Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

Звіт

з лабораторної роботи № 5

з дисципліни “Базові методології та технології програмування”

на тему

“Програмна реалізація алгоритмічних конструкцій розгалуження ”

Виконав студент

академічної групи КІ-20

Микитенко Д. Ю.

Перевірив доцент

Доренський О. П.

Кропивницький-2020

Мета роботи: полягає у набутті ґрунтовних вмінь і практичних навичок синтезу розгалужених алгоритмів розв’язування обчислювальних задач та їх формального представлення мовою програмування С (ISO/IEC 9899:2018) задля реалізації прикладного програмного забезпечення у кросплатформовому Code::Blocks IDE.

Завдання до лабораторної роботи:

1. Створити застосунок з вихідного коду програми лістинга 5.1.

2. Програмно реалізувати алгоритм розв’язування задачі 2.1.

3. Реалізувати програмне забезпечення розв’язування задачі 5.1.

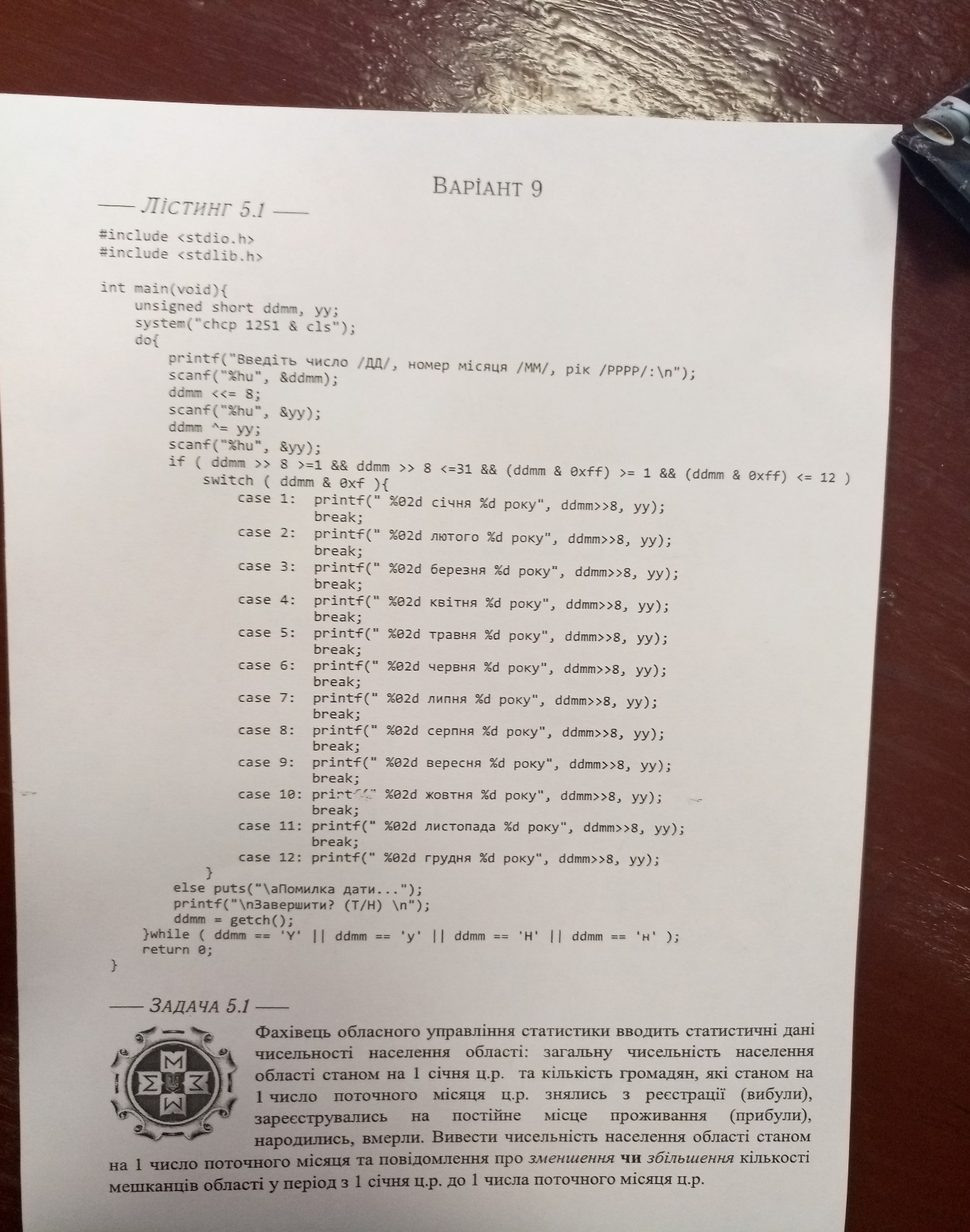


Рисунок 1 – Завдання роботи варіант № 9

1. Згідно з п.3-5 Порядку були створені деки: prj, Software, TestSuite.

Лістинг 5.1 був записаний у IDE Code::Blocks у папці prj.

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

unsigned short ddmm, yy;

system("chcp 1251 & cls");

do{

printf("Введіть число /ДД/, номер місяця /ММ/, рік /РРРР/:\n");

scanf("%hu", &ddmm);

ddmm <<= 8;

scanf("%hu", &yy);

ddmm ^= yy;

scanf("%hu", &yy);

if ( ddmm >> 8 >= 1 && ddmm >> 8 <= 31 && (ddmm & 0xff) >= 1 && (ddmm & 0xff) <= 12)

switch(ddmm & 0xf){

case 1:

printf(" %02d січня %d року", ddmm >> 8, yy);

break;

case 2:

printf(" %02d лютого %d року", ddmm >> 8, yy);

break;

case 3:

printf(" %02d березня %d року", ddmm >> 8, yy);

break;

case 4:

printf(" %02d квітня %d року", ddmm >> 8, yy);

break;

case 5:

printf(" %02d травня %d року", ddmm >> 8, yy);

break;

case 6:

printf(" %02d червня %d року", ddmm >> 8, yy);

break;

case 7:

printf(" %02d липня %d року", ddmm >> 8, yy);

break;

case 8:

printf(" %02d серпня %d року", ddmm >> 8, yy);

break;

case 9:

printf(" %02d вересня %d року", ddmm >> 8, yy);

break;

case 10:

printf(" %02d жовтня %d року", ddmm >> 8, yy);

break;

case 11:

printf(" %02d листопада %d року", ddmm >> 8, yy);

break;

case 12:

printf(" %02d грудня %d року", ddmm >> 8, yy);

break;

}

else puts("\aПомилка дати...");

printf("\nЗавершити?(Т/Н) \n");

ddmm = getch();

}while (ddmm == 'Y' || ddmm == 'y'|| ddmm == 'Н'|| ddmm == 'н');

return 0;

}

1. Алгоритм 2.1, розроблений під час виконання лабораторної роботи № 2 і представлений схемним способом, був записаний мовою програмування С

За допомогою аналізу був сформований перелік змінних(вхідні та вихідні значення).

Вихідні значення:

1. m – ціле число, відповідає за результат;

Вхідні значення:

1. а – ціле число;
2. b – ціле число;
3. с – ціле число;

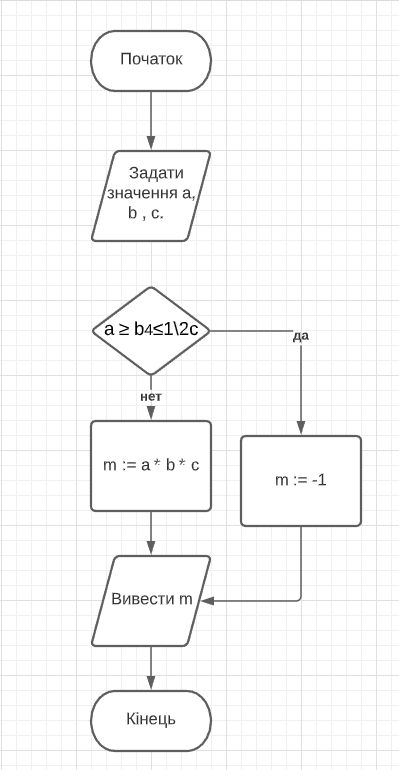


Рисунок 2 – Блок-схема алгоритму

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int main()

{

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

int a, b, c, m;

printf("Задайте значення а: ");

scanf("%d", &a);

printf("Задайте значення b: ");

scanf("%d", &b);

printf("Задайте значення c: ");

scanf("%d", &c);

if(a >= pow(b,4))

if(pow(b, 4) <= c/2)

{

m = a \* b \* c;

printf("%d", m);

return 0;

}

m = -1;

printf("%d", m);

return 0;

}

1. За допомогою аналізу був сформований перелік змінних(вхідні та

вихідні значення) для задачі 5.1.

Вихідні дані:

1. AmountNow – ціле число, відповідає за кількість населення на зараз.
2. Diff – ціле число, віповідає за різницю кількості населення.

Вхідні дані:

1. AmountDecember – ціле число, відповідає за загальну чисельність області станом на 1 січня ц.р.
2. DeregisteredNow – ціле число, відповідає за кількість громадян, які знялись з реєстрації.
3. RegisteredNow – ціле число, відповідає за кількість населення, які зареєструвались.
4. BornNow – ціле число, відповідає за кількість народжених.
5. DiedNow – ціле число, відповідає за кількість померлих.

Математична модель:

1. AmountNow = AmountDecember - DeregisteredNow + RegisteredNow + BornNow – DiedNow;
2. Diff = AmountNow – AmountDecember;

Алгоритм був сформульований у вербальній формі:

1. Задати значення AmountDecember;
2. Задати значення DeregisteredNow;
3. Задати значення RegisteredNow;
4. Задати значення BornNow;
5. Задати значення DiedNow;
6. AmountNow = AmountDecember – DeregisteredNow + RegisteredNow + BornNow – DiedNow;
7. Якщо AmountNow > AmountDecember

Diff = AmountNow – AmountDecember;

1. Перейти на крок 10;
2. Diff = AmountDecember – AmountNow;
3. Вивести AmountNow;
4. Вивести Diff.

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int main()

{

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

int AmountDecember, DeregisteredNow, RegisteredNow, BornNow, DiedNow, AmountNow, Diff;

char display[11];

printf("Оберіть мову інтерфейсу:\nChoose the language of the display:\n(у/e)\n");

gets(display);

switch(display[0])

{

case('y'):

printf("Введіть загальну чисельність населення області станом на 1 січня ц.р.: ");

scanf("%d", &AmountDecember);

printf("Введіть кількість громадян, які станом на 1 січня поточного місяця ц.р. вибули: ");

scanf("%d", &DeregisteredNow);

printf("Введіть кількість громадян, які станом на 1 число поточного місяця ц.р. прибули: ");

scanf("%d", &RegisteredNow);

printf("Введіть кількість громадян, які станом на 1 число поточного місяця ц.р. народилися: ");

scanf("%d", &BornNow);

printf("Введіть кількість громадян, які станом на 1 число поточного місяця ц.р. померли: ");

scanf("%d", &DiedNow);

AmountNow = AmountDecember - DeregisteredNow + RegisteredNow + BornNow - DiedNow;

if(AmountNow < 0 || AmountDecember < 0)

{

printf("Помилка: були неправильно введені дані...");

return 0;

}

if(AmountNow > AmountDecember)

{

Diff = AmountNow - AmountDecember;

printf("\nКількість мешканців на зараз складає: %d", AmountNow);

printf("\nКількість мешканців області у період з 1 січня ц.р. до 1 числа поточного місяця ц.р. збільшилось на %d", Diff);

}

if (AmountNow < AmountDecember)

{

Diff = AmountDecember - AmountNow;

printf("\nКількість мешканців на зараз складає: %d", AmountNow);

printf("\nКількість мешканців області у період з 1 січня ц.р. до 1 числа поточного місяця ц.р. зменшилась на %d", Diff);

}

if (AmountNow == AmountDecember)

printf("\nКількість мешканців не змінилась.");

break;

case('e'):

printf("Enter the total population of the region as of January 1 this year.: ");

scanf("%d", &AmountDecember);

printf("Enter the number of citizens who, as of January 1 of the current month of this year, was deregestered: ");

scanf("%d", &DeregisteredNow);

printf("Enter the number of citizens who, as of January 1 of the current month of this year, was regestered: ");

scanf("%d", &RegisteredNow);

printf("Enter the number of citizens who, as of January 1 of the current month of this year, was born: ");

scanf("%d", &BornNow);

printf("Enter the number of citizens who, as of January 1 of the current month of this year, died: ");

scanf("%d", &DiedNow);

AmountNow = AmountDecember - DeregisteredNow + RegisteredNow + BornNow - DiedNow;

if(AmountNow < 0 || AmountDecember < 0)

{

printf("Error: wrong input...");

return 0;

}

if(AmountNow > AmountDecember)

{

Diff = AmountNow - AmountDecember;

printf("\nThe number of inhabitants is currently: %d", AmountNow);

printf("\nThe number of residents of the region in the period from January 1 this year until the 1st of the current month this year increased by %d", Diff);

}

if (AmountNow < AmountDecember)

{

Diff = AmountDecember - AmountNow;

printf("\nThe number of inhabitants is currently: %d", AmountNow);

printf("\nThe number of residents of the region in the period from January 1 this year until the 1st of the current month this year decreased by %d", Diff);

}

if (AmountNow == AmountDecember)

printf("\nThe number of inhabitants has not changed.");

break;

default:

break;

}

return 0;

}

Висновок: у ході виконання лабораторної роботи № 5 треба було записати лістинг й скомпілювати його, записати мовою програмування С алгоритм2.1, поданий блок-схемою, у лабораторній роботі № 2, а також реалізувати програмне забезпечення задачі 5.1.

Щоб виконати перше завдання достатньо було отримати листок з варіантом роботи у викладача, проаналізувати лістинг, після чого створити проєкт консольного застосування у IDE Code::Blocks й безпосередньо записати сам лістинг 5.1. Під час виконання завдання труднощів не було виявлено й проєкт компілювався. Після цього .exe файл першого завдання був перенесений до теки Software, що була створена до цього на подобі до лабораторної роботи №4.

Для виконання другого завдання потрібно було віднайти блок-схему алгоритму 2.1, що був розроблений під час виконання лабораторної роботи №2. Алгоритм було знову проаналізовано й був сформований перелік вхідних та вихідних значень для нього. Після цього алгоритм було записано мовою програмування С у IDE Code::Blocks. Також вихідний .exe файл було переміщено до теки Software.

Задля виконання третього завдання потрібно було уважно проаналізувати дану задачу, задокументувати вхідні та вихідні значення, їх типи та побудувати математичну модель. Після цього був створений шаблон програмного забезпечення до задачі 5.1, адже ПЗ не було протестовано. Потім було створено 12 тест-сьютів для умови задачі й було перевірено кожен з них для нашого програмного забезпечення, після чого виявилося, що всі тест-сьюти пройшли тестування. Також для задачі 5.1 був розроблений інтерфейс не тільки на українській мові, а й на англійській з вибором їх на початку.

Підбиваючи підсумки цієї лабораторної роботи, були закріплені практичні вміння праці з тестуванням ПЗ, закріпленні знання з теми структурного вигляду алгоритму (блок-схеми). Також були здобуті практичні вміння з розробкою ПЗ для окремої задачі та створення інтерфейсу до неї.