Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 9

з навчальної дисципліни

“Базові методології та технології програмування”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ РОЗГАЛУЖЕНИХ ТА ІТЕРАЦІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення Доренський О. П.

ВИКОНАВ

студент академічної групи КІ-23

Замай Володимир

ПЕРЕВІРИВ

викладач кафедри кібербезпеки та програмного забезпечення Дрєєва Г.М.

Кропивницький – 2024

**Мета:** полягає у набутті ґрунтовних вмінь і практичних навичок реалізації технології модульного програмування, застосування операторів С/С++ арифметичних, логічних, побітових операцій, умови, циклів та вибору під час розроблення статичних бібліотек, заголовкових файлів та програмних засобів у кросплатформовому середовищі Code::Blocks.

**Варіант 6**

**Алгоритм 9.1**

1. Зчитайте кількість відпрацьованих годин у понеділок-п'ятницю.
2. Обчисліть базову зарплату, яка складається з оплати за 5 робочих днів по 8 годин на день з погодинною ставкою 40 грн/год.
3. Перевірте, чи відпрацьовано більше 40 годин. Якщо так, обчисліть премію за кожну додаткову годину.
4. Підрахуйте загальну зарплату, додавши базову зарплату і премії за додаткові години.
5. Обчисліть суму кожного податку: 15% податку на прибуток, 2% до Пенсійного фонду, 0.6% до Фонду загальнообов'язкового державного соціального страхування на випадок безробіття, і 1% загальнообов'язкового державного соціального страхування у зв'язку з тимчасовою втратою працездатності.
6. Знайдіть загальну суму вирахованих податків.
7. Обчисліть суму до видачі працівникові, віднявши загальну суму вирахованих податків з загальної зарплати.
8. Виведіть результати.

**Лістинг коду програми 9.1**

#include <iostream>

#include <iomanip>

#include <locale.h>

int main() {

setlocale(LC\_ALL, "ukr");

const double hourly\_rate = 40.0;

const double bonus\_per\_extra\_hour = 300.0;

int hours\_worked;

std::cout << "Введіть кількість відпрацьованих годин: ";

std::cin >> hours\_worked;

double base\_salary = 5 \* 8 \* hourly\_rate;

double extra\_hours\_pay = (hours\_worked > 40) ? (hours\_worked - 40) \* bonus\_per\_extra\_hour : 0;

double total\_salary = base\_salary + extra\_hours\_pay;

double income\_tax = total\_salary \* 0.15;

double pension\_fund = total\_salary \* 0.02;

double social\_insurance = total\_salary \* 0.006;

double temporary\_disability\_insurance = total\_salary \* 0.01;

double total\_taxes = income\_tax + pension\_fund + social\_insurance + temporary\_disability\_insurance;

double net\_salary = total\_salary - total\_taxes;

std::cout << "Нарахована сума зарплати: " << total\_salary << " грн." << std::endl;

std::cout << "Розмір вирахованих податків:" << std::endl;

std::cout << "Податок на прибуток: " << income\_tax << " грн." << std::endl;

std::cout << "Внесок до Пенсійного фонду: " << pension\_fund << " грн." << std::endl;

std::cout << "Внесок до Фонду загальнообов'язкового державного соціального страхування на випадок безробіття: "

<< social\_insurance << " грн." << std::endl;

std::cout << "Внесок до загальнообов'язкового державного соціального страхування у зв'язку з тимчасовою втратою працездатності: "

<< temporary\_disability\_insurance << " грн." << std::endl;

std::cout << "Сума до видачі працівникові (з вирахуваними податками): " << net\_salary << " грн." << std::endl;

std::cout << "Натисніть будь-яку клавішу та Enter для завершення програми...";

char any\_key;

std::cin >> any\_key;

return 0;

}

**Алгоритм 9.2**

1. Створіть таблицю відповідності розмірів шкарпеток для української системи та систем США та ЄС.
2. Зчитайте розмір чоловічих шкарпеток в українській системі.
3. Перевірте, чи є введений розмір в таблиці відповідності.
4. Якщо розмір знайдено, виведіть відповідні розміри у системах США та ЄС.
5. Якщо розмір не знайдено, виведіть повідомлення про помилку.

Ось узагальнений алгоритм:

1. Ініціалізувати таблицю відповідності розмірів шкарпеток для української системи та систем США та ЄС.
2. Зчитати розмір чоловічих шкарпеток в українській системі.
3. Перевірити, чи є введений розмір у таблиці відповідності.
4. Якщо розмір знайдено, вивести відповідні розміри у системах США та ЄС.
5. Якщо розмір не знайдено, вивести повідомлення про помилку. Цей алгоритм можна легко реалізувати в будь-якій мові програмування

**Лістинг коду програми 9.2**

#include <iostream>

#include <map>

using namespace std;

map<int, pair<string, string>> getSockSizeChart() {

map<int, pair<string, string>> sockSizeChart = {

{2, {"37/38", "37/38"}},

{3, {"39/40", "39/40"}},

{4, {"41/42", "41/42"}},

{5, {"43/44", "43/44"}},

{6, {"45/46", "45/46"}}

};

return sockSizeChart;

}

void printSockSizeConversion(int size, const pair<string, string>& conversion) {

cout << "Розмір " << size << " (Україна) відповідає розмірам " << conversion.first << " (США) та " << conversion.second << " (ЄС)" << endl;

}

int main() {

setlocale(LC\_ALL, "ukr");

map<int, pair<string, string>> sockSizeChart = getSockSizeChart();

cout << "Таблиця відповідності розмірів шкарпеток (чол.):" << endl;

cout << "Україна\tСША\t\t\tЄС" << endl;

for (const auto& pair : sockSizeChart) {

cout << pair.first << "\t" << pair.second.first << "\t\t" << pair.second.second << endl;

}

int userInput;

cout << "Введіть розмір шкарпеток за українською системою: ";

cin >> userInput;

auto it = sockSizeChart.find(userInput);

if (it != sockSizeChart.end()) {

printSockSizeConversion(userInput, it->second);

} else {

cout << "Введений розмір не знайдено в таблиці." << endl;

}

system("pause");

return 0;

}

**Алгоритм 9.3**

1. Прийняти вхідні дані - натуральне число N від 0 до 7483650.
2. Визначити біт D1 числа N:
   * Використати побітове І для отримання значення біта D1: D1 = (N & 2) >> 1.
3. Вибрати метод підрахунку залежно від значення біта D1:
   * Якщо D1 рівний 1:
     + Обчислити кількість двійкових нулів у числі N.
   * Якщо D1 рівний 0:
     + Обчислити кількість двійкових одиниць у числі N.
4. Повернути результат підрахунку.
5. Вивести результат.

**Лістинг коду програми 9.3**

#include <iostream>

using namespace std;

int countZeros(int n) {

int count = 0;

while (n > 0) {

if (n % 2 == 0) {

count++;

}

n /= 2;

}

return count;

}

int countOnes(int n) {

int count = 0;

while (n > 0) {

if (n % 2 == 1) {

count++;

}

n /= 2;

}

return count;

}

int main() {

setlocale(LC\_ALL, "ukr");

int N;

cout << "Введіть натуральне число N від 0 до 7483650: ";

cin >> N;

int D1 = (N & 2) >> 1;

int result = D1 ? countZeros(N) : countOnes(N);

cout << "Результат: " << result << endl;

system("pause");

return 0;

}

**Висновок:**

Наведені лістинги коду програм та відповідні алгоритми демонструють практичні застосування різних аспектів програмування.

У лістингу програми 9.1 представлений приклад розрахунку зарплати працівника з урахуванням різних видів податків та внесків. Ця програма демонструє використання змінних, умовних операторів та арифметичних операцій для обчислення результату на основі вхідних даних.

У лістингу програми 9.2 показано використання контейнера map для створення таблиці відповідності розмірів чоловічих шкарпеток між українською системою та системами США та ЄС. Ця програма використовує пари ключ-значення для збереження відповідностей та демонструє простий спосіб пошуку даних у відображенні.

У лістингу програми 9.3 реалізовано алгоритм для підрахунку кількості двійкових нулів або одиниць у введеному натуральному числі в залежності від значення його першого біту. Ця програма показує використання побітових операцій та циклу для обчислення результату згідно з визначеним алгоритмом.

Загальний висновок полягає в тому, що наведені приклади демонструють різні аспекти програмування, такі як обчислення, взаємодія з користувачем, використання контейнерів та роботу з бітовими операціями. Ці приклади можуть бути використані для навчання та розуміння основ програмування.