Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних технологій, автоматики та метрології

Кафедра «Комп’ютеризовані системи автоматики»



ЗВІТ

про виконання лабораторної роботи №2

«ОПЕРАТОРИ ВИБОРУ»

із навчальної дисципліни: «Алгоритми і структури даних»

Варіант №1

Виконав:

студент групи ІР-24

Кіндрат Віктор Романович

Прийняв:

к. ф.-м. н., доцент

Сиротюк С.В

Львів 2024

**Мета:** ознайомитися з умовними операторами if…else, операторами-перемикачами, навчитися будувати найпростіші програми, які розгалужуються.

**Програма 2\_1. Програма визначає, чи є введене число парним.**

#include<stdio.h>

#include <stdlib.h>

#include <iostream>

int main()

{int n;

puts("Input number:");

scanf("%d",&n);

if(n%2==0) printf("number %d even \n", n);

else printf("number %d odd\n\n", n);

system("pause");

return 0;}

**Завдання до програми 2\_1.** Розробіть програму, яка виконує задані обчислення та виводить на екран відповідні повідомлення.

1. Перевірте, чи більше 100 введене з клавіатури ціле число.

**Програмний код до завдання:**

#include <stdio.h>

void check\_if\_more\_than\_100(int n) {

if (n > 100) {

printf("Number %d is greater than 100.\n", n);

} else {

printf("Number %d is not greater than 100.\n", n);

}

}

int main() {

int n;

puts("Enter an integer: ");

scanf("%d", &n);

check\_if\_more\_than\_100(n);

return 0;

}

**Приклад роботи програмного коду:**

viktorkindrat@MacBook-Air-Viktor lab2 % ./1.out

Enter an integer:

-> 10

<- Number 10 is not greater than 100.

viktorkindrat@MacBook-Air-Viktor lab2 % ./1.out

Enter an integer:

-> 110

<- Number 110 is greater than 100.

**Програма 2\_2.** Програма визначає, чи є введене число додатним та більшим чи меншим заданого числа.

**Завдання до програми 2\_2.** Розробіть програму, яка виконує задані обчислення, використовуючи вкладений умовний оператор, та виводить на екран чотири відповідні повідомлення.

1. Введіть з клавіатури ціле число, якщо воно парне, перевірте чи є

додатним, якщо – непарне, чи більше 25.

**Програмний код до завдання:**

#include <stdio.h>

void check\_number(int n) {

if (n % 2 == 0) {

if (n > 0) {

puts("Is even and more than 0");

} else {

puts("Is even and less than 0");

}

} else {

if (n > 25) {

puts("Is odd and more than 25");

} else {

puts("Is odd and less than 25");

}

}

}

int main() {

int x;

printf("Input value: ");

scanf("%d", &x);

check\_number(x);

return 0;

}

**Приклад роботи програмного коду:**

viktorkindrat@MacBook-Air-Viktor lab2 % ./2.out

-> Input value: 10

<- Is even and more than 0

viktorkindrat@MacBook-Air-Viktor lab2 % ./2.out

-> Input value: 9

<- Is odd and less than 25

viktorkindrat@MacBook-Air-Viktor lab2 % ./2.out

-> Input value: -2

<- Is even and less than 0

viktorkindrat@MacBook-Air-Viktor lab2 % ./2.out

-> Input value: 27

<- Is odd and more than 25

**Програма 2\_3.** Програма в залежності від введеного знаку визначає результат виконання дії за допомогою оператора switch() із застосуванням оператора break.

#include <stdio.h>

#include <stdlib.h>

int main() {

float a, b;

char s;

puts("Input value a, b:");

scanf("%f %f", &a, &b);

fflush(stdin);

puts("Input sing (-, +, \*, /):");

s = getchar();

switch(s) {

case '-': printf("%.2f - %.2f = %.2f\n", a, b, a - b); break;

// ... remaining code

}

system("pause");

return 0;

}

**Завдання до програми 2\_3.** Розробіть програму з використанням оператора switch(), яка згідно з введеною з клавіатури першої букви виводить на екран задану повну назви або повідомлення про помилку, якщо введена буква не співпадає з заданими.

1. Назви 5 країн Азії.

**Програмний код до завдання:**

#include <stdio.h>

int main() {

char s;

while (1) {

puts("Please input a character:");

scanf("%c", &s);

switch (s) {

case 'i':

puts("India");

break;

case 's':

puts("South Corea");

break;

case 'c':

puts("China");

break;

case 'a':

puts("Albania, Afganistan");

break;

default:

printf("Error! No counry was found\n");

}

fflush(stdin);

}

return 0;

}

**Приклад роботи програмного коду:**

viktorkindrat@MacBook-Air-Viktor lab2 % ./3.out

<- Please input a character:

-> i

<- India

<- Please input a character:

-> s

<- South Corea

<- Please input a character:

-> c

<- China

<- Please input a character:

-> a

<- Albania, Afganistan

<- Please input a character:

-> s

<- South Corea

<- Please input a character:

-> g

<- Error! No country was found

**Програма 2\_4.** Програма в залежності від введеної цифри виводить назву дня тижня.

#include <stdio.h>

#include <stdlib.h>

int main() {

int a;

puts("Input number from 1 to 7, you know day of week:");

scanf("%d", &a);

switch(a) {

case 1: puts("Monlay"); break;

// ... remaining code

}

system("pause");

return 0;

}

**Завдання до програми 2\_4.** Розробіть програму, в якій вводиться з клавіатури ціле число та на екран виводиться відповідне повідомлення або повідомлення про помилку, якщо введене число не співпадає з заданими (кожне з повідомлень може бути позначене кількома мітками).

1. Введіть номер місяця та виведіть одне з відповідних повідомлень: "Осінній семестр".

**Програмний код до завдання:**

#include <stdio.h>

int main() {

int a;

puts("Input number from 1 to 12 to know which period is now:");

scanf("%d", &a);

switch (a) {

case 2:case 3:case 4:case 5:case 6:

puts("Spring semester");

break;

case 7:case 8:

puts("Holidays");

break;

case 1:case 9:case 10:case 11:case 12:

puts("Autumn semester");

break;

default:

puts("Error! Please input a value between 1 and 12.");

}

return 0;

}

**Приклад роботи програмного коду:**

viktorkindrat@MacBook-Air-Viktor lab2 % ./4.out

<- Input number from 1 to 12 to know which period is now:

-> 3

<- Spring semester

viktorkindrat@MacBook-Air-Viktor lab2 % ./4.out

<- Input number from 1 to 12 to know which period is now:

-> 12

<- Autumn semester

viktorkindrat@MacBook-Air-Viktor lab2 % ./4.out

<- Input number from 1 to 12 to know which period is now:

-> 7

<- Holidays

**Програма 2\_5.** Програма визначає максимальне число серед трьох введених з клавіатури.

#include <stdio.h>

#include <stdlib.h>

int main() {

int a, b, c, max;

puts("Input value a, b, c:");

scanf("%d %d %d", &a, &b, &c);

if(a > b && a > c) max = a;

// ... remaining code

system("pause");

return 0;

}

**Завдання до програми 2\_5.** Розробіть програму, яка виконує задані обчислення та виводить на екран відповідні повідомлення.

Введіть з клавіатури чотири дійсних числа, визначте найменше з них, перевірте додатне воно чи від’ємне.

**Програмний код до завдання:**

#include <stdio.h>

int main() {

int a, b, c, d;

puts("Input value a, b, c, d:");

scanf("%d %d %d %d", &a, &b, &c, &d);

int min = a;

if (b < min) {

min = b;

}

if (c < min) {

min = c;

}

if (d < min) {

min = d;

}

if (min > 0) {

puts("Minimum value is positive.");

} else {

puts("Minimum value is negative.");

}

printf("Minimum value = %d\n", min);

return 0;

}

**Приклад роботи програмного коду:**

viktorkindrat@MacBook-Air-Viktor lab2 % ./5.out

<- Input value a, b, c, d:

-> 1 2 3 4

<- Minimum value is positive.

<- Minimum value = 1

viktorkindrat@MacBook-Air-Viktor lab2 % ./5.out

<- Input value a, b, c, d:

-> -2 -5 6 1

<- Minimum value is negative.

<- Minimum value = -5

**Висновок:**

У процесі виконання лабораторної роботи я ознайомився із основами роботи з умовними операторами в мові програмування C. Зокрема, вивчив на практиці, як використовувати оператори if...else для перевірки чисел на парність, додатнісь/від’ємність та інші умови, що дозволило реалізувати програми, які обробляють різні вхідні дані та виводять відповідні результати.

Оператор switch виявився корисним для спрощення коду і підвищення його читабельності, оскільки він дозволяє виконувати різні дії в залежності від введеного значення.

Виконання різноманітних завдань, таких як визначення мінімального чи максимального значення, перевірка на парність, а також виконання арифметичних операцій, сприяло закріпленню отриманих знань на практиці. У результаті лабораторна робота допомогла зрозуміти, як умовні оператори можуть бути використані для прийняття рішень на основі вхідних даних, закладаючи основу для подальшого вивчення більш складних концепцій програмування.