**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут **КНІТ**

Кафедра **ПЗ**

# ЗВІТ

До лабораторної роботи № 1

**На тему:** *“Основні поняття мови C. Реалізація алгоритмів з розгалуженням.”*

**З дисципліни:** *“Основи програмування”*

**Лектор:**

ст. викладач кафедри ПЗ

Муха Т.О.

**Виконав:**

студ. групи ПЗ-13

Топала Т.Д.

**Прийняв:**

доц. кафедри ПЗ

Дяконюк Л.М.

«\_\_\_» \_\_\_\_\_\_ 2021 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2021

**Тема роботи:** Основні поняття мови C. Реалізація алгоритмів з розгалуженням.

**Мета роботи:** навчитися програмувати на мові С найпростіші лінійні алгоритми та

алгоритми з галуженням.

## Індивідуальне завдання

## 1. Скласти програми на мові С у відповідності із завданнями.

## 2. Виконати обчислення по програмі.

## 3. Підготувати та здати звіт про виконання лабораторної роботи.

## Завдання 1. Написати програму для обчислення заданих арифметичних виразів.

## Вважати, що X, Y –змінні, значення яких слід вводити з клавіатури,

## всі інші величини виразу описати як константи.

## b = x(arctg z+e-(x+3)), де: x = -0.622, z = 5.541

## Завдання 2. Не використовуючи оператор циклу скласти програму,

## яка за заданим натуральним восьмизначним числом

## знаходить подвоєний добуток ненульових цифр цього числа та кількість нулів.

## Теоретичні відомості

Кожна мова програмування має свій *алфавіт*, тобто набір допустимих символів з яких може складатися програма. Алфавіт мови С містить:

1) великі і малі букви латинського алфавіту (А...Z, a...z);

2) арабські цифри від 0 до 9;

3) прості спеціальні символи: ‘+’, ‘-‘, ‘\*’, ‘/’, ‘:’,’;’, ‘,’,’=’, ‘>’, ‘<’, ‘[‘,’]’, ‘(‘,’)’,‘{‘,’}’,’ ‘;

4) складені спеціальні символи: ‘!=’, ‘>=’, ‘<=’ , ‘->’.

Із цих символів складаються базові елементи мови. До них належать:

1) ідентифікатори;

2) числа;

3) рядки.

Усі імена, які використовуються в програмі для означення різних об’єктів і конструкцій називаються *ідентифікаторами*. Усі ідентифікатори мови С можна поділити на три групи:

**Службові (зарезервовані) слова** мови: призначення цих слів визначено самою мовою, їх не можна використовувати інакше, як це передбачено правилами (синтаксисом) мови. Основні службові слова подано нижче:

asm

auto

break

case

char

const

continue

default

do

double

else

enum

extern

float

for

goto

if

int

long

register

return

short

signed

sizeof

static

struct

switch

typedef

union

unsigned

void

volatile

while

*Числа* в мові C можуть записуватися в десятковій системі числення, а також у вісімковій (починаються з префікса 0) та шістнадцятковій (починаються з префікса 0х) системах. Числа у десятковій формі можуть бути цілими і дійсними. *Цілі числа* записуються у звиклій формі послідовності цифр. Особливістю мови С є те, що при записі цілих чисел можна використовувати суфікси, які означають довжину цілого (кількість двійкових розрядів). Наприклад, суфікс L означає довге ціле зі знаком, а суфікс UL – беззнакове довге ціле. Числа дійсного типу мають дві форми запису:

1) з фіксованою крапкою – у вигляді цілої та дробової частини, розділеної крапкою: 25.34, 3.1415;

2) з плаваючою крапкою – у вигляді мантиси та порядку, розділених буквою Е: 0.31415Е1.

*Рядки* – це послідовність символів, записаних між парою подвійних лапок: “Hello World!”.

Будь-яка програма задає конкретні команди, які виконує комп’ютер. Ці команди мають вигляд операторів мови програмування, і вони, як правило, передбачають наявність певних величин або об’єктів над якими виконуються дії задані операторами. Ці величини прийнято називати *даними*. Іншими словами, в найбільш загальному випадку можна стверджувати, що ***програма=оператори+дані***. Для представлення даних в програмі найчастіше використо-вуються такі програмні об’єкти як змінні та константи. *Змінні* – це величини, які можуть змінюватися в процесі виконання програми, а *константи* – це величини, які не змінюють свого значення протягом усього часу виконання програми. Константи записуються або значеннями: 3.1415, 10, або іменами : Pi, EPS – іменовані константи.

Всі змінні в С-програмі повинні бути обов’язково оголошені (задекларовані) до того як вони будуть використані. Оголошення змінної має задавати *тип даних змінної* та ідентифікатор змінної, а також може містити початкове значення змінної *(ініціалізація* змінної). Типи даних у мові С поділяються на *прості (скалярні) та складені (структуровані)*. До скалярних типів відносяться:

1) ***char*** – одиничний байт, що містить один символ;

2) ***int*** – ціле число;

3) float – число з плаваючою крапкою одиничної точності;

4) ***double*** – число з плаваючою крапкою подвійної точності.

Для розширення базових типів використовуються *кваліфікатори*:

1) ***short –*** короткий; 2) ***long*** – довгий.

Ці кваліфікатори застосовуються до цілого типу: ***short int*** (можна писати просто short) – короткий цілий, **long int** (скорочено long) – довгий цілий. Наприклад, якщо ціле число типу int може займати в пам’яті машини або 16 біт або 32 біти, то long займає 32 біти, а short – 16 біт.

Кваліфікатор *long* може розширювати тип double. Тип ***long double*** – числа з плаваючою крапкою підвищеної точності.

Кваліфікатор 1) ***signed*** – із знаком; 2) ***unsigned*** – без знака. Застосовуються до типів int i char. Якщо значенню типу char відводиться 8 біт (1 байт), то unsigned приймає значення від 0 до 255, а signed від -128 до 127.

У мові C не існує логічного або булевського типу, хоча логічні операції використо-вуються. Треба запам’ятати, що значенню “істина” відповідає “не нуль”, тобто будь-яке число, що не дорівнює нулю, а “не істина” – “нуль”.

Вибір імені (ідентифікатора) змінної бажано узгоджувати з її змістом. Приклади оголо-шення та ініціалізації змінних:

char sym, ls=’A’;

int n=10, m, k;

float eps=0.001;

long int number;

unsigned char s;

До будь-якої змінної в оголошенні можна використати кваліфікатор ***const***, тоді змінна перетворюється в константу, тобто її значення при виконанні програми не змінюється:

const int n=20;

const double pi=3.1415;

Будь-яка мова програмування високого рівня (С не є виключенням) має свої вимоги та правила щодо організації програми. Традиційно для пояснення структури програми мовою С розглядається найпростіша програма, яка виводить певне текстове повідомлення (наприклад, “Моя перша програма мовою С!”) на екран монітора. Вихідний текст (адаптований до середовища програмування MS Visual C++ 2010 Express Edition) такої програми має вигляд:

// Моя перша програма мовою С

#include <stdio.h>

main()

{

printf("Моя перша програма мовою С! \n");

return 0;

}

Ця проста програма ілюструє цілий ряд важливих особливостей мови С. Розглянемо детально кожний рядок цієї програми.

Перший рядок

// Моя перша програма мовою С

починається з символа //, який означає, що наступний після нього текст є коментарем. Коментарі використовуються для того, щоб документувати програму і полегшити її розуміння.

Коментарі допомагають іншим людям читати і розуміти Вашу програму. Коментарі ігноруються компілятором С, і тому вони не викликають ніяких дій з боку комп’ютера. Як правило, коментар у першому рядку програми просто описує призначення програми. Коментар, який починається з символа // називається *однострічковим* коментарем, тому, що він закінчується в кінці поточного рядка. *Багатострічковий* коментар в С повинен розміщатися між парою символів /\* та \*/.

Рядок

#include < stdio.h>

є *директивою препроцесора* мови С, яка дає вказівку препроцесору включити в текст програми вміст файлу **stdio.h**. Сам файл з розширенням .h прийнято називати *заголовочним файлом* або *header-файлом*. Рядки програми на С, які починаються з символа # є директивами і обробляються препроцесором перед початком компіляції програми. В даному випадку заголовний файл **stdio.h** містить оголошення функцій вводу/виводу.

Рядок

main()

повинен бути пристунім у кожній програмі на С. Пара круглих дужок після слова **main** означають, що main – це програмний блок, який називається *функцією*. Будь-яка С-програма складається з однієї чи декількох функцій, одна з яких повинна обов’язково називатися main. Функція main називається *головною функцією* і вона є точкою входу в програму. Це означає, що кожна програма на С починає виконуватися з функції main, навіть якщо ця функція не є першою в тексті програми.

Ліва фігурна дужка { починає *тіло* кожної функції. Відповідно, права фігурна дужка } повинна завершувати кожну функцію. Проводячи аналогію з мовою Паскаль, пара фігурних дужок {} відповідає ключовим словам begin end мови Паскаль.

Рядок

printf("Моя перша програма мовою С! \n");

задає команду комп’ютеру вивести на екран рядок символів, поміщених в подвійні лапки. Ця команда має вигляд виклику функції виведення на консоль. Такий виклик складається з ідентифікатора (назви) функції printf(), рядка-параметра “Моя перша програма мовою С! \n” та символа крапки з комою ; , який є *ознакою завершення оператора чи команди*. Кожний оператор або команда в С повинні закінчуватися цим символом.

Однак, якщо виконати цю програму, то можна побачити, що символи \n з даного літералу не виводяться на екран. Символ \ (*обернений слеш*) називається *символом переходу* або *escape-символом*. Він позначає собою початок так званої *керуючої послідовності* або *escape-послідовності*, яка задає виведення певного спеціального символу. У нашому випадку керуюча послідовність **\n** означає *початок нового рядка* і викликає переміщення курсора (тобто індикатора поточної позиції на екрані) на початок наступного рядка на екрані. Деякі інші, найчастіше вживані escape-послідовності наведені в табл.1.

Таблиця 1. Деякі керуючі послідовності

|  |  |
| --- | --- |
| Керуюча послідовність | Опис призначення |
| \n | Новий рядок, курсор встановлюється на початок нового рядка |
| \t | Горизонтальна табуляція, курсор встановлюється на наступну позицію табуляції |
| \r | Повернення каретки, курсор встановлюється на початок поточного рядка |
| \\ | Друк символа \ |

Рядком

return 0;

має завершуватися функція main(), якщо вона має тип результату відмінний від **void**. Відсутність типу результату функції main() в заголовку означає, що вона повертає результат типу **int** за замовчуванням. Сам оператор **return** вказує на вихід з функції (тобто на завершення виконання функції), причому значення 0 означає, що функція main(), а отже і сама програма, завершилася успішно.

На даному етапі вивчення мови С відомі нам елементарні операції можна поділити на три групи:

1) арифметичні операції;

2) операції порівняння;

3) операція присвоєння.

Таблиця 2. Елементарні операції мови С

|  |  |  |  |
| --- | --- | --- | --- |
| Назва операції | Операція в С | Алгебраїчний вираз | Вираз мовою С |
| Додавання | + | a+5 | a+5 |
| Віднімання | - | x-y | x-y |
| Множення | \* | abc | a\*b\*c |
| Ділення | / | 2z | z/2 |
| Остача від ділення | % | a mod 2 | a%2 |
| Інкремент | ++ | a+1 | a++ |
| Декремент | -- | k-1 | k-- |
| Рівність | == | x=y | x==y |
| Нерівність | != | x y  | x!=y |
| Більше | > | x>y | x>y |
| Менше | < | x<y | x<y |
| Більше або рівне | >= | x y  | x>=y |
| Менше або рівне | <= | x y  | x<=y |
| Присвоєння | = | a=b+c | a=b+c |

При використанні операцій важливе значення відіграють пріоритет (старшинство) та асоціативність (порядок виконання) операцій. Для тих операцій з якими ми вже познайомилися групування за старшинством (в порядку спадання пріоритету) наведено у таблиці 3.

Таблиця 3. Пріоритет та асоціативність операцій

|  |  |
| --- | --- |
| Операція | Асоціативність |
| () | Зліва направо |
| ++ -- | Справа наліво |
| \* / % | Зліва направо |
| + - | Зліва направо |
| < <= > >= | Зліва направо |
| == != | Зліва направо |
| << >> | Зліва направо |
| = | Справа наліво |

Таблиця 4. Основні математичні функції мови С

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я функції | Математичний запис | Тип і межі зміни аргументів | Тип результату |
| sin(x) | sin x | double | double |
| cos(x) | cos x | double | double |
| tan(x) | tg x | double | double |
| asin(x) | arcsin x | double xϵ[-1,1] | [-π/2, π /2] |
| acos(x) | arccos x | xϵ [-1,1] | [0, π] |
| atan(x) | arctg x | xϵ [- π /2, π /2] | double |
| sinh(x) | sh x | double | double |
| cosh(x) | ch x | double | double |
| tanh(x) | th x | double | double |
| exp(x) | ex | double | double |
| log(x) | ln x | x>0 | double |
| log10(x) | lg x | x>0 | double |
| pow(x,y) | xy | x≠0; y>0 | double |
| sqrt(x) | x | x≥0 | double |
| fabs(x) | |x| | double | double |
| ldexp(x,n) | x . 2n | x-double, n-int | double |
| fmod(x,y) | Залишок від ділення дійсних чисел х на у | double | double |

## Код програми

Назва файлу: **Task1.c**

#include <stdio.h>

#include <math.h>

int main(void)

{

const float fEulerConstant = exp(1.0);

float fX, fZ, fB;

scanf\_s("%f", &fX);

scanf\_s("%f", &fZ);

fB = fX \* (atan(fZ) + pow(fEulerConstant, -1.0 \* (fX + 3.0)));

printf("%f", fB);

## }

Назва файлу: **Task2.c**

#include <stdio.h>

int main(void)

{

int iEightDigit, iProduct=1, iZeroCount=0,iDigitCount=0, iDigit=0;

scanf\_s("%i", &iEightDigit);

nextDigit:

iDigit = iEightDigit % 10;

if (iDigit == 0)

iZeroCount++;

else

iProduct \*= iDigit;

iEightDigit /= 10;

iDigitCount++;

if (iDigitCount < 8)

goto nextDigit;

printf("Double product: %i \n", iProduct \* 2);

printf("Zero count: %i", iZeroCount);

}

## Протокол роботи

Завдання 1.

Протокол вводу:

-0.622

5.541

Протокол виводу на екран:

*-*0.923658

На рис. 1 наведено результати виконання програми.

Зображення, що містить текст

Автоматично згенерований опис

Рис. 1. Результат виконання програми 1

Завдання 2.

Протокол вводу:

12340056

Протокол виводу на екран:

Double product: 1440

Zero count: 2

Зображення, що містить текст

Автоматично згенерований опис

Рис. 2. Результат виконання програми 2

## Висновки

На лабораторній роботі розглянуто основні поняття мови С, а саме: алфавіт та елементарні конструкції мови С, поняття змінних і констант, структуру програми, операції мови С та основні математичні функції. В першому завданні створено програму для обчислення заданого арифметичного виразу. Для виконання другого завдання розглянуто оператори галуження if та else та алгоритм поділу багатоцифрового числа на окремі цифри, створено програму з використанням цього алгоритму.