***Практическая работа № 3***

***Программирование алгоритмов разветвляющейся структуры***

**ЦЕЛЬ ЦЕЛЬ РАБОТЫ:** приобретение навыков программирования вычислительных  разветвляющихся процессов с использованием условного, со ставного операторов, оператора выбора (множественного ветв ления) и оператора безусловного перехода.

**ВЫПОЛНЕНИЕ РАБОТЫ:** в соответствии с вариантом составить и реализовать  программы.

**Задание I**

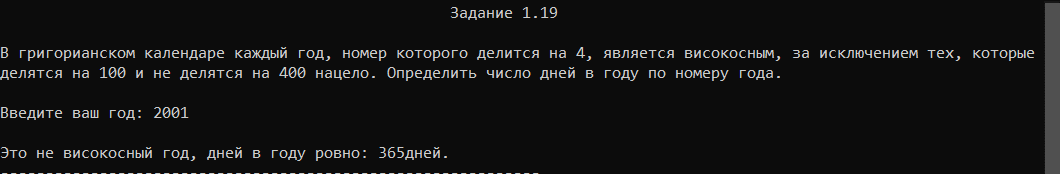
***Решить задачу с использованием конструкции ветвления:***

19. В григорианском календаре каждый год, номер которого делится на 4, является високосным, за исключением тех, которые делятся на 100 и не делятся на 400 нацело. Определить число дней в году по номеру года. Т.о. 1900 г. - не високосный, 2000 г. – високосный.

2.

|  |  |
| --- | --- |
| 2) Алгоритм: | 3) Программа: |
|  | *#include <iostream>*  *#include <math.h>*  *#include "Header.h"*  *using namespace std;*  *int YearNumber, Day, Сourse, NumberApples, DailyNormFirstCourse, DailyNormSecondCourse, DailyNormThirdCourse, Price, Price2, Result, Result2;*  *int main()*  *{*  *Task1TextOutput(); cin >> YearNumber;*    *if (YearNumber % 4 == 0)*  *{*  *Day = 366;*  *if ((YearNumber % 100 == 0) && (YearNumber % 400 != 0)) { Day = 365; }*  *}*  *else { Day = 365; }*  *if (Day == 365) { NotLeapYear(); cout << Day; DayFunction(); }*  *else { LeapYear(); cout << Day; DayFunction(); };*  *line();*      *return 0;*  *}* |

4. Вывод программы:



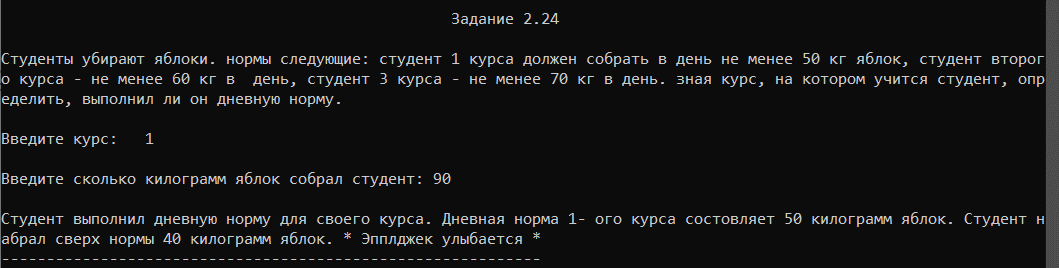
**Задание 2**

24. На первом курсе студенты четырёх групп сдают зачёт по информатике. На

втором курсе группа No 1 сдаёт экзамен по дисциплине «Информационные технологии в экономике», группа No 2 – экзамен по дисциплине «Информационные технологии в физике», группа No 3 – экзамен по дисциплине «Информационные технологии в географии», группа No 4 – экзамен по дисциплине «Информационные технологии в математических исследованиях». Задать с клавиатуры номер курса и номер группы и вывести на экран сообщение, какой зачёт или экзамен группа будет сдавать.

|  |
| --- |
| 3)Программа. |
| *#include <iostream>*  *#include <math.h>*  *#include "Header.h"*  *using namespace std;*  *int YearNumber, Day, Сourse, NumberApples, DailyNormFirstCourse, DailyNormSecondCourse, DailyNormThirdCourse, Price, Price2, Result, Result2;*  *int main()*  *{*  DailyNormFirstCourse = 50;  DailyNormSecondCourse = 60;  DailyNormThirdCourse = 70;  Task2TextOutput();  СourseInput(); cin >> Сourse; AppleInput(); cin >> NumberApples;  if (Сourse > 4) {  CourseNumber(); cout << Сourse; CourseNone();  } if (Сourse < 0) { CourseNumber(); cout << Сourse; CourseNone(); }  if (Сourse == 1) {  if (NumberApples < DailyNormFirstCourse) { StudentCouldNot(); cout << Сourse; DailyNormFalse(); cout << DailyNormFirstCourse; ApplesInput(); NumberApples = DailyNormFirstCourse - NumberApples; cout << NumberApples; ApplesInputElse(); }  else {  StudentCouldTrue(); cout << Сourse; StudentCouldTrueText(); cout << DailyNormFirstCourse;ApplesInput2(); NumberApples = NumberApples - DailyNormFirstCourse; OverNorm(); cout << NumberApples; OverNormeAJGoodHorse();  }  }  if (Сourse == 2) {  if (NumberApples < DailyNormSecondCourse) { StudentCouldNot(); cout << Сourse; DailyNormFalse(); cout << DailyNormSecondCourse; ApplesInput(); NumberApples = DailyNormSecondCourse - NumberApples; cout << NumberApples; ApplesInputElse(); }  else { StudentCouldTrue(); cout << Сourse; StudentCouldTrueText(); cout << DailyNormSecondCourse;ApplesInput2(); NumberApples = NumberApples - DailyNormSecondCourse; OverNorm(); cout << NumberApples; OverNormeAJGoodHorse(); }  }  };  if (Сourse == 3) {  if (NumberApples < DailyNormThirdCourse) { StudentCouldNot(); cout << Сourse; DailyNormFalse(); cout << DailyNormThirdCourse; ApplesInput(); NumberApples = DailyNormThirdCourse - NumberApples; cout << NumberApples; ApplesInputElse(); }  else { StudentCouldTrue(); cout << Сourse; StudentCouldTrueText(); cout << DailyNormThirdCourse;ApplesInput2(); NumberApples = NumberApples - DailyNormThirdCourse; OverNorm(); cout << NumberApples; OverNormeAJGoodHorse(); } |

4.Вывод программы:

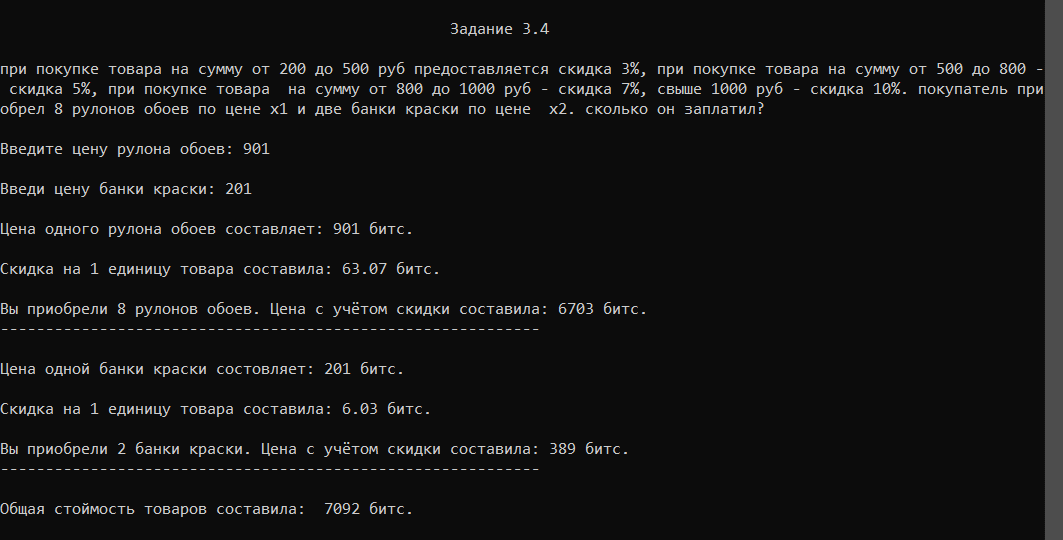


**Задание 3**

**4.** При покупке товара на сумму от 200 до 500 руб предоставляется скидка 3%, при покупке товара на сумму от 500 до 800 – скидка 5%, при покупке товара на сумму от 800 до 1000 руб – скидка 7%, свыше 1000 руб – скидка 10%. По- купатель приобрел 8 рулонов обоев по цене Х1 и две банки краски по цене Х2. Сколько он заплатил?

|  |
| --- |
| 3)Программа. |
| *#include <iostream>*  *#include <math.h>*  *#include "Header.h"*  *using namespace std;*  *int YearNumber, Day, Сourse, NumberApples, DailyNormFirstCourse, DailyNormSecondCourse, DailyNormThirdCourse, Price, Price2, Result, Result2;*  *int main()*  *{*  line();  Task3TextOutput();  PriceWallpaper(); cin >> Price;    PaintBottle(); cin >> Price2;  Result = (Price - Cost1(Price))\*8;  Result2 = (Price2 - Cost1(Price2))\*2;    CalculationAmount(); cout << Price; Bits(); CalculationAmount1(); cout << Cost1(Price); Bits(); CalculationAmount3(); cout << Result; Bits(); line(); CalculationAmount2(); cout << Price2; Bits(); CalculationAmount1(); cout << Cost1(Price2); Bits(); CalculationAmount4(); cout << Result2; Bits(); line(); CalculationAmount5(); cout << Result + Result2; Bits();  } |

4.Вывод программы.

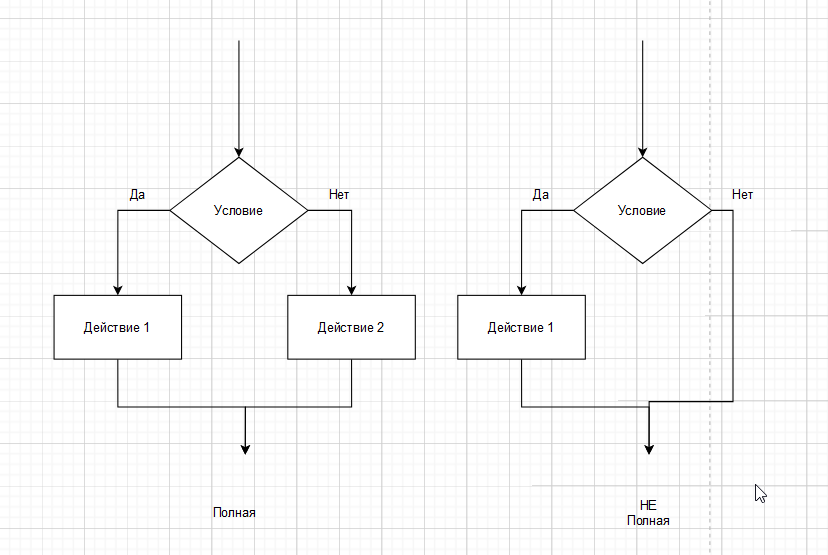


Контрольные вопросы

1. Что такое составной оператор?

Составные операторы содержат (группы) других операторов; они каким-либо образом влияют или контролируют исполнение других операторов.

1. Какова полная (неполная) форма команды ветвления (блок-схема)?



1. Каков алгоритм выполнения команды ветвления?

Алгоритм, в котором в зависимости от результатов проверки условия выполняется либо одно действие, либо другое.

1. Каков алгоритм выполнения команды множественного ветвления (выбора)?

Оператор выбора позволяет выбрать одно из нескольких возможных продолжений программы. Параметром, по которому осуществляется выбор, служит так называемый ключ выбора (или селектор) - выражение любого типа (кроме типов REAL и STRING).

1. Какие операторы сравнения используются в C++?

|  |  |  |
| --- | --- | --- |
| **Оператор** | **Символ** | **Пример** |
| Больше или равно | >= | x >= y |
| Меньше или равно | <= | x <= y |
| Равно | == | x == y |
| Не равно | != | x != y |
|  | | |

1. Что называется простым условием? Приведите примеры.

Простым условием (отношением) называется выражение, составленное из двух арифметических выражений или двух текстовых величин (иначе их еще называют операндами), связанных одним из знаков:

< - меньше, чем...

> - больше, чем...

<= - меньше, чем... или равно

>= - больше, чем... или равно

<> - не равно

= - равно

1. Что такое составное условие?

Составное условие – логическое выражение, содержащее несколько простых условий объединенных логическими операциями. Это операции not, and, or.

8. Какие логические операторы допускаются при составлении сложных усло-

вий?

Логические операции образуют сложное (составное) условие из нескольких простых (два или более) условий.

Логическая операция И && , нам уже известная;

Логическая операция ИЛИ || ;

Логическая операция НЕ ! или логическое отрицание.

1. Каков результат применения оператора логическое "И" (&&)?

Логический оператор AND ( && ) возвращает значение true , если оба операнда являются и возвращаются true false в противном случае.

1. Каков результат применения оператора логическое "ИЛИ" (||)?

Если хоть одно из двух условий является истинным, то логический оператор ИЛИ является true.

1. Каков результат применения оператора логическое отрицание (!)?

Выполняет логическое отрицание операнда, возвращая true , если операнд имеет значение false , и false , если операнд имеет значение true .

1. Каков общий вид (формат) инструкции «Ветвление»?

Ветвление в программировании — операция, применяющаяся в случаях, когда выполнение или невыполнение некоторого набора команд должно зависеть от выполнения или невыполнения некоторого условия. Ветвление — одна из трёх

(наряду с последовательным исполнением команд и циклом) базовых конструкций структурного программирования.

13. Каков алгоритм выполнения условной (тернарной) операции (?:)? Приведите

пример.

Используется он обычно в тех случаях, если условие и код, который надо выполнить, в результате проверки условия, очень простые. К примеру, спросить у пользователя хочет он продолжить работать в программе или же хочет выйти из неё

14. Каков общий вид (формат) инструкции «Выбор»?

if (оператор ветвления) и case (оператор выбора).

15. Может ли оператор ветвления содержать внутри себя другие ветвления?

Да.