

**Межрегиональная предметная олимпиада КФУ
по предмету «Информатика»
заключительный этап (решения)
2018-2019 учебный год
11 класс**

№1. Кубик. (25 баллов)

На вертикальной клетчатой доске размером N (высота) на M (ширина) клеток находится магнитная игральная кость (кубик). Она расположена в левой верхней клетке (с координатами $1, 1$) так, что ее грань на доске занимает ровно одну клетку, т.е. каждая грань кости равна по размерам одной клетке. Напротив грани кости с числом 1 написано число 6 , напротив числа 2 находится грань с числом 5 и напротив числа 3 написано 4 . За один ход кость перекачивается через ребро грани на доске (перекаптовывается, а не сдвигается) в соседнюю клетку, не выходя за рамки листа, и занимает всю соседнюю клетку. Теперь одна из боковых граней находится на доске. При этом ведется подсчет очков тех граней, которые соприкасаются с клеткой, т.е. на которую примагничена кость в данный момент. Вычислить и напечатать минимальную сумму очков, которая наберется, если прокатить игральную кость в правый нижний угол листа (с координатами N, M). Начальное расположение кости не известно. Исходные данные – натуральные числа N и M – размеры листа ($1 < N, M < 100$).

Входные данные:	Выходные данные
2 2	6

Решение/

Похожая, но более общая, задача имеется на сайте acm.timus.ru (номер 1016 «кубик на прогулке»). В ней на каждой грани может быть записано произвольное натуральное число.

Для ее решения представим матрицу как неориентированный граф с весами ребер, где вершина – это каждая клетка с учетом текущего положения кубика (т.е. с учетом поворота граней), а ребро соединяет смежные вершины (клетки) и имеет вес, равный числу на новой нижней грани кубика. Дальше, можно использовать один из известных алгоритмов поиска кратчайшего пути из начальной вершины в конечную – алгоритм Дейкстры, Форда-Беллмана, поиск в ширину, поиск в глубину и другие. Так как начальное положение граней кубика не известно (какая грань внизу, какие по бокам), следует выполнить этот процесс для каждого варианта расположения кубика.

В нашем, более простом случае, можно заметить, что положение кубика повторяется после нескольких (сколько точно?) поворотов по горизонтали и по вертикали. Остается рассмотреть небольшой набор значений для n и m .

Баллы

- 25 баллов за полное и оптимальное решение.
- 15 баллов за неполное решение.
- 8 баллов за наличие идей решения

№2. Выборы. (10 баллов)

В выборах Президента Страны участвуют N кандидатов. Проведенный социологический опрос показал, что за i -го кандидата хотят проголосовать A_i избирателей. Кандидат номер 1 хочет победить наверняка и решается подкупить некоторое число избирателей, чтобы они проголосовали за него. Для подкупа одного любого избирателя нужно дать ему одну конфетку. Какое минимальное число конфеток должен приготовить Кандидат номер 1 , чтобы набрать голосов больше, чем любой другой кандидат. $1 \leq N \leq 1000$, $A_i < 1000$.

В первой строке задано N и во второй строке N целых чисел, перечисленных через пробел, – количество избирателей голосующих за кандидатов.

Входные данные:	Выходные данные
5 2 11 7 14 13	9

Решение.

```

Var i,n,j,k,a1,b:integer;
    A:array[2..1000] of integer;
Begin
    Read(n,b); a1:=b;
    For i:=2 to n do read(A[i]);
    For i:=n downto 3 do
        For j:=2 to i-1 do
            Begin k:=a[j];a[j]:=a[j-1]; a[j-1]:=k end;
        K:=2;r:=a[2];
    While a[1]<=A[k] do begin
        A[k]-=1;
        a1+=1;
        k+=1;
        if (a[k]<r) or (k>n) then k:=2
    End;
    Write(a1-b)
End.

```

Баллы

- 10 баллов за полное и оптимальное решение.
- 7 баллов за неполное решение.
- 3 баллов за наличие идей решения.

№3. Контрразведка. (15 баллов)

Пять разведывательных групп хранили общую секретную информацию в одной камере хранения. Каждая группа шифр записывала в своей системе исчисления. Первая группа в двоичной системе, 2-ая группа – 4-чной системе, 3-я группа – 8-чной системе, 4-я группа – 16-чной системе, 5-ая – в 32-чной. Контрразведчики, наблюдая за разведчиками, узнали некоторые цифры шифра.

Вывести минимальный и максимальный возможные коды, которые нужно перебрать при вскрытии сейфа, чтобы получить доступ к информации. Ответ выдать в десятичном виде.

В 32-чной системе используются цифры 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V.

В четырех строчках приведены обнаруженные коды контрразведчиков. Длина кодов не превышает 200 символов. Неизвестные разряды кода задаются символом ”*”.

Входные данные:	Выходные данные
* 0***** *****	450
*****2	1478
**0*	
8	
E	

Решение.

```

Var c32,c2,c4,c8,c16:string;
    i,k,n,d:integer; min,max:int64;

```

```

r4:array['0'..'3'] of string=('00','01','10','11');
r8:array['0'..'7'] of string=('000','001','010','011','100','101','110','111');
r16:array['0'..'9','A'..'F'] of string=
    ('0000','0001','0010','0011','0100','0101','0110','0111',
     '1000','1001','1010','1011','1100','1101','1110','1111');
r32:array['0'..'9','A'..'V'] of string=
    ('00000','00001','00010','00011','00100','00101','00110','00111',
     '01000','01001','01010','01011','01100','01101','01110','01111',
     '10000','10001','10010','10011','10100','10101','10110','10111',
     '11000','11001','11010','11011','11100','11101','11110','11111');

```

Begin

```

Readln(c2);
Readln(c4);
Readln(c8);
Readln(c16);
Readln(c32);
N:=max(max(max(length(c2),2*length(c4)),max(3*length(c8),4*length(c16))),5*length(c32));
C2[1]:= '1'; // старший значащий разряд – '1'
d:=length(c2); // число 2-ичных значащих разрядов
For i:= d+1 to n do c2:='*' + c2;
// разбор c4
For i:=length(c4) downto 1 do
    If c4[i]<>'*' then begin
        C2[n-2*(i-1)]:=r[c4[i]][1];
        C2[n-2*(i-1)-1]:=r[c4[i]][2]
    End;
// аналогично разбор c8
// аналогично разбор c16
// аналогично разбор c32
Min:=0;max:=0;;
For i :=length(c2) downto n-d+1 do
    If c2[i]='*' then begin min*:=2; max:=max*2+1 end
    Else begin min:=min*2+a[i]; max:=max*2+a[i] end
WriteLn(min);
Write(max)
End.

```

Баллы

- 15 баллов за полное и оптимальное решение.
- 10 баллов за неполное решение.
- 5 баллов за наличие идей решения.

№4. Лабиринт. (20 баллов)

Дан лабиринт размерности $N \times N$, по которому можно ходить направо или вниз. Из клетки со звездочкой «*» можно делать телепортацию в любую клетку по диагонали, горизонтали или вертикали к выходу, где нет препятствий при перемещении (с клетки А,В в клетку А+К,В+К или А+К,В или А,В+К). Написать программу, которая по заданному расположению стенок и проходов в клетках определит число путей, ведущих из левого верхнего входа к нижнему правому выходу.

Входные данные. В первой строке задается N ($N < 100$). Далее в N строках заданы строки лабиринта, где клетка телепортации задается «*», проходная клетка «.», а не проходная «X».

Входные данные:	Выходные данные
4	6
.*..	
.X..	
.XX.	
....	

	*		

Решение.

```

Var i,j,k,:integer;
    L:array[1..100,1..100] of char;
    P:array[0..100,0..100] of integer;
Begin
    Read(n);
    For i:=0 to 100 do begin P[i,0]:=0; P[0,i] end;
    For i:= to n do begin
        For j:=1 to n do begin read(L[i,j]); P[i,j]:=0 end;
        Readln
    End;
    For i:=1 to n do
        For j:= to n do
            If i+j=2 then P[I,j]:=1 else
            If L[i,j]<>'X' then begin
                P[I,j] +=P[i-1,j]+P[I,j-1];
                K:=2;
                While (k<min(i,j)) and (L[i-k,j-k]<>'X') do begin
                    If L[i-k,j-k]='*' then P[i,j]+=P[i-k,j-k];
                    K+=1
                end;
                K:=2;
                While (k<i) and (L[i-k,j]<>'X') do begin
                    If L[i-k,j]='*' then P[i,j]+=P[i-k,j];
                    K+=1
                end;
                K:=2;
                While (k<j) and (L[i,j-k]<>'X') do begin
                    If L[i,j-k]='*' then P[i,j]+=P[i,j-k];
                    K+=1
                end
            End;
        Write(P[n,n])
    End.

```

Баллы

- 20 баллов за полное и оптимальное решение.
- 13 баллов за неполное решение.
- 7 баллов за наличие идей решения.

№5. Слоги. (15 баллов)

Задан текст на татарском языке.

В татарском языке 6 типов слогов: Г, ГС, СГ, ГСС, СГС, СГСС, где Г – гласная буква, С – согласная буква. Если в слове перед гласной буквой есть согласная, то выбирается слог с этой согласной буквой в начале. Например: артык – ар-тык, артта – арт-та.

Написать программу, которая для этого текста выведет количества слогов каждого типа.

Входные данные:	Выходные данные
Тук, тук – тукрыан тукулдатып тукуган.	Г 0 ГС 0 СГ 6 ГСС 0 СГС 6 СГСС 0

Решение.

```
#import <bits/stdc++.h>
using namespace std;
main(){
    int i,j,dl=0,kg=0,kgs=0,ksg=0,kgss=0,ksgs=0,ksgss=0;
    char s[20];
    while (cin >> s) {
        int c=0, d;
        d=strlen(s)-1+alpha(s[strlen(s)-1]);
        while (d>=0) {
            if (s[d]=='a' | s[d]=='A' | ...){ // проверка на гласность – Г, СГ
                if (d=0 | d>0 & (s[d-1]=='a' | s[d-1]=='A' | ...)) { // – Г
                    kg++;
                    d--;
                } else { // СГ
                    ksg++;
                    d-=2;
                }
            }
            else { // ГС, ГСС, СГС, СГСС
                d--; // ГС(Г), ГСС(ГС), СГС(СГ), СГСС(СГ)
                if (s[d]=='a' | s[d]=='A' | ...){ // проверка на гласность – ГС(Г), СГС(СГ)
                    if (d=0 | d>0 & (s[d-1]=='a' | s[d-1]=='A' | ...)) { // – ГС
                        kgs++;
                        d--;
                    } else { // СГС
                        ksgs++;
                        d-=2;
                    }
                }
            }
            else { // ГСС(ГС), СГСС(СГ)
                d--; // ГСС(Г), СГСС(СГ)
                if (d=0 | d>0 & (s[d-1]=='a' | s[d-1]=='A' | ...)) { // – ГСС
                    kgss++;
                    d--;
                } else { // СГСС
                    ksgss++;
                    d-=2;
                }
            }
        }
    }
}
```

```

cout << "Г " << kg
    << "/nГС " << kgs
    << "/nСГ " << ksg
    << "/nГСС " << kgss
    << "/nСГС " << ksgs
    << "/nСГСС " << ksgss;
}

```

Баллы

- 15 баллов за полное и оптимальное решение.
- 10 баллов за неполное решение.
- 5 баллов за наличие идей решения.

№6. Раз, два... (15 баллов)

Сколько N-значных чисел можно составить, используя цифры 1 и 2, в которых три одинаковые цифры не стоят рядом? Вход – число N ($1 \leq N \leq 30$).

Выход – нужно вывести одно число – количество чисел с указанным свойством.

Входные данные:	Выходные данные
4	10

Решение.

Простой анализ для небольших значений N показывает, что ответ равен удвоенному очередному числу Фибоначчи. Например, для N = 1 ответ 2, для N = 2 ответ 4, для N = 3, ответ 6. Последовательность чисел Фибоначчи 1, 2, 3, 5, 8, 13, 21,... строится по простому принципу: первые два числа задаются, а каждое следующее вычисляется (оно равно сумме двух предыдущих).

```

#include <iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    int f1=0, f2=1;
    for(int i=0;i<n;i++)
    {
        swap(f1, f2);
        f2+=f1;
    }
    cout<<f2*2;
    return 0;
}

```

2-ой вариант. Перебор всех N-разрядных чисел с цифрами 1, 2. Далее подсчет количества чисел, не содержащих 3 подряд идущих единиц или двоек.

```

Var s:string;
    l,k:integer;
Begin
    Read(n);
    sC:='';
    For i:=1 to n do s:=s+'1'
    k:=0;

```

```
Repeat
  i:=n;
  while (s[i]='2') and (i>0) do begin
    s[i]:=1;
    i:=i-1;
  end;
  if i>0 then s[i]:='2';
  k+=(pos('111',s)=0) and (pos('222',s)=0);
Until i=0;
Write(k);
End.
```

Баллы

- 15 баллов за полное и оптимальное решение.
- 10 баллов за неполное решение.
- 5 баллов за наличие идей решения.