 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №5**

з дисципліни «Алгоритмізація та програмування»

на тему: «Алгоритми розгалуженої структури (оператор перемикання)»

Варіант №17

**Виконав:**

студент гр. БС-81

Сєров О. В.

**Перевірив:**

доцент каф. БМК

к.т.н. Алхімова С.М.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2019

**Завдання:**

1. Вивчити теоретичні основи написання алгоритмів розгалуженої

структури з використанням оператора перемикання.

1. Відповідно до свого варіанту визначити умови, скласти список різних варіантів отримання початкових даних, визначити вираз для оператора перемикання та за допомогою формул описати варіанти виконання необхідних дій.
2. Побудувати блок-схему алгоритму вирішення завдання.
3. Відповідно до свого варіанту написати програму для вирішення наведеного у варіанті завдання з передбаченням ситуації неправильного введення вихідних даних:

**Вводиться число Т - порядковий номер дня в році. Визначити номери місяця M і дня D, що відповідають Т. Наприклад, якщо Т = 365, то М = 12, а D = 31.**

1. Скласти і захистити звіт по роботі.

**🞏 Комп’ютерний практикум без зауважень**

**🞏 Комп’ютерний практикум має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до блок-схеми:**

**🞏 блок-схема не відповідає коду**

**🞏 в блок-схемі присутній код**

**🞏 виконані не за стандартом:**

**🞏 блок умови 🞏 визначений процес (функція)**

**🞏 оператор вибору 🞏 перехід**

**🞏 цикл 🞏 розміри блоків**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 використано глобальні змінні**

**🞏 типи даних визначені хибно**

**🞏 недостатня декомпозиція на функції користувача**

**🞏 функція main містить лише виклик іншої функції**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 результати виконання програми на рисунках не відповідають коду**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 незнання теоретичного матеріалу**

**🞏 маються інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Блок-схема:**



**Код:**

#include <iostream>

#include <stdio.h>

#include <conio.h>

int main()

{

int D = 0;

std::cout << "Enter day from 1 to 365: ";

std::cin >> D;

if (D >= 1 && D <= 365)

{

int M = 1;

while (D > 31)

{

switch (M)

{

case 1: case 3: case 5: case 7: case 8: case 10: D -= 31; break;

case 4: case 6: case 9: case 11: D -= 30; break;

case 2: D -= 28 ; break;

}

M++;

}

printf("You choose the right day!\nMonth: %d\nDay: %d", M, D);

}

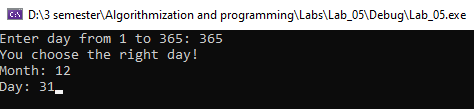
else printf("\nIncorrect input!");

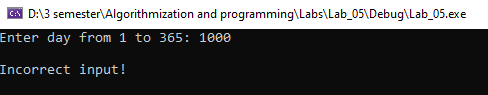
\_getch();

return 0;

}

**Результати:**

****

****

**Контрольні питання:**

1. ***В яких випадках слід використовувати оператор перемикання switch?***

Оператор перемикання використовується, коли необхідно вибрати одну чи більше (коли відсутній break) з декількох альтернатив процесу обробки даних, тобто одну з n можливих дій в залежності від константних значення ключа (перемикача).

1. ***Визначити синтаксис оператора перемикання.***

Формат оператора перемикання:

switch (expression)

{

case constant-expression-1 : { statement; } [break;]

case constant-expression-2 : { statement; } [break;]

…

case constant-expression-n : { statement; } [break;]

[default:{statement;} [break] ]

}

де, expression – вираз (ключ), який являє собою певне значення, що порівнюється і мітками, statement – складний або простий оператор, break – оператор переривання, що завершує роботу switch, default – дія, що виконується, якщо значення ключа не співпало ні з одною міткою.

1. ***Чи можуть case-гілки з одного оператора switch мати однакові значення міток?***

Мітки case-гілок не можуть мати одне і те ж саме значення.

1. ***Як default впливає на виконання тіла оператора перемикання?***

Якщо жодна з міток не спрацювала, гілка default виконає свої операції.

1. ***Чи можуть гілки case та default бути розташовані у довільному порядку?***

Так. Гілки можуть розташовуватись у довільному порядку в операторі switch.

1. ***Чи може під час роботи програми значення виразу оператора перемикання співпасти відразу з декількома кейсами?***

Константні вирази не можуть співпасти, тож і кілька кейсів відразу не можуть виконатись.

1. ***Що може бути використано в якості міток в операторі перемикання switch?***

В якості міток можуть використовуватися цілі або символьні константи (крім рядків).

e.g.:

case 1: case 2:

case ‘a’: case ‘A’:

1. ***Навіщо у операторі перемикання використовують оператор переривання?***

Оператор переривання потрібен для дострокового завершення роботи switch, при успішному виконанні одної з гілок.

У випадку відсутності break – виконуються усі подальші прості або складені оператори до наступного break чи кінця switch.

1. ***Коли краще використовувати умовний оператор, а коли оператор перемикання?***

Умовний оператор краще використовувати при неявних умовах або при необхідності охвату деякого проміжку значень.

Оператор перемикання краще використовувати при визначених константних умовах та коли кількість міток не буде заважати комфортності читання коду.

1. ***Чи може оператор перемикання switch мати вкладений оператор умови, ще один оператор перемикання?***

При виконанні однієї з гілок може бути декілька операторів, отже є можливість вкласти оператор умови чи ще один оператор перемикання. У таких випадках для кращого читання коду є сенс написати окрему функцію з необхідними вкладеними операторами.