

Ключи к заданиям муниципальной олимпиады для школьников по информатике 9-11 классы

Задача 1. Дорожная политика

Количество различных дорог равняется количеству способов выбрать два города из N . В комбинаторике это обозначается C^2_N и находится по формуле:

$$C^2_N = N! / (N-2)! / 2! = N * (N-1) / 2$$

Все недополученные баллы лежат в переполнении типов. Для данной задачи не хватает даже типов `longlong` и `Int64`. Поэтому в C++ следует воспользоваться `unsignedlonglong`. А в Паскале сначала поделить, а только потом умножить.

Решение на языке C++

```
#include<iostream>
#include <cstdio>
using namespace std;
int main(){
freopen("input.txt", "r", stdin);
freopen("output.txt", "w", stdout);
unsigned long long n;
cin>>n;
cout<<n*(n-1)/2;
return 0;
}
```

Решение на языке Паскаль

```
var
n : Int64;
begin
assign(input, 'input.txt'); reset(input);
assign(output, 'output.txt'); rewrite(output);
Read(n);
if n mod 2 = 0 then Write(n div 2 * (n-1))
else Write((n-1) div 2 * n);
end.
```

Задача 2. Спичрайтер Йоды

Будем считывать по одному слову и складывать их в стек. Как только в конце очередного слова замечаем точку, следует вывести все слова из стека (в обратном порядке, именно поэтому и используется стек). А считывание продолжить дальше.

Особым моментом является вывод пробелов между словами и предложениями (а также в начале и в конце строки). При тестировании задачи необходимо обратить на это внимание и, возможно, добавить условие (в решении на C++ это строки 18 и 19).

При решении на языке Паскаль дополнительной сложностью является отсутствие встроенных возможностей считывания, по словам, а также ручная реализация стека. Но количество строк в итоге, всё равно, получается меньше.

Решение на языке C++

```
#include<iostream>
#include <cstdio>
#include <cstring>
#include <stack>
using namespace std;
int main(){
freopen("input.txt", "r", stdin);
freopen("output.txt", "w", stdout);
stringstr;
stack<string>st;
```

```

bool f=false;
while(cin>>str){
if(str[str.size()-1]=='.'){
str=str.substr(0, str.size()-1);
if(f) cout<<" ";
else f=true;
cout<<str;
while(!st.empty()){
cout<<" "<<st.top();
st.pop();
}
cout<<".";
}elsest.push(str);
}
return 0;
}

```

Решение на языке Паскаль

```

var
s :AnsiString;
ms : array [0..20000] of AnsiString;
i, lm, j : Longint;
f : Boolean;
begin
assign(input, 'input.txt'); reset(input);
assign(output, 'output.txt'); rewrite(output);
ReadLn(s);
lm := 0;
f := false;
ms[0] := "";
for i:=1 to length(s) do
if s[i] = '.' then begin
if f then Write(' ')
else f := true;
Write(ms[lm]);
for j:=lm-1 downto 0 do Write(' ', ms[j]);
Write('.');
lm := -1;
end else begin
if s[i] <> '' then ms[lm] := ms[lm] + s[i]
else begin
inc(lm);
ms[lm] := "";
end;
end;
end.

```

Задача 3. Прыжки в длину

Считывание данных в этой задаче для «паскалистов» выглядит очень непросто. Тут надо уметь не только выделять слова в строке, но и исключать лишние пробелы. А также преобразовать символьное представление числа в числовой тип данных. Заступ удобно заменить на число -1 .

Для нахождения ТОП-3 при маленьких ограничениях удобно отсортировать участников. Но заметим, что поиск ТОП-3 можно осуществить и за $O(N)$.

Так при сортировки участников нам придётся сравнивать шесть их попыток в порядке убывания, то для каждого участника тоже предварительно можно отсортировать его попытки.

После этого для определения лучшего среди двух надо будет сначала сравнить шестые попытки, если они равны, то – пятые, и так далее.

В решении на C++ опять же используется стандартная сортировка с компаратором, а в решении на Паскале в обоих случаях применяется Гномья сортировка.

Обратите внимание, что в решении на Паскале сами участники не сортируются, а переставляются лишь их порядковые номера в отдельном массиве. Это немного усложняет обращение к данным, но экономит время на копировании всей информации об участнике из одной строки массива в другую.

Решение на языке C++

```
#include<iostream>
#include <cstdio>
#include <algorithm>
using namespace std;
int mm[109][6], m[109];
boolcmp(int a, int b){
int i;
for(i=5;i>=0;--i) if(mm[a][i]!=mm[b][i]) break;
return ((i>=0 && mm[a][i] > mm[b][i]));
}
int main(){
freopen("input.txt", "r", stdin);
freopen("output.txt", "w", stdout);
int n;
charss[109][3][99];
cin>>n;
for(int i=0;i<n;++i){
for(int j=0;j<3;++j) cin>>ss[i][j];
for(int j=0;j<6;++j){
char c;
cin>>c;
if(c=='x') mm[i][j]=-1;
else{
int a;
mm[i][j] = (c-'0')*100;
cin>>c>>a;
mm[i][j]+=a;
}
}
sort(mm[i], mm[i]+6);
m[i]=i;
}
sort(m, m+n, cmp);
bool f=false;
for(int i=0;i<3 && i<n;++i){
if(mm[m[i]][5]>=0){
cout<<i+1<<" " <<ss[m[i]][0]<<" " <<ss[m[i]][1]<<" " <<ss[m[i]][2]<<" " <<mm[m[i]][5]/100<<". ";
if(mm[m[i]][5]%100<10)cout<<0;
cout<<mm[m[i]][5]%100<<endl;
f=true;
}
}
}
if(!f) cout<<"No results.\n";
```

```
return 0;
}
```

Решение на языке Паскаль

```
usesmath;
var
n, i, j, c, t : Longint;
ss : array [0..109] of AnsiString;
s, s1 : AnsiString;
mm : array [0..109,0..5] of Double;
m : array [0..109] of Longint;
f : Boolean;
d : Double;
function check(a, b : Longint) : Boolean;
var
i : Longint;
begin
i := 5;
while i >= 0 do begin
if mm[a, i] <> mm[b, i] then break;
dec(i);
end;
check := (i >= 0) and (mm[a, i] < mm[b, i]);
end;
begin
assign(input, 'input.txt'); reset(input);
assign(output, 'output.txt'); rewrite(output);

ReadLn(n);
for i:=0 to n-1 do begin
ReadLn(s);
s := s + ' ';
c := 0;
j := 1;
while c<3 do begin
ss[i] := ss[i] + s[j];
inc(j);
if (s[j] = ' ') and (s[j-1] <> ' ') then inc(c);
end;
for c:=0 to 5 do begin
while s[j] = ' ' do inc(j);
s1 := "";
while s[j] <> ' ' do begin
s1 := s1 + s[j];
inc(j);
end;
if s1='x' then mm[i, c] := -1
elseval(s1, mm[i, c], t);
end;
m[i] := i;
j := 0;
while j<5 do
if mm[i, j] > mm[i, j+1] then begin
d := mm[i, j];
```

```

mm[i, j] := mm[i, j+1];
mm[i, j+1] := d;
if j>0 then dec(j);
end else inc(j);
end;
i:=0;
while i<n-1 do
if check(m[i], m[i+1]) then begin
t := m[i];
m[i] := m[i+1];
m[i+1] := t;
if i>0 then dec(i);
end else inc(i);
f := false;
for i:=0 to min(3, n)-1 do
if mm[m[i]][5] >= 0 then begin
WriteLn(i+1, ' ', ss[m[i]], ' ', mm[m[i], 5]:0:2);
f := true;
end;

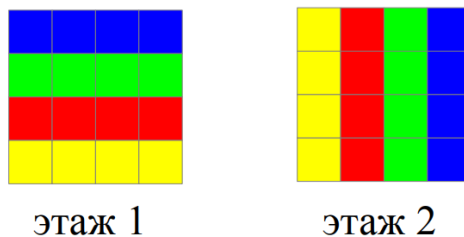
if not f then WriteLn('No results. ');
end.

```

Задача 4. Офисы

Это единственная идейная задача олимпиады. Только в этой задаче надо было что-то придумать, а не реализовать какой-либо стандартный алгоритм или формулу. Рассмотрим один из вариантов решения.

Можно заметить, что для размещения фирм достаточно всего двух этажей, если их поселить следующим образом (офисы одной фирмы покрашены в один цвет):



Все остальные этажи можно заполнить любым цветом, либо можно просто продублировать второй этаж до самого верха здания.

После заполнения осталось лишь в правильном порядке вывести трёхмерный массив.

Решение на языке C++

```

#include<iostream>
#include <cstdio>
using namespace std;
int main(){
freopen("input.txt", "r", stdin);
freopen("output.txt", "w", stdout);
int n;
cin>>n;
int mmm[26][26][26];
for(int i=0;i<n;++i) for(int j=0;j<n;++j) mmm[0][i][j]='Z'-i;
for(int i=0;i<n;++i) for(int j=0;j<n;++j) for(int k=1;k<n;++k) mmm[k][j][i]='Z'-i;
for(int i=0;i<n;++i){
for(int j=0;j<n;++j){
for(int k=0;k<n;++k) cout<<(char)mmm[i][j][k];

```

```
cout<<" ";
}
cout<<endl;
}
return 0;
}
```

Решение на языке Паскаль

```
var
n, i, j, k : Longint;
mmm : array [0..25,0..25,0..25] of Longint;
begin
assign(input, 'input.txt'); reset(input);
assign(output, 'output.txt'); rewrite(output);
Read(n);
for i:=0 to n-1 do for j:=0 to n-1 do
mmm[0, i, j] := ord('Z') - i;
for i:=0 to n-1 do for j:=0 to n-1 do for k:=1 to n-1 do
mmm[k, j, i] := ord('Z') - i;
for i:=0 to n-1 do
for j:=0 to n-1 do begin
for k:=0 to n-1 do Write(chr(mmm[i, j, k]));
WriteLn;
end;
end.
```