# Постановка задачи

Существуют дискеты, стоимостью 11 рублей 50 копеек каждая. Существуют коробки, одна коробка вмещает в себя 12 дискет и стоит 114 рублей 50 копеек. Существуют ящики, один ящик стоит 1255 рублей и вмещает 12 коробок. Стоимость дискеты, коробки, ящика представлена в виде констант. Вывести:

1. Стоимость дискет, купленных по отдельности.
2. Количество купленных коробок и ящиков, при сохранении общего количества дискет. Вывести потраченную сумму.
3. Потратить как можно меньше, при покупке ящиков, коробок, при сохранении общего количества дискет (учесть, что коробка вмещает в себя 12 дискет, ящик вмещает в себя 12 коробок, коробка вмещает в себя12 дискет). Вывести потраченную сумму, а также количество коробок/ящиков/дискет. В этом же пункте вывести количество бонусных дискет (в этом пункте, при использовании наивыгоднейшей суммы вы купите больше дискет, чем в пункте 1)).

Приложение А

(обязательное)

Исходный код программы

Program FloppyDisk;

{Программа подсчитывает сумму дискет, купленных по отдельности, после подсчитывает количество ящиков, коробок, дискет, которые можно купить с учетом начального количества дискет, а также «выгодности» в их приобретении. Подсчет количества бонусных дискет. Подсчёт количества сэкономленных денег. Вывод полученных значений}

//Консольное приложение

{$APPTYPE CONSOLE}

//Вызов системного модуля

Uses

SysUtils;

//Объявление вещественных констант

Const

floppycost :real = 11.5; // floppycost-цена дискеты

costofboxes :real = 114.5;// costofboxes-цена коробки

costofchest:real = 1255; // costofchest-цена ящика

Var

ns:string;

N,Boxe,Chest,k,n1:Integer;

FirstSt,SecondSt,e:real;

//ns-количество дискет, введенных как строка,

//boxe-количество ящиков, chest- количество ящиков

//n-количество дискет,k-метка для проверки

//перевода строки в число,n1-количество

//дискет, FirstSt-стоимость дискет

//SecondSt-выгодная стоимость2, e-выгодная стоимость1

Begin

//Ввод числа

//Проверка ввода числа

Write('Enter the NATURAL number of the floppy disk:

');

Readln(ns);

Val(ns,n,k);

while (k<>0) or (n<=0) do

begin

Val(ns,n,k);

if (k<>0) or (n<=0) then

begin

Writeln('Incorrect input');

Writeln('Try again');

Writeln('Enter the NATURAL POSITIVE number of

the floppy disk: ');

Readln(ns);

end

else

k:=0;

end;

//Вычисление стоимости дискет,

//купленных по отдельности

//вывод полученных значений

FirstSt:=N\*floppycost;

WriteLn('1)When buying floppies separately, their

cost will be: ' + Floattostr(FirstSt) + '

rub');

WriteLn;

//Вычисление количества коробок, ящиков,

//оставшихся дискет

//вычисление полной стоимости, купленных ящиков,

//дискет, коробок

//сохранение количества дискет

n1:=n;

Chest := N div 144;

N:= N mod 144;

boxe:= N div 12;

N:= N mod 12;

e:=floppycost\*N+costofboxes\*boxe+costofchest\*chest;

//Вывод количества приобретенных товаров

//и стоимости приобретенных товаров

Writeln('2)We can bought:');

Writeln(' Chest - ' + inttostr(chest));

Writeln(' Boxe - ' + inttostr(boxe));

Writeln(' Floppy disk - ' + inttostr(N));

Writeln('----------------');

Writeln('Cost: '+floattostr(e)+' rub ');

WriteLn;

//Если выгодно купить коробку, покупка коробки,

//вместо дискет

if N\*floppycost >= costofboxes then

begin

N:=0;

boxe:=boxe+1;

end;

//Если выгодно купить ящик, покупка ящика,

//вместо дискет и коробок

if boxe\*costofboxes >= costofchest then

begin

boxe:=0;

N:=0;

chest:=chest+1;

end;

//Подсчет наивыгоднейшего варианта

//Если вторая сумма меньше первой, то вывод полученных

//значений

//Подсчет бонусных дискет

SecondSt:=floppycost\*N+costofboxes\*boxe+

costofchest\*chest;

if e>secondst then

begin

Writeln('For most economy you can bought: ');

Writeln(' Chest - ' + inttostr(chest));

Writeln('Boxe - ' + inttostr(boxe));

Writeln(' Floppy disk - ' + inttostr(N));

Writeln('----------------');

Writeln('3)Cost: '+ floattostr(SecondSt)+' rub');

WriteLn;

Writeln('And you have ',abs(n1-

(boxe\*12+(chest\*12\*12))),'floppy disk as

bonus');

end

else

writeln ('3)No other variants ');

Writeln;

Writeln('When using the most advantageous option,

you will save: ' + floattostr(FirstSt

* SecondSt) + ' rub');

Readln;

End.

Приложение Б

(обязательное)

Тестовые наборы

Тестовая ситуация для проверки ввода данных

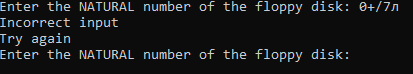
Тестовая ситуация: проверка, сообщит ли программа о некорректном вводе, если введено не число

Исходные данные: 0/+7л

Ожидаемый результат:



Полученный результат:

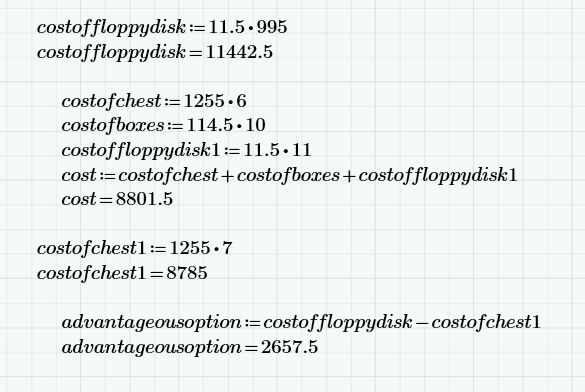


Тестовая ситуация для проверки вычислений 1

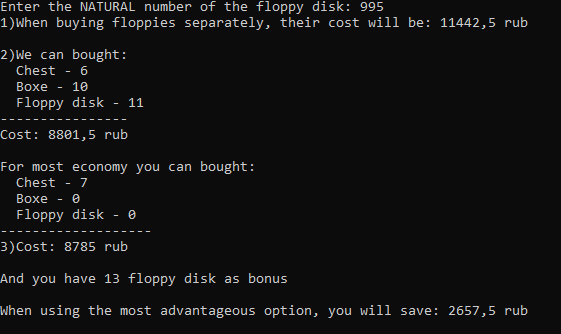
Тестовая ситуация для проверки вывода правильных решений при количестве дискет = 995

Исходные данные: n=995

Ожидаемый результат:



Полученный результат:

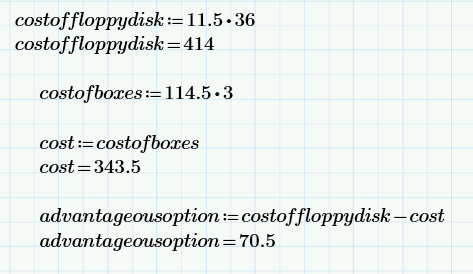


Тестовая ситуация для проверки вычислений 2

Тестовая ситуация для проверки вывода правильных решений при количестве дискет = 36

Исходные данные: n=36

Ожидаемый результат:



Полученный результат:

