Содержание

[1 Постановка задачи 3](#_Toc116916278)

[1.1 Исходное условие задачи 3](#_Toc116916279)

[1.2 Модифицированное условие задачи 3](#_Toc116916280)

[1.3 Исходные данные 3](#_Toc116916281)

[2 Структура данных 5](#_Toc116916282)

[3 Детали и методика решения задачи 6](#_Toc116916283)

[3.1 Детали задачи 6](#_Toc116916284)

[3.2 Используемые формулы 6](#_Toc116916285)

[3.3 Проверка исходных данных с помощью цикла с постусловием 7](#_Toc116916286)

[3.4 Использование Try … Except … End; 9](#_Toc116916287)

[4 Результаты расчетов 10](#_Toc116916288)

[Приложение А 11](#_Toc116916289)

[Приложение Б 17](#_Toc116916290)

# Постановка задачи

## Исходное условие задачи

Есть определенное количество дискет в коробке и определенное количество коробок в ящике; есть определенная стоимость одной дискеты, коробки и ящика. Вводится количество дискет, необходимых к покупке. Вывести вариант покупки необходимого количества дискет за самую низкую стоимость, если при покупке 11 дискет выгоднее купить одну коробку, чем 11 дискет по отдельности.

## Модифицированное условие задачи

Вводится количество дискет в коробке D[2] и в ящике D[3]; стоимость одной дискеты CD[1], коробки CD[2] и ящика CD[3]. Вводится количество дискет N, необходимых к покупке. Вывести вариант покупки необходимого количества дискет за самую низкую стоимость, если при покупке 11 дискет выгоднее купить одну коробку, чем 11 дискет по отдельности.

## Исходные данные

Программа получает от пользователя:

* количество дискет в коробке;
* количество дискет в ящике;
* стоимость одной дискеты;
* стоимость коробки;
* стоимость ящика;
* необходимое к покупке количество дискет.

Вывод данных происходит в следующем виде:

1. Способ покупки заданного количества дискет по наименьшей стоимости.
2. Выгода в стоимости при способе покупки заданного количества дискет с помощью нескольких типов (единичные дискеты, коробки, ящики).

# Структура данных

Таблица 1 – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| N | Integer | Количество дискет, которое нужно купить |
| I | Integer | Вспомогательная переменная для подсчетов |
| R1 | Integer | Вспомогательная переменная для подсчетов |
| R2 | Integer | Вспомогательная переменная для подсчетов |
| R3 | Integer | Вспомогательная переменная для подсчетов |
| R4 | Integer | Вспомогательная переменная для подсчетов |
| C | Integer | Вспомогательная переменная для вывода на печать результатов |
| D | Array of integer | Массив количества дисков в каждом типе (коробки, ящики) |
| CD | Array of real | Массив цен каждого типа (коробки, ящики) |
| CDN | Array of real | Массив цен, если покупать одним типом (единичными дискетами / коробками / ящиками) необходимое к покупке количество дискет |
| CFinish | Real | Цена, если покупать необходимое к покупке количество дискет разными типами |
| L | Boolean | Переменная для проверки условия на ввод неверного типа данных |

# Детали и методика решения задачи

## Детали задачи

Детали задачи таковы:

* количество дискет в ящике больше, чем в коробке;
* покупка оптом должна быть выгоднее;
* значения могут быть только числовыми и больше нуля;
* количество дискет в каждом типе может быть только целочисленным;
* значения цен могут быть вещественными.

## Используемые формулы

При решении задачи используются следующие формулы:

* отношение цен:

Где формула (1) выражает отношение цен ящика и коробки, а формула (2) – коробки и единичной дискеты.

Проанализировав дополнительное условие, а именно «если при покупке 11 дискет выгоднее купить одну коробку, чем 11 дискет по отдельности», можно сделать вывод, что:

Чтобы такое условие выполнялось в общем виде, необходимо, чтобы:

То есть n должно быть больше отношения цен. Таким образом, если число дискет меньше этого отношения, то дешевле купить единичными дискетами;

* цена, если покупать только единичными дискетами или коробками, или ящиками:
* цена, если покупать несколькими типами:

Где формула (4) обозначает покупку коробками и единичными дискетами, (5) – ящиками и единичными дискетами, (6) – ящиками и коробками и (7) – ящиками, коробками и единичными дискетами.

## Проверка исходных данных с помощью цикла с постусловием

При проверке исходных данных использовался цикл с постусловием, так как в таком случае цикл будет запущен хотя бы один раз:

Repeat

Try

Writeln('Введите кол-во дискет в коробке и ящике: ');

Readln(D[2], D[3]);

while (D[2] < 0) or (D[3] <0) do

begin

Writeln('Введите кол-во дискет в коробке и ящике: ');

Readln(D[2], D[3]);

end;

Writeln('Введите стоимость 1 дискеты, коробки дисков и

ящика: ');

Readln(CD[1], CD[2], CD[3]);

while (CD[1] < 0) or (CD[2] < 0) or (CD[3] < 0) do

begin

Writeln('Введите стоимость 1 дискеты, коробки дисков

и ящика: ');

Readln(CD[1], CD[2], CD[3]);

end;

Writeln('Введите кол-во дискет для покупки: ');

Readln(N);

while N < 0 do

begin

Writeln('Введите кол-во дискет для покупки: ');

Readln(N);

end;

L := True;

Except

on e:EInOutError do

begin

Writeln('Введены некорректные данные!');

Writeln('');

L := False;

end;

End;

Until L;

С помощью этого цикла мы повторяем вывод фразы “Введите кол-во дискет в коробке и ящике: ” , повторяем получение данных от пользователя в массив переменных D, если было правильно введено значение, иначе фразу “Введены некорректные данные! ” до тех пор, пока полученные значения не будет выполнять всем требованиям. Так же происходит и с вводом стоимости каждого типа и необходимым к покупке количеством дискет.

## Использование Try … Except … End;

В данной задаче данные вводятся пользователем с клавиатуры. Нельзя утверждать, что пользователь введёт верные данные, а именно тип Integer.

Для этого используется «Try … Except … End;» При введении некорректных данных (это может быть как и слишком большое число, так и не тип Integer вовсе) происходит ошибка, которую считывает Except и выполняет следующие строки:

Writeln('Введены неверные данные!');

Writeln('');

E := False;

# Результаты расчетов

Вследствие выполнения программы на экран выводятся следующие результаты расчетов:

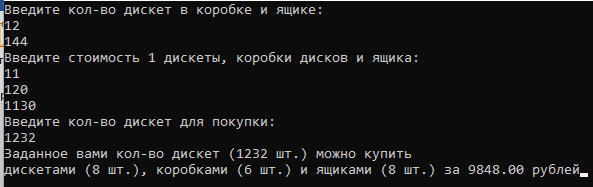


Рисунок 1– Результаты расчётов

Приложение А

(обязательное)

Исходный код программы

program Диски;

{

program counts the cheapest way to buy required amount of floppies

}

{$APPTYPE CONSOLE}

uses

System.SysUtils;

var N, I, R1, R2, R3, R4, C: integer;

D : array[2..3] of integer;

CDN : array[1..3] of real;

CD : array[1..3] of real;

CFinish: real;

L : boolean;

// N - required amount of floppies

// I, R1, R2, R3, R4 - auxiliary variables

// D - amount of floppies in each type (boxes, craters)

// CD - price of each type

// CDN - price of required amount of floppies by using

// only one type

// CFinish - price of required amount of floppies by using

// all types

// L - variable for checking errors

// C - variable for deviding on cases for easer outputting

// information

begin

// Requaring and checking variables (D[i], CD[i], N)

Repeat

// Trying to require variables (D[i], CD[i], N)

Try

Writeln('Введите кол-во дискет в коробке и ящике: ');

Readln(D[2], D[3]);

// Checking variables D[i] for positive value

while (D[2] < 0) or (D[3] <0) do

begin

Writeln('Введите кол-во дискет в коробке и ящике: ');

Readln(D[2], D[3]);

end;

Writeln('Введите стоимость 1 дискеты, коробки дисков и

ящика: ');

Readln(CD[1], CD[2], CD[3]);

// Checking variables CD[i] for positive value

while (CD[1] < 0) or (CD[2] < 0) or (CD[3] < 0) do

begin

Writeln('Введите стоимость 1 дискеты, коробки дисков

и ящика: ');

Readln(CD[1], CD[2], CD[3]);

end;

Writeln('Введите кол-во дискет для покупки: ');

Readln(N);

// Checking variable N for positive value

while N < 0 do

begin

Writeln('Введите кол-во дискет для покупки: ');

Readln(N);

end;

// Stop repeating

L := True;

// If user gave wrong types of variables

Except

on e:EInOutError do

begin

Writeln('Введены некорректные данные!');

Writeln('');

// Continue repeating

L := False;

end;

End;

Until L;

CDN[1] := (N\*CD[1]);

CDN[2] := (N div D[2] +1)\*CD[2];

CDN[3] := (N div D[3] +1)\*CD[3];

// If we can use boxes and single flappies

// for requaring amount of flappies

if (N < (CD[3]/CD[2])\*D[2]) and not (N < CD[2]/CD[1]) then

Begin

R1 := N div D[2];

CFinish := (N mod D[2])\*CD[1] + CD[2]\*R1;

C := 1;

End;

// If we can use craters with other types

// for requaring amount of flappies

if N > (CD[3]/CD[2])\*D[2] then

begin

R2 := N div D[3];

R3 := N mod D[3];

// If we can use craters and single flappies

// for requaring amount of flappies

if R3 < (CD[2]/CD[1]) then

Begin

CFinish := R3\*CD[1] + CD[3]\*R2;

C := 2;

End

else

begin

R4 := R3 div D[2];

//If we can use craters and boxes

// for requaring amount of flappies

if ((R3-R4\*D[2])>(CD[2]/CD[1])) or (R3=R4\*D[2]) then

Begin

Cfinish := CD[3]\*R2 + CD[2]\*(R4);

C := 3;

End

else

// If we can use all types

// for requaring amount of flappies

Begin

CFinish := CD[3]\*R2 + CD[2]\*R4 + (N - R2\*D[3] –

R4\*D[2])\*CD[1];

C := 4;

End;

end;

end;

// If it's cheaper to buy required amount of flappies by using

// different types

If ((CFinish<CDN[1]) And (CFinish<CDN[2]) and (CFinish<

CDN[3])) and (CFinish <> 0) then

Begin

case C of

1: begin

Writeln('Заданное вами кол-во дискет (',N,' шт.)

можно купить ');

Writeln('дискетами (', (N mod D[2]),' шт.) и

коробками (', R1,' шт.) ');

Writeln('за ', CFinish:7:2, ' рублей');

end;

2: begin

Writeln('Заданное вами кол-во дискет (',N,' шт.)

можно купить ');

Writeln('дискетами (', (N mod D[3]),' шт.) и

ящиками (', R2,' шт.)');

Writeln('за ', CFinish:7:2, ' рублей');

end;

3: begin

Writeln('Заданное вами кол-во дискет (',N,' шт.)

можно купить ');

Writeln('коробками (', R4,' шт.) и ящиками

(', R2,' шт.)');

Writeln('за ', CFinish:7:2, ' рублей');

end;

4: begin

Writeln('Заданное вами кол-во дискет (',N,' шт.)

можно купить ');

Write('дискетами (', (N - R2\*D[3] - R4\*D[2]),

' шт.), коробками (');

Write(R4,' шт.) и ящиками (', R2,' шт.) за ');

Writeln(CFinish:7:2, ' рублей');

end;

end;

Write('Выгода, покупая данным способом, а не ');

Writeln('только дискетами: ', (CDN[1]-CFinish):6:2);

Write('Выгода, покупая данным способом, а не ');

Writeln('только коробками: ', (CDN[2]-CFinish):6:2);

Write('Выгода, покупая данным способом, а не ');

Writeln('только ящиками: ', (CDN[3]-CFinish):6:2);

End;

// If it's cheaper to buy required amount of flappies by

// using concrete type

If (CFinish>CDN[1]) or (CFinish>CDN[2]) or (CFinish>

CDN[3]) or (CFinish = 0) then

Begin

// ... by using single flappies

if (CDN[1] < CDN[2]) and (CDN[1] < CDN[3]) then

begin

Writeln('Заданное вами кол-во дискет (',N,' шт.) можно

купить ');

Write('дискетами (', N,' шт.) за ', CDN[1]:7:2, '

рублей');

end;

// ... by using boxes

If (CDN[2] < CDN[1]) and (CDN[2] < CDN[3]) then

begin

Writeln('Заданное вами кол-во дискет (',N,' шт.) можно

купить ');

Write('коробками (', (N div D[2] + 1),' шт.) за ');

Write(CDN[2]:7:2, ' рублей');

end;

// ... by using craters

If (CDN[3] < CDN[1]) and (CDN[3] < CDN[2]) then

begin

Write('Заданное вами кол-во дискет (',N,' шт.) можно

купить ');

Write('ящиками (', (N div D[3] + 1),' шт.) за ');

Write(CDN[3]:7:2, ' рублей');

end;

End;

readln;

end.

Приложение Б

(обязательное)

Тестовые наборы

Тест 1

Исходные данные:

* количество дискет в коробке – 12;
* количество дискет в ящике – 144;
* стоимость одиночной дискеты – 11.5;
* стоимость коробки – 114.5;
* стоимость ящика – 1255;
* количество дискет необходимых к покупке – 9.

Ожидаемый результат:

Так как 9 < 10 (формула (2)), то самая низкая стоимость покупки необходимого количества дискет будет рассчитана по формуле (3):

CFinish = 9\*11.5 = 103.5

Полученный результат:

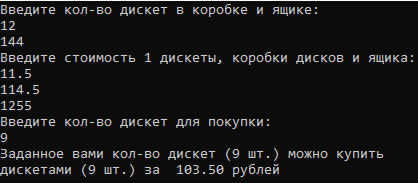


Рисунок 2 – Самая низкая стоимость при покупке одиночными дискетами

Тест 2

Исходные данные:

* количество дискет в коробке – 12;
* количество дискет в ящике – 144;
* стоимость одиночной дискеты – 14;
* стоимость коробки – 215.4;
* стоимость ящика – 1234;
* количество дискет необходимых к покупке – 430.

Ожидаемый результат:

Так как 430 = 2\*144 + 11\*12 + 10, где 10 < 15 (формула (2)), то расчет стоимости необходимого количества дискет с помощью всех типов будет по формуле (7):

СFinish = 2\*1234 + 11\*215.4 + 10\*14 = 4977.4

Однако 430 = 3\*144 - 2, где 2 < 11 (формула (2)), тогда расчет стоимости необходимого количества дискет с помощью ящиков будет по формуле (3):

CFinish = 3\*1234 = 3702

Можно увидеть, что, покупая с помощью ящиков, стоимость будет наименьшей.

Полученный результат:

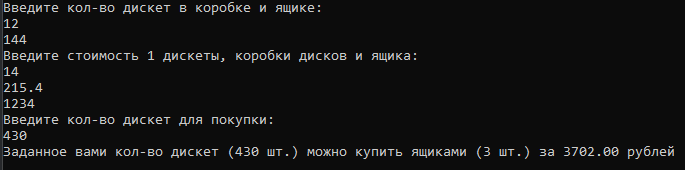


Рисунок 3 – Самая низкая стоимость при покупке ящиками

Тест 3

Исходные данные:

* количество дискет в коробке – 12;
* количество дискет в ящике – 144;
* стоимость одиночной дискеты – 18;
* стоимость коробки – 125.73;
* стоимость ящика – 1734;
* количество дискет необходимых к покупке – 80.

Ожидаемый результат:

Так как 80 = 6\*12 + 8, где 8 > 7 (формула (2)), то расчет стоимости необходимого количества дискет с помощью одиночных дискет и коробок будет невыгоден.

Однако 80 = 7\*12 - 4, где 4 < 7 (формула (2)), тогда расчет стоимости необходимого количества дискет с помощью коробок будет по формуле (3):

CFinish = 7\*125.73 = 880.11

Можно увидеть, что, покупая с помощью ящиков, стоимость будет наименьшей.

Полученный результат:

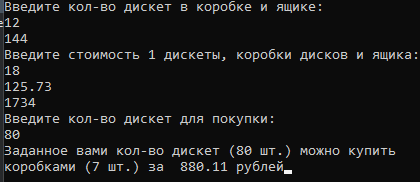


Рисунок 4 – Самая низкая стоимость при покупке коробками

Тест 4

Исходные данные:

* количество дискет в коробке – 12;
* количество дискет в ящике – 144;
* стоимость одиночной дискеты – 11.5;
* стоимость коробки – 114.5;
* стоимость ящика – 1255;
* количество дискет необходимых к покупке – 100.

Ожидаемый результат:

Так как 100 = 8\*12 + 4, где 4 < 10 (формула (2)), то расчет стоимости необходимого количества дискет с помощью одиночных дискет и коробок будет по формуле (4):

CFinish = 8\*114.5 + 4\*11.5 = 962

Полученный результат:

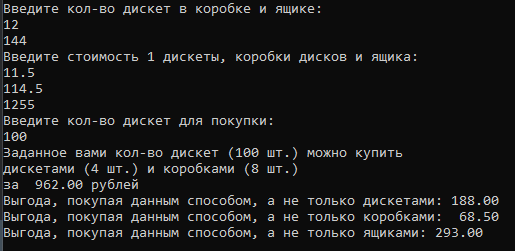


Рисунок 5 – Самая низкая стоимость при покупке одиночными дискетами и коробками

Тест 5

Исходные данные:

* количество дискет в коробке – 12;
* количество дискет в ящике – 144;
* стоимость одиночной дискеты – 11.52;
* стоимость коробки – 135.87;
* стоимость ящика – 1438;
* количество дискет необходимых к покупке – 147.

Ожидаемый результат:

Так как 147 = 1\*144 + 3, где 3 < 12 (формула (2)) то расчет стоимости необходимого количества дискет с помощью одиночных дискет и ящиков будет по формуле (5):

CFinish = 1\*1438 + 3\*11.52 = 1472.56

Полученный результат:

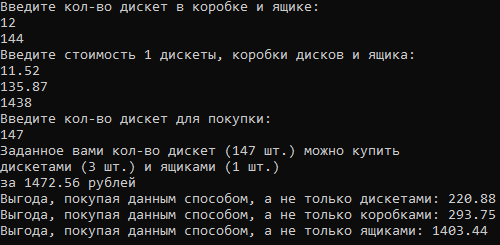


Рисунок 6 – Самая низкая стоимость при покупке одиночными дискетами и ящиками

Тест 6

Исходные данные:

* количество дискет в коробке – 12;
* количество дискет в ящике – 144;
* стоимость одиночной дискеты – 15;
* стоимость коробки – 120.53;
* стоимость ящика – 1430;
* количество дискет необходимых к покупке – 1188.

Ожидаемый результат:

Так как 1188 = 8\*144 + 3\*12, то расчет стоимости необходимого количества дискет с помощью коробок и ящиков будет по формуле (6):

CFinish = 8\*1430 + 3\*15 = 11801.59

Полученный результат:

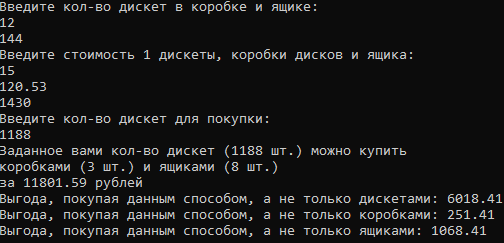


Рисунок 7 – Самая низкая стоимость при покупке коробками и ящиками

Тест 7

Таблица 5 – Тестовые ситуации

|  |  |  |
| --- | --- | --- |
| Тест | Тестовые ситуации (вместе с расчётами) | Проверка полученного результата |
| 1. |  | [См. Детали задачи](#_Детали_задачи) |
| 2. |  | [См. Детали задачи](#_Детали_задачи) |
| 3. |  | [См. Детали задачи](#_Детали_задачи) |
| 4. |  | [См. Детали задачи](#_Детали_задачи) |
| 5. |  | [См. Детали задачи](#_Детали_задачи) |

Продолжение таблицы 5

|  |  |  |
| --- | --- | --- |
| Тест | Тестовые ситуации (вместе с расчётами) | Проверка полученного результата |
| 6. |  | [См. Детали задачи](#_Детали_задачи) |