Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по разминочной работе №1

Тема работы: Решение задачи

Выполнил

студент: гр. 551003 Дементей В.С

Проверила: Фадеева Е.П.

Минск 2015

Содержание

[1 Постановка задачи 4](#_Toc432337856)

[2 Структура данных 5](#_Toc432337857)

[3 Результаты расчетов 6](#_Toc432337858)

[Приложение А 7](#_Toc432337859)

[Приложение Б 9](#_Toc432337860)

# Постановка задачи

Конкретные условия

100 монахов едят 100 пирогов. Ведущие монахи по 10 пирогов, простые по 5, ученики по 0,5. Определить: сколько монахов каждой группы.

Обобщенный вид

*allM* монахов едят *allP* пирогов. Монахи едят по чину: ведущие – *a*, простые – *b*, ученики – *c*. Определить: сколько монахов каждой группы. Разные группы монахов едят разное количество пирогов.

* 1.1 Поиск решения

Всего монахов = allM,

Всего пирогов = allP,

Ведущие монахи = V,

Простые монахи = Pr

Ученики = St

Едят ведущие = а

Едят простые = b

Едят ученики = с

Имеем систему из 2 уравнений

V+Pr+St=allM (1)

V\*a+Pr\*b+St\*c=allP (2)

Выражаем V

V=allM-Pr-st (3)

Подставим в формулу (2) и проведем группировку

St\*(a-c) +Pr\*(a-b) =allM\*a-allP

Выразим St

St= (allM\*a-allP-(a-b)\*Pr)/(a-c)

# Структура данных

Таблица - Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| allM | integer | Общее количество монахов |
| allP | integer | Общее количество пирогов |
| V | integer | Количество монахов ведущих |
| Pr | integer | Количество монахов простых |
| St | integer | Количество монахов учеников |
| a | real | Количество пирогов, которые съедает 1 ведущий монах |
| b | real | Количество пирогов, которые съедает 1 простой монах |
| c | real | Количество пирогов, которые съедает 1 монах ученик |

# Результаты расчетов

В результате поиска решения приходим к выводу, что при решении конкретных условий мы должны к прийти к одному варианту ответа. Где будет 1 ведущий монах, 9 простых монахов и 90 монахов учеников.

Приложение А

(обязательное)

Исходный код программы

**Program** TaskOne1;

{$APPTYPE CONSOLE}

**Uses**

SysUtils,

windows;

**Var**

allM,allP,V,Pr,St,variant:integer;

a,b,c:real;

{Всего моназов = allM,

Всего пирогов = allP,

Ведущие монахи = V,

Простые монахи = Pr

Ученики = St

Едят ведущие = а

Едят простые = b

Едят ученики = с

variant}

**begin**

// Подключение русского языка

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

variant:=0;

// Ввод исходных данных

Writeln('Введите количество монахов');

readln(allM);

**while** (allM=0) **or**(allM<0) **do**

**begin**

Writeln('Ошибка');

writeln('Количество монахов должно быть больше нуля');

Writeln('Повторите ввод');

Writeln('Введите количество монахов');

readln(allM);

**end**;

Writeln('Введите количество пирогов');

readln(allP);

**while** (allP=0) **or**(allP<0) **do**

**begin**

Writeln('Ошибка');

writeln('Количество пирогов должно быть больше нуля');

Writeln('Повторите ввод');

Writeln('Введите количество пирогов');

readln(allP);

**end**;

Writeln('Разный тип монахов');

writeln('должен есть разное количество пирогов');

Writeln('Сколько пирогов съедает ведущий монах?');

readln(a);

**while** a<0 **do**

**begin**

Writeln('Ошибка! Повторите ввод');

readln(a);

**end**;

Writeln('Сколько пирогов съедает простой монах?');

readln(b);

**while** (a=b) **or**(b<0) **do**

**begin**

Writeln('Ошибка!');

Writeln('Повторите ввод');

Writeln('Сколько пирогов съедает простой монах?');

readln(b);

**end**;

Writeln('Сколько пирогов съедает ученик?');

readln(c);

**while** (a=c) **or** (b=c) **or**(c<0)**do**

**begin**

Writeln('Ошибка!');

Writeln('Повторите ввод');

Writeln('Сколько пирогов съедает ученик?');

readln(c);

**end**;

// цикл по простым монахам

//от нуля до максимального значения

**for** Pr:=0 **to** allM **do**

**begin**

// вычисляем количество учеников

// используем округление для получения целых вариантов

St:=trunc((allM\*a-allP-(a-b)\*Pr)/(a-c));

// вычисляем количество ведущих

V:= allM-Pr-St;

// проверка на выполнение условия не отрицательности

// и проверка условия по общему количеству

**if** (V>=0) **and** (St>=0) **and** (V\*a+Pr\*b+St\*c=allP) **then**

**begin**

//количество решений

variant:= variant+1;

Writeln('Вариант №',variant);

writeln('Ведущие монахи = ',V);

writeln('Простые монахи = ', Pr);

writeln('Ученики монахи = ', St);

**end**;

**end**;

// если решений нет

**if** variant=0 **then**

writeln('Вариантов нет');

readln;

**end**.

Приложение Б

(обязательное)

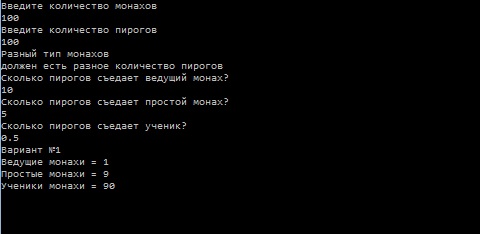
Тестовые наборы

Тест 1

Тестовая ситуация: Проверка конкретного условия

Ожидаемый результат: Один вариант решения, 1 ведущий монах, 9 простых монахов и 90 монахов учеников.

Результат:

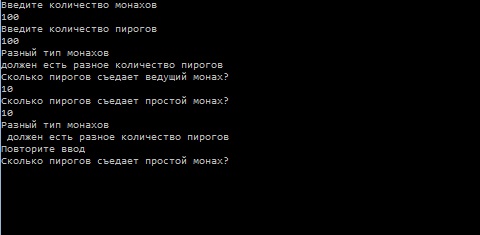


Тест 2

Тестовая ситуация: Ввод одинаковых значений a(Сколько съедает ведущий монах) и b (Сколько съедает простой монах)

Ожидаемый результат: Вывод предупреждения и повтор ввода b

Результат:

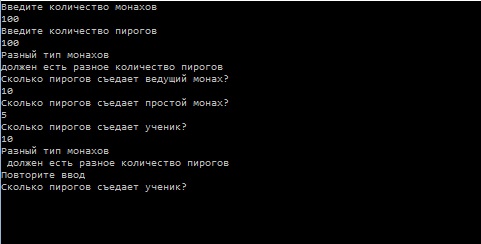


Тест 3

Тестовая ситуация: Ввод одинаковых значений a(Сколько съедает ведущий монах) и с (Сколько съедает монах-ученик)

Ожидаемый результат: Вывод предупреждения и повтор ввода с

Результат:

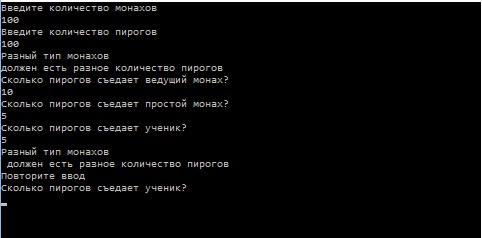


Тест 4

Тестовая ситуация: Ввод одинаковых значений b(Сколько съедает ведущий монах) и с (Сколько съедает монах-ученик)

Ожидаемый результат: Вывод предупреждения и повтор ввода с

Результат:

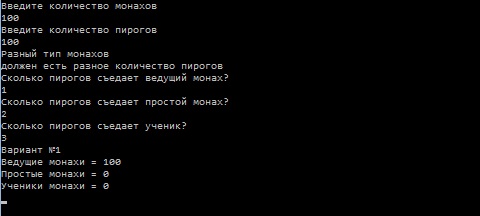


Тест 5

Тестовая ситуация: Ввод условий при которых количество любых 2 групп монахов будет равно 0

Ожидаемый результат: Вывод результатов с количеством монахов в 2 группах равных нулю

Результат:



Тест 6

Тестовая ситуация: Ввод условий при которых будет множество решений

Ожидаемый результат: Вывод результатов множеством решений

Результат:

