**Отчет по лабораторной работе № 1.**

**№ группы: ПМ-2401.**

**Выполнила: Вялых Мария Викторовна.**

**Вариант: 7.**

**Cодержание:**

* Постановка задачи;
* Входные и выходные данные;
* Выбор структуры данных;
* Алгоритм;
* Программа;
* Анализ правильности решения.

**1. Постановка задачи**

Программа получает на вход 4 числа, число Х- кг бананов, которые есть в магазине, остальные числа А, В и С- кг бананов, которые хотят купить покупатели номер 1, 2 и 3 соответственно. Все числа не превышают по модулю 109. Нужно найти количество покупателей, которые смогут купить бананы.

Данную задачу можно решить через каскадный условный оператор if, поочерёдно проверяя, хватит ли в магазине бананов для покупателя.

* Для задачи нужно рассмотреть несколько случаев:
  1. X<=0 или A<=0 или B<=0 или C<=0
     + Выводим сообщение о некорректных данных.
  2. X>0 и A>0 и B>0 и C>0
     + Рассматриваем 2 случая, для каждого из случаев рассматриваем ещё два варианта событий, и для каждого варианта рассматриваем ещё два исхода.

1. X>=A
2. X>=B
3. X>=C
4. X<C
5. X<B
6. X>=C
7. X<C
8. X<A
9. X>=B
10. X>=C
11. X<C
12. X<B
13. X>=C
14. X<C

Всего 2\*2\*2=8 случаев.

**2. Входные и выходные данные**

**Данные на вход**

На вход программа должна получить 4 числа, в условии сказано, что все числа натуральные. Нижняя граница понятна из определения натуральных чисел (числа не меньше 1), верхняя граница не дана, будем считать её равной 109.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Тип** | **min значение** | **max значение** |
| X (Число 1) | Натуральное число | 1 | 109 |
| A (Число 2) | Натуральное число | 1 | 109 |
| В (Число 3) | Натуральное число | 1 | 109 |
| С (Число 4) | Натуральное число | 1 | 109 |

**Данные на выход**

Программа должна вывести количество покупателей, которые смогли купить бананы. Так как количество покупателей равняется 3, на выходе мы получим число, не меньшее 0 и не большее 3.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Тип** | **min значение** | **max значение** |
| k (Число на выходе) | Целое неотрицательное число | 0 | 3 |

**3. Выбор структуры данных**

Программа получает 4 натуральных числа, не превышающих по модулю 109. Поэтому для их хранения можно выделить 4 переменных (X, A, B и С) типа int. Результат хранится в отдельной переменной k типа int. В переменной k хранится число от 0 до 3, включая 0 и 3.

|  |  |  |
| --- | --- | --- |
|  | **Название переменной** | **Тип (в Java)** |
| X (Число 1) | X | int |
| А (Число 2) | A | int |
| В (Число 3) | B | int |
| С (Число 4) | C | int |
| k (Число 5) | k | int |

**4. Алгоритм**

**Алгоритм выполнения программы:**

1. **Ввод данных:**  
   Программа считывает 4 натуральных числа, обозначенных как Х, А, В, С.
2. **Проверка корректности данных**

Если среди чисел окажется хотя бы одно неположительное число, программа выведет сообщение о некорректности введённых данных. Если все числа натуральные, то программа будет проверять следующее условие.

1. **Сравнение количества кг бананов в магазине и кг, которые требует 1 покупатель:**  
   Программа сравнивает значения Х и А. Если X больше или равно A, программа вычитает из Х А, прибавляет к счётчику k единицу и переходит к сравнению кг оставшихся бананов и требуемого количества кг для 2 покупателя. Если А больше, чем Х, то программа переходит к сравнению кг оставшихся бананов и требуемого количества кг для 2 покупателя, не увеличивая счётчик.
2. **Сравнение количества кг бананов в магазине и кг, которые требует 2 покупатель:**

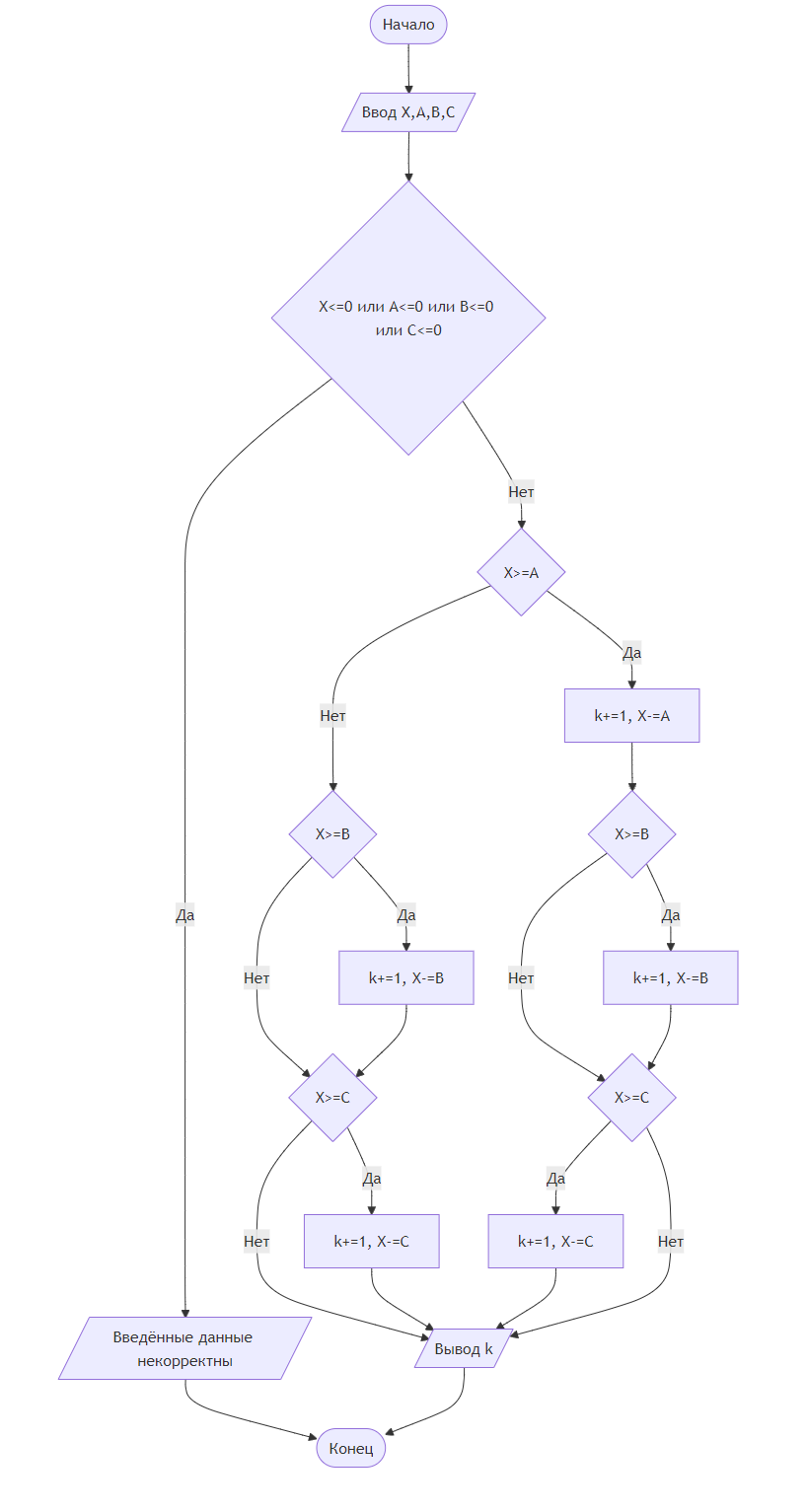
Программа сравнивает значения Х и В. Если X больше или равно В, программа вычитает из Х В, прибавляет к счётчику k единицу и переходит к сравнению кг оставшихся бананов и требуемого количества кг для 3 покупателя. Если В больше, чем Х, то программа переходит к сравнению кг оставшихся бананов и требуемого количества кг для 3 покупателя, не увеличивая счётчик.

1. **Сравнение количества кг бананов в магазине и кг, которые требует 3 покупатель:**

Программа сравнивает значения Х и С. Если X больше или равно С, программа вычитает из Х С, прибавляет к счётчику k единицу и завершает работу. Если В больше, чем Х, то программа не увеличивает счётчик.

1. **Вывод результата:**  
   На экран выводится переменная k, в которой хранится количество покупателей, которые смогли купить бананы.

**Блок-схема**

****

**5. Программа**

import java.io.PrintStream;

import java.util.Scanner;

public class Main {

public static Scanner in = new Scanner(System.in);//объект класса Scanner для ввода данных

public static PrintStream out = System.out;//объект класса PrintStream для вывода данных

public static void main(String[] args) {//массив строк,которые передаются при запуске

int X = in.nextInt(); //количество (кг) бананов в магазине

int A = in.nextInt(); //столько кг бананов хочет купить первый покупатель

int B = in.nextInt(); //столько кг бананов хочет купить второй покупатель

int C = in.nextInt(); //столько кг бананов хочет купить третий покупатель

int k = 0; //количество покупателей, которые смогли купить бананы

if (X<=0 || A<=0 || B<=0 || C<=0)//проверяем, являются ли введённые числа натуральными

out.print("Введённые данные некорректны");//выводим сообщение о неправильных данных

else {//если все числа натуральные

if (X>=A) { //если для первого покупателя хватает бананов

X=X-A; //столько кг осталось в магазине после покупки

k=k+1; //смог купить бананы,поэтому увеличиваем счётчик

if (X>=B) { //если после 1 покупателя бананов хватает на 2

X=X-B; //столько останется кг бананов после покупки

k=k+1;//смог купить бананы, поэтому увеличиваем счётчик

if (X>=C) { //если после двух покупателей хватило для третьего

X=X-C; //столько кг бананов останется в магазине

k=k+1; //смог купить бананы, поэтому увеличиваем счётчик

}

}

else

if (X>=C) { //не хватило 2,проверяем, хватит ли третьему

X=X-C; //столько кг бананов останется в магазине

k=k+1; //смог купить бананы, поэтому увеличиваем счётчик

}

}

else // если первому покупателю нужно больше баананов, чем есть в магазине

if (X>=B) { //первый ушёл, проверяем для второго

X=X-B; //на второго хватает, столько кг остаёся в магазине

k=k+1; //смог купить бананы, поэтому увеличиваем счётчик

if (X>=C) { //если после покупки остаётся на третьего

X=X-C; //столько останется в магазине

k=k+1; //смог купить бананы, поэтому увеличиваем счётчик

}

}

else //если 1 и 2 покупатели хотят больше, чем есть в магазине

if (X>=C) { //они уходят, провнряем для третьего

X=X-C; //столько кг останется в магазине

k=k+1; //смог купить, поэтому увеличиваем счётчик

}

//если никому не хватило, то счётчик остаётся равным 0

out.print(k);//выводим количество покупателей, которые смогли купить бананы

}

}

}

**6. Анализ правильности решения**

Программа работает корректно на всем множестве решений с учетом ограничений.

1. Тест на X>=A, X>=B, X>=C:
   * **Ввод:**
   * 40
   * 15
   * 14
   * 7
   * **Вывод:**
   * 3
2. Тест на X>=A, X>=B, X<=C:
   * **Ввод:**
   * 50
   * 14
   * 15
   * 25
   * **Вывод:**
   * 2
3. Тест на X>=A, X<=B, X>=C:
   * **Ввод:**
   * 30
   * 12
   * 25
   * 7
   * **Вывод:**
   * 2
4. Тест на X>=A, X<=B, X<=C:
   * **Ввод:**
   * 50
   * 40
   * 12
   * 13
   * **Вывод:**
   * 1
5. Тест на X<=A, X<=B, X<=C:
   * **Ввод:**
   * 12
   * 15
   * 13
   * 14
   * **Вывод:**
   * 0
6. Тест на X<=A, X>=B, X>=C:
   * **Ввод:**
   * 30
   * 35
   * 20
   * 5
   * **Вывод:**
   * 2
7. Тест на X<=A, X>=B, X<=C:
   * **Ввод:**
   * 20
   * 25
   * 15
   * 8
   * **Вывод:**
   * 1
8. Тест на X<=A, X<=B, X>=C:
   * **Ввод:**
   * 10
   * 15
   * 12
   * 8
   * **Вывод:**
   * 1
9. Тест на ограничение задачи (Х<=0 или A<=0 или B<=0 или C<=0):
   * **Ввод:**
   * 0
   * 15
   * 12
   * 8
   * **Вывод:**
   * Введённые данные некорректны.