синтаксической ошибки. | 4 |
| Программа работает верно, но все входные данные (или баллы участника) запоминаются в массиве, размер которого совпадает с количеством участников, или входные данные считываются несколько раз. Возможно используется сортировка всех баллов участников и/или алгоритм поиска минимума, просматривающий баллы всех призеров. Допускается наличие от одной до трех синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное | 3 |

| | |
|---|---|
| слово языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для соответствующего типа данных. | |
| Программа работает в целом верно, эффективно или нет, но, в реализации алгоритма содержатся 1–2 ошибки (выход за границу массива, перевод символов в числа, используется знак “<” вместо “<=”, “or” вместо “and” и т.п.). Возможно, некорректно организовано считывание входных данных. Одной из двух ошибок может быть следующая логическая ошибка: при анализе результата последнего участника, вошедшего в 25%, вместо условия $(s+1)*100 \leq N*25$ используется $(s+cnt[b])*100 \leq N*25$, т.е. проверяется, что и набравшие вместе с ним тот же балл попали в 25%, или в качестве пограничных участников проверяются участники, не вошедшие в 25%. Ошибкой может быть также учет баллов, которые никто не набрал. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше. | 2 |
| Программа, возможно не всегда верно определяет количество победителей и призеров и, соответственно, минимальный балл у призеров. При использовании сортировки допущены ошибки в ее реализации. Допускается до 4 различных ошибок в реализации алгоритма, в том числе описанных в критериях присвоения двух баллов. Допускается наличие от одной до семи синтаксических ошибок, описанных выше. | 1 |
| Задание не выполнено или выполнено неверно | 0 |
| <i>Максимальный балл</i> | 4 |