

Олимпиадное задание по информатике

Задача 1. Вес 25 баллов

Можно ли разменять 25 рублей 9 купюрами достоинством в 1, 3 и 5 рублей? Вывести на экран все допустимые комбинации купюр. Пример вывода:

1 куп. по 1 руб, 8 куп. по 3 руб, 0 куп. по 5 руб

Если допустимой комбинации нет, вывести соответствующее сообщение.

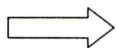
Задача 2. Вес 35 баллов

Матрица символьного типа размером $M \times N$ заполняется английскими буквами из диапазона (A,Z) с помощью датчика случайных чисел. Значения M и N вводит пользователь с клавиатуры. Вывести построчно на экран исходную матрицу. В матрице заменить символом * все повторные появления других символов. Вывести построчно на экран измененную матрицу. Исключить из матрицы все строки, содержащие только символ *, создав новую матрицу. Новую матрицу построчно вывести на экран. Если таких строк нет, вывести соответствующее сообщение.

Примеры заполнения матрицы размером 5×5 , замены символов и исключения строк приведены на следующем рисунке.

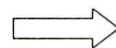
Исходная матрица

D	K	J	I	N
B	N	D	B	G
N	I	A	Z	J
A	B	C	D	I
N	J	C	G	C



Пример замены буквы D

D	K	J	I	N
B	N	*	B	G
N	I	A	Z	J
A	B	C	*	I
N	J	C	G	C



Замена всех повторений

D	K	J	I	N
B	*	*	*	G
*	*	A	Z	*
*	*	C	*	*
*	*	*	*	*

Исключение строки:

D	K	J	I	N
B	*	*	*	G
*	*	A	Z	*
*	*	C	*	*

Задача 3. Вес 40 баллов

В выражении

$(((((1 ? 2) ? 3) ? 4) ? 5) ? 6$

вместо каждого знака ? поставить знак одной из четырех операций: +, -, *, /,

так, чтобы результат вычислений был равен N. Значение N ввести с клавиатуры. Выведите на экран все допустимые комбинации операций, например, для $N=35$:

$(((((1 * 2) * 3) * 4) + 5) + 6 = 35$

и т.д.

Если допустимой комбинации нет, вывести соответствующее сообщение.

Примечание: / – операция целочисленного деления;

* – операция умножения.

```
# Хакимов, задача 1
money = float(input()) # вводим сумму
n = 9 # создаём переменную кол-ва купюр
changeable = False # создаём переменную, которая показывает нашли ли способ
разменять сумму или нет

while money <= 0 or money % 1 != 0: # проверяем правильность входных данных,
если они неверны - вводим снова
    if money <= 0:
        print('Введена неположительная сумма. Введите сумму снова.')
        money = float(input())
    else:
        print('Введена нецелая сумма. Введите сумму снова.')
        money = float(input())
money = int(money)

for i in range(money // 5): # Ищем способы разменять данную сумму. Если нашли -
выводим
    for j in range(n - i):
        if i * 5 + j * 3 + n - i - j == money:
            print(n - i - j, 'куп. по 1 рублю,', j, 'куп. по 3 рубля,', i, 'куп.
по 5 рублей.')
            changeable = True

if not changeable: # Если разменять не получилось выводим сообщение о
невозможности разменять сумму
    print('Данную сумму разменять нельзя')
```

06.02.2015

Хакимов Хакимов Артем Игоревич

```

# Хакимов, задача 2
import random # Импортруем библиотеку для генерации случайных чисел

M = int(input())
N = int(input()) # Вводим количество столбцов и строк в матрице

while N <= 0 or M <= 0: # Проверяем правильность входных данных
    print('Введено неположительное число. Повторите ввод')
    M = int(input())
    N = int(input())

matrix = [] # Создаём пустую матрицу
for i in range(N):
    matrix.append([])

for i in range(N): # Заполняем матрица случайными буквами от A до Z
    for j in range(M):
        matrix[i].append(chr(random.randint(65, 90)))

for i in range(N): # выводим матрицу
    print(matrix[i])

for i in range(65, 91): # Заменяем повторные символы на звёздочки
    letter = chr(i)
    first_time = True
    for a in range(N):
        for b in range(M):
            if matrix[a][b] == letter:
                if first_time:
                    first_time = False
                else:
                    matrix[a][b] = '*'

new = [] # Создаём новую матрицу, в которой нет строк со звёздочками
for i in range(N):
    need = False
    for j in range(M):
        if matrix[i][j] != '*':
            need = True
            break
    if need:
        new.append(matrix[i])

print() # Отступаем строчку и выводим новую матрицу
for i in range(len(new)):
    print(new[i])

```

06.02.2015

Хакимов

Хакимов Артём Игоревич

```

# Хакимов, задача 3
N = int(input()) # Вводим данные
possible = False # Объявляем переменную, которая проверяет найдена ли нужная комбинация знаков

while N <= 0:
    print('Введено неположительное число. Повторите ввод')
    N = int(input())

for a in range(4): # Находим нужные комбинации путём перебора (5 операций, по 4 варианта на каждую)
    for b in range(4):
        for c in range(4):
            for d in range(4):
                for e in range(4):
                    answer = 1
                    actions = [a, b, c, d, e]
                    for i in range(5):
                        if actions[i] == 0:
                            answer += (i + 2)
                        if actions[i] == 1:
                            answer -= (i + 2)
                        if actions[i] == 2:
                            answer *= (i + 2)
                        if actions[i] == 3:
                            answer /= (i + 2)
                    if answer == N:
                        for i in range(5):
                            if actions[i] == 0:
                                actions[i] = '+'
                            if actions[i] == 1:
                                actions[i] = '-'
                            if actions[i] == 2:
                                actions[i] = '*'
                            if actions[i] == 3:
                                actions[i] = '/'
                        print('((( (1', actions[0], '2)', actions[1], '3)',
actions[2], '4)', actions[3], '5)', actions[4], '6 =', N)
                        possible = True

if not possible: # Если комбинация не найдена, выводим сообщение об этом
    print('Знаки подобрать невозможно')

```

06.02.2015

Хакимов

Хакимов Артём Игоревич

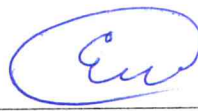
```
//Елисеев, задача 1
#include <iostream>

using namespace std;

int main()
{
    setlocale(0, "");
    int mass[9]={0, 0, 0, 0, 0, 0, 0, 0, 0}; //будем генерировать девятизначные числа в
    троичной системе, в этом массиве хранится промежуточный результат

    for(; mass[0]<3; mass[8]++)//перебор девятизначных чисел в троичной системе
    {
        for(int j=8; j>0; j--)
            if(mass[j]==3){
                mass[j]=0;
                mass[j-1]++;
            }
        //ставим в соответствие 0 1 2 с 1 3 5 и получаем все комбинации девяти купюр 1
        3 5 руб. проверяем, чтобы их сумма давала 25, тогда выводим
        int sum=0, m1=0, m3=0, m5=0;//сумма купюр, количество купюр достоинством 1 3 5
        руб соответственно
        for(int q=0; q<9; q++){
            if(mass[q]==0) {sum+=1; m1++;}
            if(mass[q]==1) {sum+=3; m3++;}
            if(mass[q]==2) {sum+=5; m5++;}
        }
        if(sum==25) cout<<m1<<" куп. по 1 руб, "<<m3<<" куп. по 3 руб, "<<m5<<" куп.
        по 5 руб"<<endl;//выводит долго, тк cout время занимает некоторое
    }

    system("pause");
    return 0;
}
```



```
//Елисеев, задача 2
#include <iostream>
#include <map>

using namespace std;

int main()
{
    setlocale(0, ""); //Русский язык в консоли
    char matrix[1000][1000]={{'\0'}}; //Массив для хранения матрицы
    int m, n; //Переменные, для хранения размера матрицы, задаваемый пользователем
    map <char, int>lett; //Ассоциативный массив для упорядоченного хранения элементов и их количества матрицы
    bool exit=false, error=false; //Для выхода из программы. Для обработки ошибки

    while(!exit){

        //Задаем размеры матрицы
        cout<<"Введите количество строк в матрице:";
        cin>>m;
        cout<<"Введите количество столбцов в матрице:";
        cin>>n;
        if(n<1 && m<1){
            error=true;
            cout<<"Ошибка ввода. Ширина и высота должны быть положительны и больше 0"<<endl<
        <endl;
        }

        //Заполняем матрицу и массив
        if(!error){
            cout<<"Введите матрицу шириной "<<n<<" и высотой "<<m<<" строчек. Вводить только латинские буквы. Все элементы будут переведены в верхний регистр"<<endl;
            for(int i=0; i<m; i++){
                for(int j=0; j<n; j++){
                    cin>>matrix[i][j];
                    if(!(matrix[i][j]>='a' && matrix[i][j]<='z' && matrix[i][j]>='A' && matrix[i][j]<='Z')) error=true; //Проверка на латинскую букву
                    if(matrix[i][j]>='a' && matrix[i][j]<='z') matrix[i][j]+='A'-'a'; //Перевод в верхний регистр
                    lett[matrix[i][j]]=0;
                }
            }
            if(error) cout<<"Ошибка. Один или несколько элементов матрицы являются не латинскими буквами"<<endl<<endl;

            //Выводим матрицу
            if(!error)
            cout<<"Исходная матрица (Верхний регистр автоматический):"<<endl;
            for(int i=0; i<m && !error; i++){
                for(int j=0; j<n; j++) cout<<matrix[i][j]<<" ";
                cout<<endl;
            }

            //Заменяем некоторые элементы *, и выводим+маркеруем строчки, полностью состоящие из *
            if(!error){
                cout<<"Все повторяющиеся буквы заменены *:"<<endl;
                for(int i=0; i<m; i++){
                    bool star=true; //Если в строке все *, останется true

                    for(int j=0; j<n; j++){
                        if(lett[matrix[i][j]]>0) matrix[i][j]='*'; //Если эту букву уже выводили, заменяем ее *
                        cout<<matrix[i][j]<<" ";
                        lett[matrix[i][j]]++; //Считаем сколько раз выводили эту букву

                        if(matrix[i][j]!='*') star=false;
                    }
                    if(star) matrix[i][0]='1'; //Маркеруем, эту строку не будем выводить
                    cout<<endl;
                }
            }
        }
    }
}
```

```
//Выводим матрицу, без маркерowanych строчек
cout<<"Все строки, полностью состоящие из *, не выведены"<<endl;
for(int i=0; i<m; i++){
    for(int j=0; j<n && matrix[i][j]!='1'; j++){
        cout<<matrix[i][j]<<" ";
    }
    cout<<endl;
}

//Выход
int q=2;
while(q!=0 && q!=1){
    cout<<endl<<"Хотите выйти или поввторить ввод еще раз? 0-выйти, 1-еще раз:";
    cin>>q;
    if(q==0) exit=true;
    else if(q!=1) cout<<"Ошибка";
    cout<<endl;
}
}
return 0;
}
```

Eu

```
//Елисеев, задача 3
#include <iostream>

using namespace std;

int main()
{
    setlocale(0, "");
    int mass[5]={0, 0, 0, 0, 0}, n=0;
    char bal[4]={'+', '-', '*', '/'};

    cin>>n;

    for(; mass[0]<5; mass[4]++)
    {
        for(int j=4; j>0; j--)
            if(mass[j]==4){
                mass[j]=0;
                mass[j-1]++;
            }

        int sum=1;
        for(int q=0; q<5; q++){
            if(mass[q]==0) sum+=q+2;
            if(mass[q]==1) sum-=q+2;
            if(mass[q]==2) sum*=q+2;
            if(mass[q]==3) sum/=q+2;
        }
        if(sum==n) {
            cout<<" ((( (1" << bal[mass[0]] << "2)" << bal[mass[1]] << "3)" << bal[mass[2]] << "4)" <<
            < bal[mass[3]] << "5)" << bal[mass[4]] << "6" << endl; }
    }

    system("pause");
    return 0;
}
```



```

{Дьяков, задача 1}
program nom1;
var k5,k3,k1:integer; f1:boolean; {k5-количество купюр по 5 руб,
                                  k3-количество купюр по 3 руб,
                                  k1-количество купюр по 1 руб,
                                  f1-проверка того, что комбинация
найдена}

begin

f1:=true;
for k5:=0 to 5 do {перебираем все возможные варианты
составить 25}
  for k3:=0 to ((25 -k5*5) div 3) do
    begin
      k1:=25-k5*5-k3*3;
      if (k1+k3+k5)=9 then
        begin
          f1:=true;
          writeln(k1, ' куп. по 1 руб, ', k3, ' куп. по 3 руб, ', k5, ' куп.
по 5 руб');
          end;
        end;
      if f1=false then writeln('допустимой комбинации нет');
    end.
end.

```



1

```

{Дьяков, задача 2}
program nom2;
var a, a1: array [1..100,1..100] of char; {матрицы}
i, j, i1, j1, m, n, i2, j2: integer;
mn: set of char;
fl: boolean;
begin
writeln('Ввод данных');
write('N=');
readln(n);
write('M=');
readln(m);
if (m<1) or (n<1) then writeln('некорректные входные данные')
else begin
writeln('исходная матрица');
writeln;
for i:=1 to n do
begin
for j:=1 to m do
begin
a[i, j]:=chr(65+random(20));
write(a[i, j], ' ');
mn:=mn+[a[i, j]];
end;
writeln;
end;
writeln;

for i:=1 to n do

for j:=1 to m do
begin

if not (a[i, j]='*') then
begin
if j<m then begin j1:=j+1; i1:=i end
else begin i1:=i+1; j1:=1; end;

for i2:=1 to n do

for j2:=1 to m do
if (a[i, j] =a[i2, j2]) then
if not ((i=i2) and (j=j2)) then a[i2, j2]:='*';
end;
end;

writeln('Замена всех позиций');
writeln;
for i:=1 to n do
begin
for j:=1 to m do
write(a[i, j], ' ');
writeln;
end;
writeln;

i1:=1;
for i:= 1 to n do
begin
fl:=true;

```



```
j:=1;
while (fl= true) and (j<=m) do
  if not( a[i,j]='*') then fl:=false
  else inc(j);
if fl= false then
begin
  for j:=1 to m do
    al[i1,j]:=a[i,j];
  inc(i1);
end;
end;
writeln('исключение всех строк');
writeln;
for i:=1 to i1 do
begin
  for j:=1 to m do
    write(a[i,j], ' ');
writeln;
end;
end;
readln;
end.
```



```

{Дьяков, задача 3}
program n3;
var
n:integer;
i, z1, z2, z3, z4, z5:integer;
fl:boolean;
begin
writeln('введите n');
readln(n);
fl:=false;
for z1:=1 to 4 do
for z2:=1 to 4 do
for z3:=1 to 4 do
for z4:=1 to 4 do
for z5:=1 to 4 do
begin
i:=1;
case z1 of
1: i:=i+2;
2: i:=i-2;
3: i:=i*2;
4: i:=i div 2;
end;

case z2 of
1: i:=i+3;
2: i:=i-3;
3: i:=i*3;
4: i:=i div 3;
end;

case z3 of
1: i:=i+4;
2: i:=i-4;
3: i:=i*4;
4: i:=i div 4;
end;

case z4 of
1: i:=i+5;
2: i:=i-5;
3: i:=i*5;
4: i:=i div 5;
end;

case z5 of
1: i:=i+6;
2: i:=i-6;
3: i:=i*6;
4: i:=i div 6;
end;
if n=i then begin
fl:=true;
write('(((1)');
case z1 of
1: write('+');
2: write('-');
3: write('*');
4: write('/');
end;
end;

```

```
write('2');
case z2 of
1: write('+');
2: write('-');
3: write('*');
4: write('/');
end;
write('3');
case z3 of
1: write('+');
2: write('-');
3: write('*');
4: write('/');
end;
write('4');
case z4 of
1: write('+');
2: write('-');
3: write('*');
4: write('/');
end;
write('5');
case z5 of
1: write('+');
2: write('-');
3: write('*');
4: write('/');
end;
writeln('6=',n);
end;

end;
if fl=false then writeln('не возможно');
readln;
end.
```