Министерство образования Республики Беларусь

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра экономической информатики

**Лабораторная работа №2**

**«Операторы ветвления»**

**Вариант 15/45/15**

|  |  |
| --- | --- |
| Выполнил: | студент группы 477603  Климович Иван Вячеславович |
| Проверил: | ассистент  кафедры ЭИ  Полоско Е.И. |

Минск 2024

Оглавление

[Краткие теоретические сведения 2](#_Toc177902223)

[Индивидуальное задание 3](#_Toc177902224)

[Библиотеки и функця isDigit() 3](#_Toc177902225)

[Функция main() 5](#_Toc177902226)

[Функция mainMenu() 6](#_Toc177902227)

[Функция geometrickProgression() 8](#_Toc177902228)

[Функция luckyTick() 10](#_Toc177902229)

[Функция age() …………………………………………………………………………………………………………………………………..12](#_Toc177902230)

[Вывод 14](#_Toc177902231)

**Цель:** познакомиться с правилами построения разветвляющихся алгоритмов. Научиться использовать полную и сокращенную формы оператора if.

**Задачи:** написать код программ с использованием полной и сокращенной формы оператора if. Обработать все возможные ошибки при неверно введённых в консоль исходных данных.

# **Краткие теоретические сведения**

В данной работе я создал 2 функции (mainMenu() и isDigit()), которые служат для выбора одной из подпрограмм и для проверки введённой строки (есть ли в строке какие-либо символы кроме цифр). Эти функции я скорее всего буду использовать в дальнейших лабораторных, поэтому я полностью опишу их в данном отчёте, чтобы в будущих отчётах долго на них не останавливаться.

# **Индивидуальное задание**

## **Библиотеки и функция isDigit()**

#include <stdio.h>

#include <math.h>

#include <locale.h>

#include <conio.h>

#include <stdlib.h>

#include <ctype.h>

int isDigit()

{

int n = 0;

char input[100];

fgets(input, sizeof(input), stdin);

for (int i = 0; input[i] != '\0'; i++) if (input[i] < '0' || input[i] > '9') n++;

if (n > 1)

{

printf("Явно вы ввели что-то не то\n");

printf("Press any key to continue...\n");

\_getch();

system("cls");

mainMenu();

}

else

{

sscanf\_s(input, "%d", &n);

return n;

}

}

В данной программе я использую довольно широкий список библиотек, который можно было бы сократить, но для корректной, быстрой и чёткой работы эти библиотеки необходимы.

Сама же функция isDigit() служит для проверки введённой строки. Если в строке присутствует какой-либо символ, который не является цифрой, программа будет выдавать ошибку.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Изображение выглядит как диаграмма, текст, План, линия

Автоматически созданное описание

Схема алгоритма isDigit()

Из интересного:

1. Для ввода строки вместо традиционного scanf\_s() я использовал функцию fgets(input, sizeof(input), stdin); которая помещает все введённые символы в консоли в массив input[], с которым в последствии гораздо удобнее работать.
2. Функция sscanf\_s(input, "%d", &n); выполняет практически обратную роль. Она строку(после проверки число) из массива input[] записывает в переменную n.
3. В данной функции, как и во многих последующих используется блок:  
   printf("Press any key to continue...\n");

\_getch();

system("cls");

mainMenu();  
 Данный блок используется при ошибке или окончании подпрограммы и его роль в том, чтобы очистить консоль и вызвать функцию mainMenu(); (Которую мы скоро рассмотрим) после того, как пользователь нажмёт абсолютно любую клавишу на клавиатуре.

## **Функция main()**

int main()

{

char\* locale = setlocale(LC\_ALL, "");

mainMenu();

}

Функция main() реализована всего в двух строчках. Первая строка добавляет все локализации для компилятора, а вторая вызывает рекурсивную функцию mainMenu(), которая и будет выполнять основную работу программы.

Изображение выглядит как текст, Шрифт, диаграмма, линия

Автоматически созданное описание

Схема алгоритма main()

## **Функция mainMenu()**

mainMenu()

{

int n;

printf("Что вы ходите сделать?\n");

printf("1 -- Найти сумму геометрической прогрессии\n");

printf("2 -- Узнать, счастливый ли у вас билет\n");

printf("3 -- Узнать свой возраст\n");

printf("4 -- Выход\n");

n = \_getch();

if (n > 57 || n < 48)

{

printf("Некоректный ввод\n");

printf("Press any key to continue...\n");

\_getch();

system("cls");

mainMenu();

}

else

{

n -= 48;

switch (n)

{

case 1:

{

system("cls");

geometrickProgression();

mainMenu();

break;

}

case 2:

{

system("cls");

luckyTick();

mainMenu();

break;

}

case 3:

{

system("cls");

age();

mainMenu();

break;

}

case 4:

{

break;

}

default:

{

printf("Некоректный ввод\n");

printf("Press any key to continue...\n");

\_getch();

system("cls");

mainMenu();

}

}

}

}

Пожалуй, это самая интересная функция моей программы, поскольку с ней пользователь будет взаимодействовать чаще всего. Во избежание работы с циклами и возможными проблемами в будущем, я сделал данную фукцию рекурсивной (вызывающей саму себя), благодаря чему программа работает гораздо стабильнее.

Выбор подпрограммы в данной функции реализован через функцию \_getch() из библиотеки <conio.h>, благодаря чему вместо проверки строки мне достатоно проверить лишь один ввёденный символ. После проверки введённого символа осуществляется выбор подпрограммы либо завершение самой программы.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Изображение выглядит как диаграмма, План, зарисовка, Технический чертеж

Автоматически созданное описание

**Подпрограмма 1 (вариант 15)**

## **Функция geometrickProgression()**

*Задача:*

*Найти сумму членов геометрической прогрессии b, b q, ..., bH по введенным значениям b, q, n, q ≠ 1.*

int geometrickProgression()

{

int b, q, n;

double sum;

printf("Введите первый член геометрической прогрессии\n");

b = isDigit();

printf("Введите знаменатель прогрессии\n");

q = isDigit();

if (q == 1) printf("Знаменатель не может быть равен 1!\n");

else

{

printf("Введите количесво членов прогрессии\n");

sum = (double)(b \* (1 - pow((double)q, n)) / (1. - q));

printf("Сумма вашей геометрической прогрессии равна %.3f\n", sum);

}

printf("Press any key to continue...\n");

\_getch();

system("cls");

return 0;

}

Данная функция просто считает сумму геометрической прогрессии имея в исходных данных первый член прогрессии, её знаменатель и кол-во членов, попутно используя ранее описанные функции, поэтому не вижу смысла на ней долго задерживаться.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Изображение выглядит как диаграмма, текст, План, Технический чертеж

Автоматически созданное описание

Схема алгоритма geometricProgression()

**Подпрограмма 2 (вариант 45)**

## **Функция luckyTick()**

int luckyTick()

{

int num, a, b, c, d;

printf("Введите номер билета(четырёхзначный): \n");

num = isDigit();

if (num < 1000 || num > 9999) printf("У билета номер четырёхзначный!\n");

else

{

a = num / 1000;

d = num % 1000;

b = d / 100;

d = d % 100;

c = d / 10;

d = d % 10;

if (a + b == c + d) printf("У вас СЧАСТЛИВЫЙ БИЛЕТ!\n");

else printf("Повезёт в следующий раз\n");

}

printf("Press any key to continue...\n");

\_getch();

system("cls");

return 0;

}

Данная функция также ничем не примечательна, она просто разбивает четырёхзначное число на цифры и сравнивает сумму первых и последних двух цифр, также используя ранее описанные функции.

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание**

Изображение выглядит как диаграмма, текст, План, Технический чертеж

Автоматически созданное описаниеСхема алгоритма luckyTick()

**Подпрограмма 3 (Задание 2, вариант 15)**

## **Функция age()**

*Задача:*

*Для целого числа К (от 1 до 9) напечатать фразу мне К лет, учиты-вая при этом, что при некоторых значениях К слово лет надо заменить на сло-во год или года.*

int age()

{

int K;

printf("Введите свой возраст(от 0 да 9): ");

K = isDigit();

switch (K)

{

case 0:

{

printf("Мне %d лет:\n", K);

break;

}

case 1:

{

printf("Мне %d год\n", K);

break;

}

case 2:

{

printf("Мне %d года\n", K);

break;

}

case 3:

{

printf("Мне %d года\n", K);

break;

}

case 4:

{

printf("Мне %d года\n", K);

break;

}

case 5:

{

printf("Мне %d лет\n", K);

break;

}

case 6:

{

printf("Мне %d лет\n", K);

break;

}

case 7:

{

printf("Мне %d лет\n", K);

break;

}

case 8:

{

printf("Мне %d лет\n", K);

break;

}

case 9:

{

printf("Мне %d лет\n", K);

break;

}

default:

{

printf("Ну просили же от 0 до 9\n");

break;

}

}

printf("Press any key to continue...\n");

\_getch();

system("cls");

return 0;

}

Данная функция также особо ничем не выделяется. В ней пользователь вводит цифру от 0 до 9, после чего с помощью конструкции switch-case выводится необходимая строчка.

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание**

Изображение выглядит как текст, диаграмма, План, Технический чертеж

Автоматически созданное описаниеСхема алгоритма age()

**Вывод:** В данной лабораторной работе я познакомился с правилами построения разветвляющихся алгоритмов. Научиться использовать полную и сокращенную формы оператора if. Также узнал много нового, когда обрабатывал возможные ошибки программы.