

Задания С4 по информатике

1. Дан список результатов сдачи экзамена учащимися школ некоторого района, с указанием фамилии и имени учащегося, номера школы и итогового балла.

Напишите эффективную по времени работы и по используемой памяти программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая определяет номера школ, имеющих наибольший средний балл, показанный выпускниками данной школы на экзамене. На вход программе в первой строке подается количество учащихся во всех школах района N. В каждой из последующих N строк находится информация в следующем формате: <Фамилия> <Имя> <Номер школы> <Балл>

где <Фамилия> – строка, состоящая не более, чем из 20 символов без пробелов,

<Имя> – строка, состоящая не более, чем из 20 символов без пробелов,

<Номер школы> – число от 1 до 99,

<Балл> - число от 0 до 100.

Порядок следования строк – произвольный.

Пример входных данных:

6

Иванов Сергей 7 70

Сергеев Петр 3 65

Петров Кирилл 7 68

Кириллов Егор 3 75

Егоров Николай 7 71

Николаев Иван 19 70

Программа должна вывести номера школ, имеющих наибольший средний балл, показанный на экзамене выпускниками этой школы.

Пример вывода для приведенного выше примера ввода:

3

19

При выполнении задания следует учитывать, что значение N может быть велико (до 10 000).

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Создадим два массива с индексами от 1 до 99, соответствующих номерам школ и будем хранить в этих массивах количество выпускников из этой школы, сдававших экзамен и суммарный балл выпускников этой школы. Программа читает все входные данные один раз. После считывания фамилии, имени, номера школы и балла считанное значение прибавляется к суммарному баллу для данной школы, а количество выпускников из этой школы, сдававших экзамен, увеличиваем на 1.

После окончания считывания данных проходим циклом от 1 до 99 по всем школам и определяем максимальный средний балл по всем школам. Затем во втором цикле выводим номера школ, средний балл в которых равен максимальному. Ниже приведены примеры решения задания на языках Паскаль и С++. Допускаются решения, записанные на других языках программирования. При оценивании решений на других языках программирования необходимо учитывать особенности этих языков программирования. Решение на языке Паскаль

```
var
SchoolSum, SchoolCount: array[1..99] of integer;
i, N, School, Ball, MaxI: integer;
Max: extended;
c: char;
begin
for i:=1 to 99 do
begin
SchoolSum[i] := 0;
SchoolCount[i] := 0
end;
Readln(N);
for i:=1 to N do
begin
repeat
Read(c)
until c=' ';
repeat
Read(c)
until c=' ';
Readln(School, Ball);
SchoolSum[School] := SchoolSum[School] + Ball;
SchoolCount[School] := SchoolCount[School] + 1;
end;
Max:=-1;
for i:=1 to 99 do
if (SchoolCount[i] > 0) and (SchoolSum[i] / SchoolCount[i]
> Max)
then
begin
Max := SchoolSum[i] / SchoolCount[i];
MaxI := i
end;
for i:=1 to 99 do
if (SchoolCount[i]>0) and (SchoolSum[i]*SchoolCount[MaxI]=
SchoolSum[MaxI]*SchoolCount[i])
then
```

WriteLn(i)

end.

Решение на языке C++

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
int SchoolSum[100];
```

```
int SchoolCount[100];
```

```
int i, N, School, Ball, MaxI;
```

```
double Max;
```

```
string Name;
```

```
for (i = 1; i <= 99; ++i)
```

```
SchoolSum[i] = SchoolCount[i] = 0;
```

```
cin >> N;
```

```
for (i = 0; i < N; ++i)
```

```
{
```

```
cin >> Name >> Name >> School >> Ball;
```

```
SchoolSum[School] += Ball;
```

```
SchoolCount[School]++;
```

```
}
```

```
Max=-1;
```

```
for(i=1;i<=99;++i)
```

```
if (SchoolCount[i]>0 && (double)SchoolSum[i]/SchoolCount  
[i]>Max)
```

```
{
```

```
Max=(double)SchoolSum[i]/SchoolCount[i];
```

```
MaxI=i;
```

```
}
```

```
for(i=1;i<=99;++i)
```

```
if (SchoolCount[i]>0 && SchoolSum[i]*SchoolCount[MaxI]=  
SchoolSum[MaxI]*SchoolCount[i])
```

```
cout << i << endl;
```

```
return 0;
```

```
}
```

Указания по оцениванию	Балл
<p>Программа работает верно для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве, размер которого соответствует числу N (количество человек в списке). Программа просматривает входные данные один раз, сохраняя только информацию о суммарном балле и количестве выпускников в каждой школе. 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).</p>	4
<p>Программа работает верно, но входные данные запоминаются в массиве, размер которого не меньше N или входные данные считываются несколько раз. Допускается наличие от одной до трех синтаксических ошибок. Возможно, в принципиально верно организованном вводе данных есть одна ошибка. Три балла также выставляется, если в эффективной программе, удовлетворяющей критериям выставления 4 баллов, есть одна алгоритмическая ошибка (например, отсутствует вывод результата при правильном его нахождении).</p>	3
<p>Программа работает в целом верно, эффективно или нет, но, в реализации алгоритма содержатся до двух ошибок (неверная инициализация переменных, в частности значения максимума, выход за границу массива, перевод символов в числа, используется знак "<" вместо "<=", "or" вместо "and" и т.п.). Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p>	2
<p>Программа, возможно, неверно работает при некоторых входных данных, например, выдает правильный ответ только в том случае, если школа с максимальным средним баллом только одна. Возможно, неверно организовано считывание данных и отсутствует вывод результата. 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.</p>	1
<p><i>Задание не выполнено или выполнено неверно</i></p>	0

2. Дан список результатов сдачи экзамена учащимися школ некоторого района, с указанием фамилии и имени учащегося, номера школы и итогового балла.

Напишите эффективную по времени работы и по используемой памяти программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая определяет номера школ, имеющих наименьший средний балл, показанный выпускниками данной школы на экзамене. На вход программе в первой строке подается количество учащихся во всех школах района N. В каждой из последующих N строк находится информация в следующем формате:

<Фамилия> <Имя> <Номер школы> <Балл>

где <Фамилия> – строка, состоящая не более, чем из 20 символов без пробелов,

<Имя> – строка, состоящая не более, чем из 20 символов без пробелов,

<Номер школы> – число от 1 до 99,

<Балл> - число от 0 до 100.

Порядок следования строк – произвольный.

Пример входных данных:

```
6
Иванов Сергей 7 70
Сергеев Петр 3 68
Петров Кирилл 7 75
Кириллов Егор 3 72
Егоров Николай 7 72
Николаев Иван 19 70
```

Программа должна вывести номера школ, имеющих наименьший средний балл, показанный на экзамене выпускниками этой школы.

Пример вывода для приведенного выше примера ввода:

```
3 19
```

При выполнении задания следует учитывать, что значение N может быть велико (до 10 000).

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

Создадим два массива с индексами от 1 до 99, соответствующих номерам школ и будем хранить в этих массивах количество выпускников из этой школы, сдававших экзамен и суммарный балл выпускников этой школы. Программа читает все входные данные один раз. После считывания фамилии, имени, номера школы и балла считанное значение прибавляется к суммарному баллу для данной школы, а количество выпускников из этой школы, сдававших экзамен, увеличиваем на 1. После окончания считывания данных проходим циклом от 1 до 99 по всем школам и определяем минимальный средний балл по всем школам. Затем во втором цикле

выводим номера школ, средний балл в которых равен минимальному. Ниже приведены примеры решения задания на языках Паскаль и C++. Допускаются решения, записанные на других языках программирования. При оценивании решений на других языках программирования необходимо учитывать особенности этих языков программирования. Решение на языке Паскаль

```
var
SchoolSum, SchoolCount: array[1..99] of integer;
i, N, School, Ball, MinI: integer;
Min: extended;
c: char;
begin
for i:=1 to 99 do
begin
SchoolSum[i] := 0;
SchoolCount[i] := 0
end;
Readln(N);
for i:=1 to N do
begin
repeat
Read(c)
until c=' ';
repeat
Read(c)
until c=' ';
Readln(School, Ball);
SchoolSum[School] := SchoolSum[School] + Ball;
SchoolCount[School] := SchoolCount[School] + 1;
end;
Min:=101;
for i:=1 to 99 do
if (SchoolCount[i]>0) and (SchoolSum[i] / SchoolCount[i] <
Min)
then
begin
Min := SchoolSum[i] / SchoolCount[i];
MinI := i
end;
for i:=1 to 99 do
if (SchoolCount[i]>0) and (SchoolSum[i]*SchoolCount[MinI]=
SchoolSum[MinI]*SchoolCount[i])
then
WriteLn(i)
end.
```

Решение на языке C++

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
int SchoolSum[100];
int SchoolCount[100];
int i, N, School, Ball, MinI;
double Min;
string Name;
for (i = 1; i <= 99; ++i)
SchoolSum[i] = SchoolCount[i] = 0;
cin >> N;
for (i = 0; i < N; ++i)
{
cin >> Name >> Name >> School >> Ball;
SchoolSum[School] += Ball;
SchoolCount[School]++;
}
Min=101;
for(i=1;i<=99;++i)
if (SchoolCount[i]>0 && (double)SchoolSum[i]/SchoolCount
[i]<Min)
{
Min=(double)SchoolSum[i]/SchoolCount[i];
MinI=i;
}
for(i=1;i<=99;++i)
if (SchoolCount[i]>0 && SchoolSum[i]*SchoolCount[MinI]==
SchoolSum[MinI]*SchoolCount[i])
cout << i << endl;
return 0;
}
```

Указания по оцениванию	Балл
<p>Программа работает верно для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве, размер которого соответствует числу N (количество человек в списке). Программа просматривает входные данные один раз, сохраняя только информацию о суммарном балле и количестве выпускников в каждой школе. 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).</p>	4
<p>Программа работает верно, но входные данные запоминаются в массиве, размер которого не меньше N или входные данные считываются несколько раз. Допускается наличие от одной до трех синтаксических ошибок. Возможно, в принципиально верно организованном вводе данных есть одна ошибка. Три балла также выставляется, если в эффективной программе, удовлетворяющей критериям выставления 4 баллов, есть одна алгоритмическая ошибка (например, отсутствует вывод результата при правильном его нахождении).</p>	3
<p>Программа работает в целом верно, эффективно или нет, но, в реализации алгоритма содержатся до двух ошибок (неверная инициализация переменных, в частности значения максимума, выход за границу массива, перевод символов в числа, используется знак "<" вместо "<=", "or" вместо "and" и т.п.). Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p>	2
<p>Программа, возможно, неверно работает при некоторых входных данных, например, выдает правильный ответ только в том случае, если школа с максимальным средним баллом только одна. Возможно, неверно организовано считывание данных и отсутствует вывод результата. 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.</p>	1
<p><i>Задание не выполнено или выполнено неверно</i></p>	0

3. Имеется список людей с указанием их фамилии, имени и даты рождения. Напишите эффективную по времени работы и по используемой памяти программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет определять самого старшего человека из этого списка и выводить его фамилию, имя и дату рождения, а если имеется несколько самых старших людей с одинаковой датой рождения, то определять их количество. На вход программе в первой строке подается количество людей в списке N. В каждой из последующих N строк находится информация в следующем формате: <Фамилия> <Имя> <Дата рождения>

где <Фамилия> – строка, состоящая не более, чем из 20 символов без пробелов,

<Имя> – строка, состоящая не более, чем из 20 символов без пробелов,

<Дата рождения> – строка, имеющая вид ДД.ММ.ГГГГ, где ДД – двузначное число от 01 до 31, ММ – двузначное число от 01 до 12, ГГГГ – четырехзначное число от 1800 до 2100.

Пример входной строки:

Иванов Сергей 27.03.1993

Программа должна вывести фамилию и имя самого старшего человека в списке.

Пример выходных данных:

Иванов Сергей

Если таких людей, несколько, то программа должна вывести их количество. Пример вывода в этом случае: 3

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

Программа читает все входные данные один раз, не запоминая их в массиве. Во время чтения данных определяется дата рождения каждого человека, которая сравнивается с наименьшей из известных дат рождения среди всех уже просмотренных людей из списка. Также запоминается количество людей, имеющих наименьшую дату рождения, фамилия и имя человека, имеющего наименьшую дату рождения.

Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая (например, когда только один человек имеет наименьшую дату рождения).

Ниже приведены примеры решения задания на языках Паскаль и С++.

Допускаются решения, записанные на других языках программирования. При оценивании решений на других языках программирования необходимо учитывать особенности этих языков программирования.

Решение на языке Паскаль

Var

N, i, Day, Month, Year, MinDay, MinMonth, MinYear, MinCount,

Code: integer;

Name, MinName: string;

Begin

MinYear:=9999;

{Считываем количество людей}

ReadLn(N);

for i:=1 to N do

begin

{Считываем данные об одном человеке}

ReadLn(Name);

{Извлекаем из строки значения дня, месяца, года рождения}

val(Copy(Name,Length(Name)-9,2),Day,Code);

val(Copy(Name,Length(Name)-6,2),Month,Code);

val(Copy(Name,Length(Name)-3,4),Year,Code);

{Если дата рождения совпадает с наименьшей известной датой, то увеличиваем счетчик числа людей, имеющих минимальную дату рождения}

if (Year=MinYear) and (Month=MinMonth) and (Day=MinDay)

then

MinCount:=MinCount+1;

{Если дата рождения меньше минимально известной, то сохраняем его дату рождения, как минимальное, счетчик числа людей, имеющих такую дату рождения, устанавливаем в 1, также запоминаем имя человека}

if (Year<MinYear) or (Year=MinYear) and (Month<MinMonth)

or (Year=MinYear) and (Month=MinMonth) and (Day<MinDay)

then

begin

MinYear:=Year;

MinMonth:=Month;

MinDay:=Day;

MinCount:=1;

MinName:=Copy(Name,1,length(Name)-11)

end

end;

{Если счетчик числа людей, имеющих минимальную дату рождения больше 1, то выводим значение счетчика, иначе выводим имя человека}

if MinCount>1

then

WriteLn(MinCount)

Else

```
WriteLn(MinName)
```

```
end.
```

Решение на языке C++

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
int n, i, day, month, year, minday, minmonth, minyear,
```

```
mincount;
```

```
string name, minname;
```

```
cin >> n;
```

```
getline(cin, name);
```

```
minyear=9999;
```

```
for(i=0;i<n;++i)
```

```
{
```

```
getline(cin, name);
```

```
day=(name[name.length()-10]-'0')*10 + name[name.length()-9]-'0';
```

```
month=(name[name.length()-7]-'0')*10 + name[name.length()-9]-'6';
```

```
year=(((name[name.length()-4]-'0')*10 + name[name.length()-3]-'0') * 10 + name[name.length()-2]-'0')*10 + name[name.length()-1]-'0';
```

```
if (year==minyear && month==minmonth && day==minday) ++mincount;
```

```
if (year<minyear || year==minyear && month<minmonth || year==minyear && month==minmonth && day<minday)
```

```
{
```

```
minyear=year;
```

```
minmonth=month;
```

```
minday=day;
```

```
mincount=1;
```

```
name.resize(name.length()-11);
```

```
minname=name;
```

```
}
```

```
}
```

```
if(mincount>1)
```

```
cout<<mincount<<endl;
```

```
else
```

```
cout<<minname<<endl;
```

```
return 0;
```

```
}
```

Указания по оцениванию	Балл
<p>Программа работает верно для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве, размер которого соответствует числу N (количество человек в списке). Программа просматривает входные данные один раз, сохраняя только информацию о суммарном балле и количестве выпускников в каждой школе. 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).</p>	4
<p>Программа работает верно, но входные данные запоминаются в массиве, размер которого не меньше N или входные данные считываются несколько раз. Допускается наличие от одной до трех синтаксических ошибок. Возможно, в принципиально верно организованном вводе данных есть одна ошибка. Три балла также выставляется, если в эффективной программе, удовлетворяющей критериям выставления 4 баллов, есть одна алгоритмическая ошибка (например, отсутствует вывод результата при правильном его нахождении).</p>	3
<p>Программа работает в целом верно, эффективно или нет, но, в реализации алгоритма содержатся до двух ошибок (неверная инициализация переменных, в частности значения максимума, выход за границу массива, перевод символов в числа, используется знак "<" вместо "<=", "or" вместо "and" и т.п.). Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p>	2
<p>Программа, возможно, неверно работает при некоторых входных данных, например, выдает правильный ответ только в том случае, если школа с максимальным средним баллом только одна. Возможно, неверно организовано считывание данных и отсутствует вывод результата. 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.</p>	1
<p><i>Задание не выполнено или выполнено неверно</i></p>	0

4. Имеется список результатов голосования избирателей за несколько партий, в виде списка названий данных партий. На вход программе в первой строке подается количество избирателей в списке N. В каждой из последующих N строк записано название партии, за которую проголосовал данный избиратель, в виде текстовой строки. Длина строки не превосходит 50 символов, название может содержать буквы, цифры, пробелы и прочие символы.

Пример входных данных:

6

Party one

Party two

Party three

Party three

Party two

Party three

Программа должна вывести список всех партий, встречающихся в исходном списке, в порядке убывания количества голосов, отданных за эту партию. При этом название каждой партии должно быть выведено ровно один раз, вне зависимости от того, сколько голосов было отдано за данную партию.

Пример выходных данных для приведенного выше примера входных данных:

Party three

Party two

Party one

При этом следует учитывать, что количество голосов избирателей в исходном списке может быть велико (свыше 1000), а количество различных партий в этом списке не превосходит 10.

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, размер которого равен N, а составляя только список имеющихся политических партий и количества голосов, отданных за каждую из них. Во время чтения данных просматривается список ранее сохраненных партий, если данная партия уже есть в списке, то количество голосов, отданных за эту партию, увеличивается на 1, иначе партия добавляется в список известных партий. После окончания ввода производится сортировка массивов партий и количества голосов, отданных за них в порядке убывания количества голосов, затем выводится список партий. Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая.

Ниже приведены примеры решения задания на языках Паскаль и C++. Допускаются решения, записанные на других языках программирования. При оценивании

решений на других языках программирования необходимо учитывать особенности этих языков программирования.

Решение на языке Паскаль

Var

n, NumParties, i, j, t: integer;

Count:array[1..10] of integer;

s: string;

Names: array[1..10] of string;

Begin

{Общее число партий в списке}

NumParties:=0;

{Считываем количество избирателей}

ReadLn(N);

for i:=1 to N do

begin

{Считываем название партии, за которую проголосовал избиратель}

ReadLn(S);

{Осуществляем поиск считанного названия в списке всех известных партий}

j:=1;

while (j<=NumParties) and (s<>Names[j])

do

j:=j+1;

{Если партия найдена в списке}

if j<=NumParties

then

{Увеличиваем счетчик числа голосов, отданных за эту партию}

Count[j]:=Count[j]+1

else

begin

{Иначе добавляем партию в конец списка, увеличиваем число партий на 1, устанавливаем счетчик голосов, отданных за новую партию, в 1}

Names[j]:=s;

Count[j]:=1;

NumParties:=NumParties+1

end

end;

{Сортируем массивы Names и Count в порядке убывания значений массива Count}

for i:=NumParties downto 2 do

© МИОО, 2009 г.

for j:=2 to i do

if Count[j-1]<Count[j]

then

begin

```
t:=Count[j];
Count[j]:=Count[j-1];
Count[j-1]:=t;
s:=Names[j];
Names[j]:=Names[j-1];
Names[j-1]:=s;
end;
{Выводим содержимое массива Count}
for i:=1 to NumParties do
WriteLn(Names[i])
end.
```

Решение на языке C++

```
#include<iostream>
using namespace std;
int main()
{
int n, Count[10], NumParties=0, i, j, t;
string Names[10], s;
cin>>n;
getline(cin,s);
for(i=0;i<n;++i)
{
getline(cin,s);
j=0;
while(j<NumParties && s!=Names[j])
++j;
if(j<NumParties)
++Count[j];
else
{
Names[j]=s;
Count[j]=1;
++NumParties;
}
}
for (i=NumParties-1;i>=1;--i)
for (j=1;j<=i;++j)
if (Count[j-1]<Count[j])
{
t=Count[j];
Count[j]=Count[j-1];
Count[j-1]=t;
s=Names[j];
Names[j]=Names[j-1];
Names[j-1]=s;
}
for(i=0;i<NumParties;++i)
```

```
cout<<Names[i]<<endl;
}
```

Указания по оцениванию	Балл
<p>Программа работает верно для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве, размер которого соответствует числу N (количество человек в списке). Программа просматривает входные данные один раз, сохраняя только информацию о суммарном балле и количестве выпускников в каждой школе. 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).</p>	4
<p>Программа работает верно, но все входные данные запоминаются в массиве или входные данные считываются несколько раз. Допускается наличие от одной до трех синтаксических ошибок. Три балла также выставляется, если в эффективной программе, удовлетворяющей критериям выставления 4 баллов, есть одна ошибка, например, использование неверной границы в алгоритме сортировке, выход за границы массива, используется знак "<" вместо "<=" и т.д.</p>	3
<p>Программа работает в целом верно, эффективно или нет, но, в реализации алгоритма содержатся до двух ошибок (неверная инициализация переменных, в частности значения минимума, выход за границу массива, используется знак "<" вместо "<=", "or" вместо "and" и т.п.). Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p>	2
<p>Программа, возможно, неверно работает при некоторых входных данных, например, выдает правильный ответ только в том случае, если школа с максимальным средним баллом только одна. Возможно, неверно организовано считывание данных и отсутствует вывод результата. 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.</p>	1
<p><i>Задание не выполнено или выполнено неверно</i></p>	0

5. Имеется список результатов голосования избирателей за несколько партий, в виде списка названий данных партий. При этом некоторые голоса являются недействительными, в этом случае вместо названия партии записан один символ «.» (точка). Партия проходит в парламент, если за нее отдано не менее 7% от общего числа действительных голосов (то есть от количества строк в списке, которые не равны «.»). На вход программе в первой строке подается количество избирателей в списке N. В каждой из последующих N строк записано название партии, за которую проголосовал данный избиратель, в виде текстовой строки или один символ «.», если данный голос считается недействительным. Длина строки не превосходит 50 символов, название может содержать буквы, цифры, пробелы и прочие символы.

Пример входных данных:

```
Party one
Party two
Party one
Party one
Party two
Party two
.
Party three
Party one
Party one
Party two
Party one
Party two
Party two
Party one
Party two
```

Программа должна вывести список всех партий, прошедших в парламент, в произвольном порядке. При этом название каждой партии должно быть выведено ровно один раз, вне зависимости от того, сколько голосов было отдано за данную партию.

Пример выходных данных для приведенного выше примера входных данных:

```
Party one
Party two
```

При этом следует учитывать, что количество голосов избирателей в исходном списке может быть велико (свыше 1000), а количество различных партий в этом списке не превосходит 10.

**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, размер которого равен N , а составляя только список имеющихся политических партий и количества голосов, отданных за каждую из них. Во время чтения данных проверяется, не равна ли считанная строка «.». В этом случае просматривается список ранее сохраненных партий, если данная партия уже есть в списке, то количество голосов, отданных за эту партию, увеличивается на 1, иначе партия добавляется в список известных партий. Также счетчик общего числа действительных голосов увеличивается на 1. После окончания ввода просматривается список всех партий и выводятся названия тех из них, за которые отдано не менее 7% от общего числа голосов. Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая. Ниже приведены примеры решения задания на языках Паскаль и C++.

Допускаются решения, записанные на других языках программирования. При оценивании решений на других языках программирования необходимо учитывать особенности этих языков программирования.

Решение на языке Паскаль

```
Var
n, NumParties, NumVotes, i, j: integer;
Count:array[1..10] of integer;
s: string;
Names: array[1..10] of string;
Begin
NumParties:=0;
NumVotes:=0;
{Считываем количество избирателей}
ReadLn(N);
for i:=1 to N do
begin
{Считываем название партии, за которую проголосовал избиратель}
ReadLn(S);
if S<>'.' then
begin
NumVotes:=NumVotes+1;
{Осуществляем поиск названия в списке известных партий}
j:=1;
while (j<=NumParties) and (s<>Names[j])
do
j:=j+1;
{Если партия найдена в списке}
if j<=NumParties
then
{Увеличиваем счетчик числа голосов, отданных за эту партию}
```

```
Count[j]:=Count[j]+1
else
begin
{Иначе добавляем партию в конец списка,
увеличиваем число партий на 1,
устанавливаем счетчик голосов,
отданных за новую партию, в 1}
Names[j]:=s;
Count[j]:=1;
NumParties:=NumParties+1
end
end
end;
{Проходим по списку партий и выводим только те,
за которые отдано не менее 7% голосов}
for i:=1 to NumParties do
if Count[i]*100>=NumVotes*7 then
WriteLn(Names[i])
end.
```

Решение на языке C++

```
#include<iostream>
using namespace std;
int main()
{
int n, Count[10], NumParties=0, NumVotes=0, i, j;
string Names[10], s;
cin>>n;
getline(cin,s);
for(i=0;i<n;++i)
{
getline(cin,s);
if(s!=".")
{
NumVotes++;
j=0;
while(j<NumParties && s!=Names[j])
++j;
if(j<NumParties)
++Count[j];
else
{
Names[j]=s;
Count[j]=1;
++NumParties;
}
}
}
}
```

```

for(i=0;i<NumParties;++i)
if(Count[i]*100>=NumVotes*7)
cout<<Names[i]<<endl;
}

```

Указания по оцениванию	Балл
<p>Программа работает верно для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве, размер которого соответствует числу N (количество человек в списке). Программа просматривает входные данные один раз, сохраняя только информацию о суммарном балле и количестве выпускников в каждой школе. 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).</p>	4
<p>Программа работает верно, но все входные данные запоминаются в массиве или входные данные считываются несколько раз. Допускается наличие от одной до трех синтаксических ошибок. Три балла также выставляется, если в эффективной программе, удовлетворяющей критериям выставления 4 баллов, есть одна ошибка, например, использование неверной границы в алгоритме сортировке, выход за границы массива, используется знак "<" вместо "<=" и т.д.</p>	3
<p>Программа работает в целом верно, эффективно или нет, но, в реализации алгоритма содержатся до двух ошибок (неверная инициализация переменных, в частности значения минимума, выход за границу массива, используется знак "<" вместо "<=", "or" вместо "and" и т.п.). Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p>	2
<p>Программа, возможно, неверно работает при некоторых входных данных, например, выдает правильный ответ только в том случае, если школа с максимальным средним баллом только одна. Возможно, неверно организовано считывание данных и отсутствует вывод результата. 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.</p>	1
<p><i>Задание не выполнено или выполнено неверно</i></p>	0

6. Два игрока играют в следующую игру. На координатной плоскости стоит фишка.

В начале игры фишка находится в точке с координатами $(0,0)$. Игроки ходят по очереди. Ход состоит в том, что игрок перемещает фишку из точки с координатами (x, y) в одну из трех точек: $(x + 2, y + 3)$, $(x, y + 2)$, $(x + 1, y + 1)$. Игра заканчивается, как только расстояние по прямой от фишки до начала координат станет более 9. Выигрывает игрок, который сделал последний ход. Кто выигрывает при безошибочной игре – игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

16

Party one

Party two

Party one

Party one

Party two

Party two

.

Party three

Party one

Party one

Party two

Party one

Party two

Party two

Party one

Party two

Программа должна вывести список всех партий, прошедших в парламент, в произвольном порядке. При этом название каждой партии должно быть выведено ровно один раз, вне зависимости от того, сколько голосов было отдано за данную партию.

Пример выходных данных для приведенного выше примера входных данных:

Party one

Party two

При этом следует учитывать, что количество голосов избирателей в исходном списке может быть велико (свыше 1000), а количество различных партий в этом списке не превосходит 10.

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, размер которого равен N , а составляя только список имеющихся политических партий и количества голосов, отданных за каждую из них. Во время

чтения данных проверяется, не равна ли считанная строка «.». В этом случае просматривается список ранее сохраненных партий, если данная партия уже есть в списке, то количество голосов, отданных за эту партию, увеличивается на 1, иначе партия добавляется в список известных партий. Также счетчик общего числа действительных голосов увеличивается на 1. После окончания ввода просматривается список всех партий и выводятся названия тех из них, за которые отдано не менее 7% от общего числа голосов.

Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая.

Ниже приведены примеры решения задания на языках Паскаль и C++.

Допускаются решения, записанные на других языках программирования. При оценивании решений на других языках программирования необходимо учитывать особенности этих языков программирования.

Решение на языке Паскаль

```
Var
n, NumParties, NumVotes, i, j: integer;
Count:array[1..10] of integer;
s: string;
Names: array[1..10] of string;
Begin
NumParties:=0;
NumVotes:=0;
{Считываем количество избирателей}
ReadLn(N);
for i:=1 to N do
begin
{Считываем название партии, за которую проголосовал избиратель}
ReadLn(S);
if S<>'.' then
begin
NumVotes:=NumVotes+1;
{Осуществляем поиск названия в списке известных партий}
j:=1;
while (j<=NumParties) and (s<>Names[j])
do
j:=j+1;
{Если партия найдена в списке}
if j<=NumParties
then
{Увеличиваем счетчик числа голосов, отданных за эту партию}
Count[j]:=Count[j]+1
else
begin
{Иначе добавляем партию в конец списка,
```

```
увеличиваем число партий на 1,  
устанавливаем счетчик голосов,  
отданных за новую партию, в 1}  
Names[j]:=s;  
Count[j]:=1;  
NumParties:=NumParties+1  
end  
end  
end;  
{Проходим по списку партий и выводим только те,  
за которые отдано не менее 7% голосов}  
for i:=1 to NumParties do  
if Count[i]*100>=NumVotes*7 then  
WriteLn(Names[i])  
end.
```

Решение на языке C++

```
#include<iostream>  
using namespace std;  
int main()  
{  
int n, Count[10], NumParties=0, NumVotes=0, i, j;  
string Names[10], s;  
cin>>n;  
getline(cin,s);  
for(i=0;i<n;++i)  
{  
getline(cin,s);  
if(s!=".")  
{  
NumVotes++;  
j=0;  
while(j<NumParties && s!=Names[j])  
++j;  
if(j<NumParties)  
++Count[j];  
else  
{  
Names[j]=s;  
Count[j]=1;  
++NumParties;  
}  
}  
}  
for(i=0;i<NumParties;++i)
```

```
if(Count[i]*100>=NumVotes*7)
cout<<Names[i]<<endl;
}
```

Указания по оцениванию	Балл
<p>Программа работает верно для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве, размер которого соответствует числу N (количество человек в списке). Программа просматривает входные данные один раз, сохраняя только информацию о суммарном балле и количестве выпускников в каждой школе. 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).</p>	4
<p>Программа работает верно, но все входные данные запоминаются в массиве или входные данные считываются несколько раз. Допускается наличие от одной до трех синтаксических ошибок. Три балла также выставляется, если в эффективной программе, удовлетворяющей критериям выставления 4 баллов, есть одна ошибка, например, использование неверной границы в алгоритме сортировке, выход за границы массива, используется знак "<" вместо "<=" и т.д.</p>	3
<p>Программа работает в целом верно, эффективно или нет, но, в реализации алгоритма содержатся до двух ошибок (неверная инициализация переменных, в частности значения минимума, выход за границу массива, используется знак "<" вместо "<=", "or" вместо "and" и т.п.). Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p>	2
<p>Программа, возможно, неверно работает при некоторых входных данных, например, выдает правильный ответ только в том случае, если школа с максимальным средним баллом только одна. Возможно, неверно организовано считывание данных и отсутствует вывод результата. 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.</p>	1
<p><i>Задание не выполнено или выполнено неверно</i></p>	0

7. Статья 83 закона «О выборах депутатов Государственной Думы Федерального Собрания Российской Федерации» определяет следующий алгоритм пропорционального распределения депутатских мандатов.

Сначала подсчитывается сумма голосов избирателей, поданных за каждую партию и подсчитывается сумма голосов, поданных за все партии. Эта сумма делится на 450 – количество распределяемых мандатов, получается величина, называемая «первое избирательное частное». Далее каждая партия получает столько мандатов в парламенте, чему равна целая часть от деления числа голосов за данную партию на первое избирательное частное.

Если после этого сумма количества мандатов, отданных партиям, меньше 450, то оставшиеся до 450 мандаты передаются по одному партиям, в порядке убывания дробной части частного от деления числа голосов за данную партию на первое избирательное частное. Если же для двух партий эти дробные части равны, то преимущество отдается той партии, которая получила большее число голосов.

Напишите программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая по количеству голосов, отданных за каждую партию, будет определять количество мандатов, полученных этой партией в парламенте.

На вход программе подается количество партий N , участвовавших в выборах, $N \leq 20$. Далее идет N строчек, каждая строчка содержит название партии (строка длиной не более 50 символов, возможно, содержащая пробелы), затем, через пробел, количество голосов, полученных данной партией – число, не превосходящее .

Пример входных данных:

```
3
Party One 100000
Party Two 200000
Party Three 400000
```

Программа должна вывести названия всех партий и количество голосов в парламенте, полученных данной партией. Названия необходимо выводить в том же порядке, в котором они шли во входных данных. Пример выходных данных для данного примера входных данных:

```
Party One 64
Party Two 129
Party Three 257
```

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Создадим массивы, для хранения следующих данных о партиях:

1. Название партии Names – строкового типа
2. Количество голосов, полученных партиями Votes – целочисленного типа
3. Количество мандатов в парламенте, полученных партией Mandates – целочисленного типа
4. Дробная часть от деления количества голосов на первое избирательное частное

FracPart – вещественного типа (этот массив необходим для второй части распределения голосов).

Программа считывает данные один раз, запоминая названия партии в массиве Names, а количество голосов, отданных за партию – в массиве Votes. Одновременно в переменную SumVotes считается общее число голосов.

Затем в цикле по всем партиям (индексная переменная i обозначает номер партии) каждая партия получает количество мандатов, равное целой части от деления $Votes[i]$ на $(SumVotes/450)$. Данное число мандатов записывается в элемент массива Mandates $[i]$, в массив FracPart $[i]$ записывается дробная часть от указанного частного. Одновременно в SumMandates записывается общее число распределенных между партиями мандатов.

В следующем цикле пока величина SumMandates меньше 450, распределяется по одному дополнительному мандату между партиями. Для этого определяется партия, для которой наибольшая величина FracPart, а если величины FracPart – одинаковые, то партия, для которой больше величина Votes. Номер этой партии записывается в переменную i . Затем значение Mandates $[i]$ и SumMandates увеличивается на 1, а значение FracPart $[i]$ устанавливается в 0, чтобы эта партия выбывала из процедуры вторичного распределения голосов.

Решение на языке Паскаль:

```
Const
NumMandates=450; {Число место в парламенте}
Var
N, SumVotes, SumMandates, i, j: longint;
errcore: integer;
Names: array[1..20] of string; {Названия партий}
Votes: array[1..20] of longint; {Число голосов, отданных за партию}
Mandates: array[1..20] of longint; {Число мест в парламенте}
FracPart: array[1..20] of real; {Дробная часть при определении числа мест}
Begin
ReadLn(N); {Считали число партий}
SumVotes:=0; {Общее число голосов избирателей}
for i:=1 to N do
begin
```

```
ReadLn(Names[i]); {Считали название партии и число голосов}
j:=length(Names[i]);
while Names[i][j]<>' ' do j:=j-1; {Нашли последний пробел в
строке}
val(Сору(Names[i],j+1,10),Votes[i],errcode); {Выделили из строки
число голосов}
Names[i]:=Сору(Names[i],1,j-1); {Выделили из строки название
партии}
SumVotes:=SumVotes+Votes[i]; {Увеличили общее число голосов}
end;
SumMandates:=0; {Общее число мандатов}
for i:=1 to N do
begin
FracPart[i]:=Votes[i]/(SumVotes/NumMandates);
Mandates[i]:=trunc(FracPart[i]); {Столько мандатов получила данная
партия}
SumMandates:=SumMandates+Mandates[i]; {Увеличиваем общее число
мандатов}
FracPart[i]:=frac(FracPart[i])
end;
while SumMandates < NumMandates do {Раздаем дополнительные мандаты}
begin__
i:=1;
for j:=2 to N do {Выбираем партию с наибольшим значением FracPart}
if (FracPart[j]>FracPart[i]) or (FracPart[j]=FracPart[i]) and
(Votes[j]>Votes[i])
then i:=j;
Mandates[i]:=Mandates[i]+1; {Выдаем ей дополнительный мандат}
SumMandates:=SumMandates+1; {Увеличиваем общее число мандатов}
FracPart[i]:=0;
end;
for i:=1 to N do {Выводим результаты распределения мандатов}
WriteLn(Names[i], ' ',Mandates[i])
end.
```

Решение на языке C++:

```
#include<iostream>
#include<cstdio>
#include<cmath>
using namespace std;
int main()
{
const int NumMandates=450;
int N, SumVotes, SumMandates, i, j;
string Names[20];
```

```
int Votes[20];
int Mandates[20];
double FracPart[20];
cin>>N;
getline(cin,Names[0]);
SumVotes=0;
for(i=0;i<N;i++)
{
getline(cin,Names[i]);
j=Names[i].length()-1;
while(Names[i][j]!=' ') j--;
sscanf(Names[i].c_str()+j,"%d",&Votes[i]);
Names[i].resize(j-1);
SumVotes+=Votes[i];
}
SumMandates=0;
for(i=0;i<N;i++)
{
FracPart[i]=Votes[i]/((double)SumVotes/NumMandates);
Mandates[i]=trunc(FracPart[i]);
SumMandates=SumMandates+Mandates[i];
FracPart[i]=trunc(FracPart[i]);
}
while (SumMandates < NumMandates)
{
i=0;
for(j=1;j<N;j++)
if (FracPart[j]>FracPart[i] || FracPart[j]==FracPart[i] &&
Votes[j]>Votes[i])
i=j;
Mandates[i]++;
SumMandates++;
FracPart[i]=0;
}
for(i=0;i<N;i++)
cout<<Names[i]<<" "<<Mandates[i]<<endl;
}
```

Указания по оцениванию	Балл
<p>Программа работает верно для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве, размер которого соответствует числу N (количество человек в списке). Программа просматривает входные данные один раз, сохраняя только информацию о суммарном балле и количестве выпускников в каждой школе. 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).</p>	4
<p>Программа работает верно во всех случаях, но содержит одну ошибку (например, не рассмотрена возможность равенства голосов при вторичном распределении мандатов, или программа не выводит результат, при верном его нахождении). Допускается наличие от одной до трех синтаксических ошибок: Возможно, в принципиально верно организованном вводе данных есть одна ошибка.</p>	3
<p>Программа работает в целом верно, но в реализации алгоритма содержатся до двух ошибок (неверная инициализация переменных, в частности значения максимума, выход за границу массива, перевод символов в числа, используется знак "<" вместо "<=", "or" вместо "and" и т.п.). Помимо этого, возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p>	2
<p>Программа, возможно, неверно работает при некоторых входных данных. Возможно, неверно реализован (или полностью отсутствует) механизм вторичного распределения мандатов). 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.</p>	1
<p><i>Задание не выполнено или выполнено неверно</i></p>	0

8. Дан список результатов сдачи экзамена учащимися школ некоторого района, с указанием фамилии и имени учащегося, номера школы и итогового балла. Напишите эффективную по времени работы и по используемой памяти программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая определяет список школ, в которых средний балл на экзамене, набранный учащимися из данной школы выше, чем средний балл по всему району.

На вход программе в первой строке подается количество учащихся во всех школах района N .

В каждой из последующих N строк находится информация в следующем формате:

<Фамилия> <Имя> <Номер школы> <Балл>

где <Фамилия> – строка, состоящая не более, чем из 20 символов без пробелов,

<Имя> – строка, состоящая не более, чем из 20 символов без пробелов,

<Номер школы> – число от 1 до 99,

<Балл> - число от 0 до 100.

Порядок следования строк – произвольный.

Пример входных данных:

6

Иванов Сергей 1 90

Сергеев Петр 21 80

Петров Кирилл 31 70

Кириллов Егор 1 80

Егоров Николай 21 90

Николаев Иван 31 70

Программа должна определить количество школ в районе, в которых средний балл на экзамене, набранный учащимися из данной школы выше, чем средний балл по всему району и вывести сначала количество таких школ, а в следующей строке – номера этих школ через пробел. Пример вывода для приведенного выше примера ввода:

2

1 21

При выполнении задания следует учитывать, что значение N может быть велико (до 10.000).

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

Создадим два массива с индексами от 1 до 99, соответствующих номерам школ, в которых будет храниться количество учащихся из данной школы и суммарный балл всех учащихся из данной школы на экзамене. Также создается переменная, в которую будет записываться суммарный балл по всему району.

Программа читает все входные данные один раз. После считывания фамилии, имени, номера школы и балла количество учащихся из данной школы увеличивается на 1, сумма баллов, набранных учащимися из данной школы и сумма баллов, набранных учащимися всех школ увеличивается на балл данного учащегося. После окончания исполнения алгоритма однократным проходом циклом от 1 до 99 определяется количество школ, в которых средний балл выше, чем в среднем по району. Затем однократным проходом циклом выводятся номера этих школ. Ниже приведены примеры решения задания на языках Паскаль и C++.

Допускаются решения, записанные на других языках программирования. При оценивании решений на других языках программирования необходимо учитывать особенности этих языков программирования.

Решение на языке Паскаль

```
var
SchoolSum, SchoolCount:array [1..99] of integer;
TotalSum, Answer, i, N, Ball, School:integer;
c: char;
begin
TotalSum := 0;
for i := 1 to 99 do
begin
SchoolSum[i] := 0;
SchoolCount[i] := 0
end;
Readln(N);
for i:=1 to n do
begin
repeat
Read(c)
until c=' ';
repeat
Read(c)
until c=' ';
Readln(School, Ball);
SchoolSum[School] := SchoolSum[School]+Ball;
SchoolCount[School]:=SchoolCount[School]+1;
TotalSum:=TotalSum+Ball;
end;
Answer:=0;
for i:=1 to 99 do
if SchoolSum[i]*N>TotalSum*SchoolCount[i]
then
Answer:=Answer+1;
WriteLn(Answer);
for i:=1 to 99 do
```

```
if SchoolSum[i]*N>TotalSum*SchoolCount[i]
then Write(i, ' ')
end.
```

Решение на языке C++

```
#include<iostream>
#include<string>
using namespace std;

int main()
{
int SchoolSum[100];
int SchoolCount[100];
int TotalSum=0;
int Answer = 0;
int i, N;
int Ball;
int School;
string Name;
for (i = 1; i <= 99; ++i)
SchoolSum[i] = SchoolCount[i] = 0;
cin >> N;
for (i = 0; i < N; ++i)
{
cin >> Name >> Name >> School >> Ball;
SchoolSum[School] += Ball;
SchoolCount[School]++;
TotalSum+=Ball;
}
for(i=1;i<=99;++i)
if (SchoolSum[i]*N>TotalSum*SchoolCount[i])
Answer++;
cout << Answer << endl;
for(i=1;i<=99;++i)
if (SchoolSum[i]*N>TotalSum*SchoolCount[i])
cout << i << " ";
return 0;
}
```


Указания по оцениванию	Балл
<p>Программа работает верно для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве, размер которого соответствует числу N (количество человек в списке). Программа просматривает входные данные один раз, сохраняя только информацию о количестве учащихся из каждой школы и их суммарном балле.</p> <p>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).</p>	4
<p>Программа работает верно, но входные данные запоминаются в массиве, размер которого не меньше N или входные данные считываются несколько раз. Допускается наличие от одной до трех синтаксических ошибок. Возможно, в принципиально верно организованном вводе данных есть одна ошибка. Три балла также выставляется, если в эффективной программе, удовлетворяющей критериям выставления 4 баллов, есть одна алгоритмическая ошибка (например, используется целочисленное деление вместо действительного деления при определении среднего балла).</p>	3
<p>Программа работает в целом верно, эффективно или нет, но, в реализации алгоритма содержатся до двух ошибок (неверная инициализация переменных, выход за границу массива, перевод символов в числа, используется знак "<" вместо "<=", "or" вместо "and" и т.п.). Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p>	2
<p>Программа, возможно, неверно работает при некоторых входных данных, например, неверно определяется средний балл по району, как среднее арифметическое средних баллов по школам. Возможно, неверно организовано считывание данных и отсутствует вывод результата. 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.</p>	1
<i>Задание не выполнено или выполнено неверно</i>	0

9. Некоторый поезд в пути следования останавливается на N станциях (станция номер 1 — начальная, а станция номер N — конечная). Дан список пассажиров поезда, для каждого из которых известно, на какой станции он садится, а на какой — выходит. Напишите эффективную по времени работы и используемой памяти программу, которая по этим данным определяет, на каких перегонах (то есть между какими соседними станциями) в поезде было наибольшее число пассажиров.

На вход программе в первой строке подается количество станций N и количество пассажиров P . В каждой из последующих P строк находится информация о пассажирах в следующем формате:

<Фамилия> <Имя> <станция посадки> <станция выхода>

где <Фамилия> – строка, состоящая не более, чем из 20 символов без пробелов, <Имя> – строка, состоящая не более, чем из 20 символов без пробелов, <станция посадки> и <станция выхода> — числа от 1 до N , при этом номер станции посадки меньше номера станции выхода. Пример входных данных: 5 3

Иванов Сергей 1 5

Сергеев Петр 3 5

Петров Кирилл 1 2

Программа должна вывести список перегонов, на которых в поезде было наибольшее число пассажиров. Каждый перегон выводится в виде двух последовательных номеров станций, разделенных знаком “-”. Для примера выше результат работы программы должен быть таким (на данных перегонах в поезде находилось наибольшее число пассажиров):

1-2

3-4

4-5

При выполнении задания следует учитывать, что значение N не превосходит 10, а значение P может быть большим (до 1000).

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

Решение:

Поскольку количество станций – невелико, а количество пассажиров, наоборот, — большое, то оптимальное решение не должно сохранять в памяти информацию обо всех пассажирах, запоминая только число пассажиров, которое вошло и вышло на каждой станции. Заведем массив Δ , в котором будем хранить, как изменилось число пассажиров на каждой станции. Если пассажир вошел на станции с номером a , а вышел на станции с номером b , то $\Delta[a]$ нужно увеличить на 1, а $\Delta[b]$ – уменьшить на 1 (на станции номер a общее количество пассажиров увеличивается на 1, а на станции b – уменьшается). Вместо одного массива Δ можно отдельно считать количество входящих и выходящих на каждой станции. После этого считаем количество пассажиров на каждом перегоне $Passengers[i] = Passengers[i-1] + \Delta[i]$. Затем находим наибольшее значение в массиве $Passengers$ и выводим номера

перегонов, для которых в массиве Passengers записаны наибольшие значения.

Решение на языке Паскаль

```
var
Delta: array[1..10] of integer;
Passengers: array[0..10] of integer;
i, a, b, n, p, max: integer;
c: char;
begin
for i := 1 to 10 do
Delta[i] := 0;
readln(n, p);
for i := 1 to p do
begin
repeat
read(c) until c = ' ';
repeat
read(c) until c = ' ';
readln(a, b);
Delta[a] := Delta[a] + 1;
Delta[b] := Delta[b] - 1
end;
Passengers[0] := 0;
for i := 1 to n do
Passengers[i] := Passengers[i - 1] + Delta[i];
max := 0;
for i := 1 to n - 1 do
if Passengers[i] > max then
max := Passengers[i];
for i := 1 to n - 1 do
if Passengers[i] = max then
writeln(i, '-', i + 1)
end.
```

Решение на языке C++

```
#include<iostream>
using namespace std;
int main()
{
int Delta[11], Passengers[11], i, a, b, n, p, max;
string Name;
for (i = 0; i <= 10; ++i)
Delta[i] = 0;
cin >> n >> p;
for (i = 0; i < p; ++i)
{
cin >> Name >> Name >> a >> b;
++Delta[a];
--Delta[b];
}
Passengers[0] = 0;
for (i = 1; i <= n; ++i)
Passengers[i] = Passengers[i - 1] + Delta[i];
```

```

max = 0;
for (i = 1; i < n; ++i)
if (Passengers[i] > max)
max = Passengers[i];
for (i = 1; i < n ; ++i)
if (Passengers[i] == max)
cout << i << "-" << i + 1 << endl;
return 0;
}

```

Указания по оцениванию	Балл
<p>Программа работает верно для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве, размер которого соответствует числу N (количество человек в списке). Программа просматривает входные данные один раз, сохраняя только информацию о количестве учащихся из каждой школы и их суммарном балле.</p> <p>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).</p>	4
<p>Программа работает верно, но входные данные запоминаются в массиве, размер которого не меньше P или входные данные считываются несколько раз. Допускается наличие от одной до трех синтаксических ошибок. Три балла также выставляется, если в эффективной программе, удовлетворяющей критериям выставления 4 баллов, есть одна алгоритмическая ошибка (например, результат находится правильно, но не выводится или выводится в нерерном формате).</p>	3
<p>Программа работает в целом верно, эффективно или нет, но, в реализации алгоритма содержатся до двух ошибок (неверная инициализация переменных, в частности значения максимума, выход за границу массива, перевод символов в числа, используется знак "<" вместо "<=", "or" вместо "and" и т.п.). Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p>	2
<p>Программа, возможно, неверно работает при некоторых входных данных, например, выдает правильный ответ только в том случае, если искомый перегон только один. Возможно, неверно организовано считывание данных и отсутствует вывод результата. 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.</p>	1
<p><i>Задание не выполнено или выполнено неверно</i></p>	0