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: «Алгоритмирозгалуженої структури(оператор перемикання)»

Варіант №2

**Виконав:**

студент гр. БС-71

Батуркіна А.М.

**Перевірив:**

ас. каф. БМК

Рисін С. В.

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

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

(підпис викладача)

Київ-2018

**🞏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 mainмістить 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:**

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

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

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

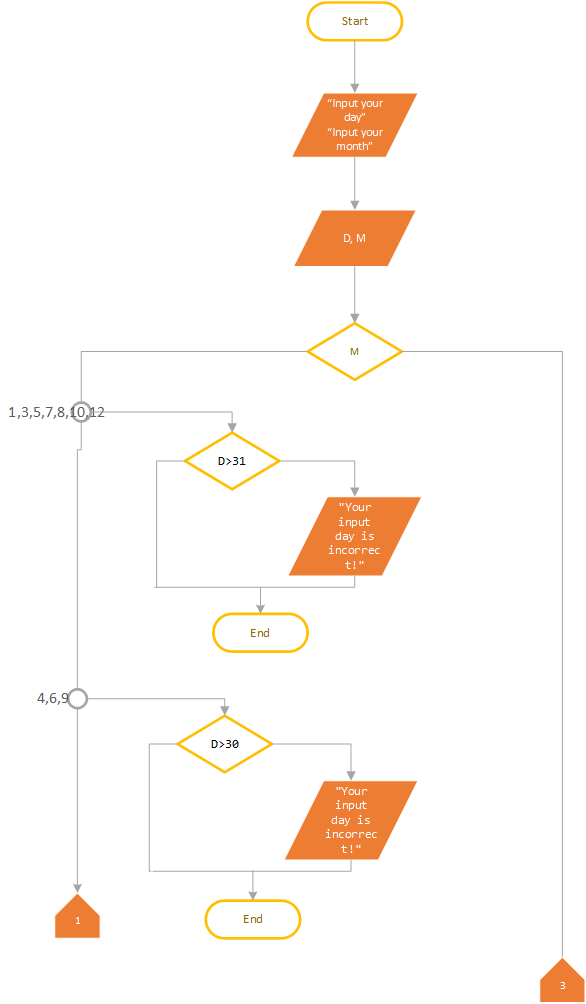
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

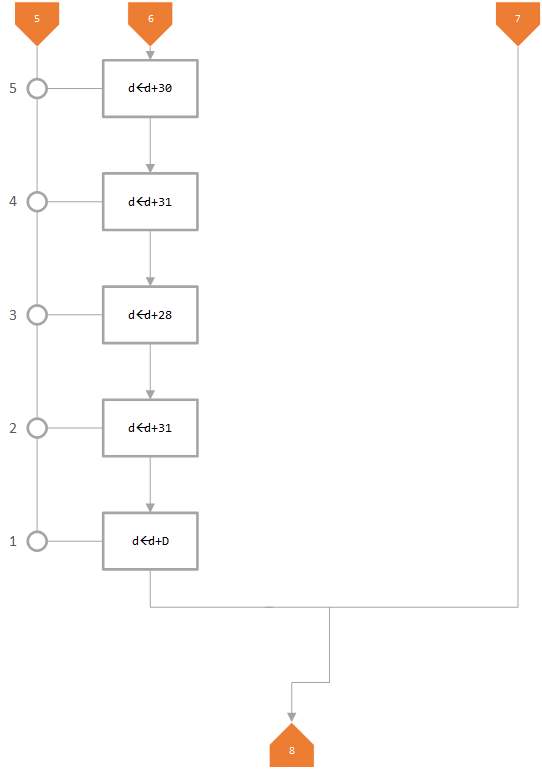
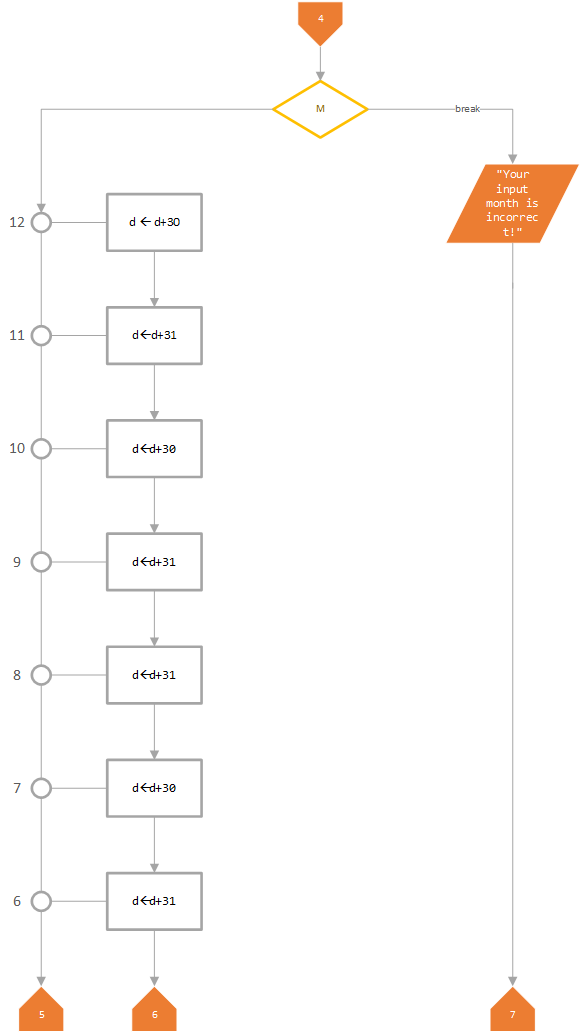
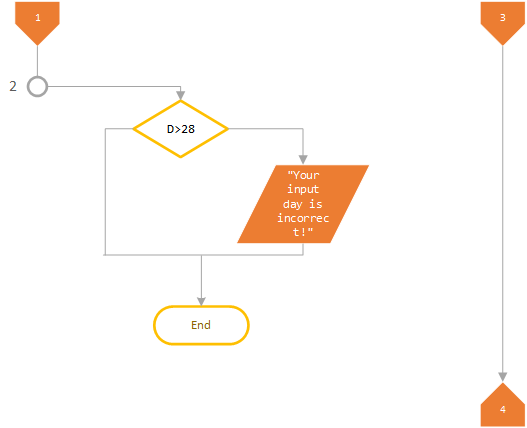
**Завдання:**

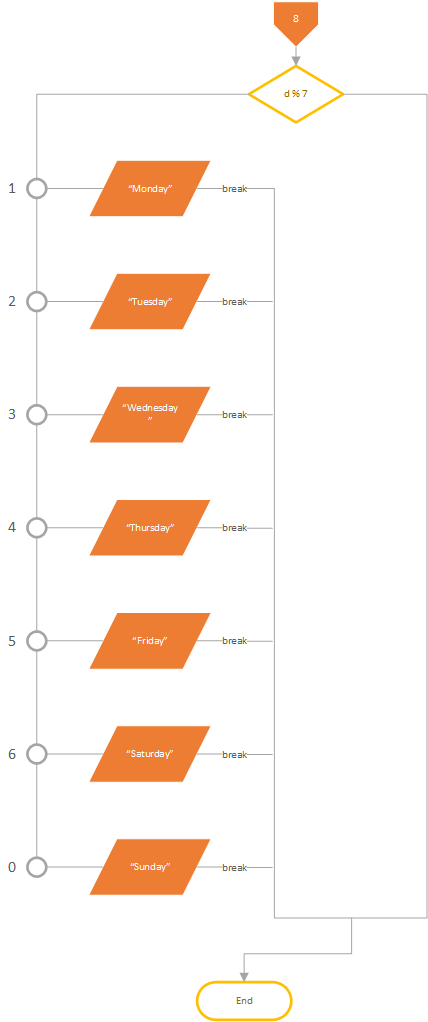
1. Вивчити теоретичні основи написання алгоритмів розгалуженої структури з використанням оператора перемикання.
2. Відповідно до свого варіанту визначити умови, скласти список варіантів отримання початкових даних, визначити вираз для оператора перемикання та за допомогою формул описати варіанти виконання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

*Вводяться номери місяця M та D дня. Визначити день тижня з даними M та D,якщо вважати, що рік починається з понеділка.*

1. Скласти та захистити звіт по роботі.

**Блок-схема:**

****

****

**Лістинг:**

#include <iostream>

#include <conio.h>

using namespace std;

int main()

{

int D;

int M;

int d = 0;

cout << "Input your day: ";

cin >> D;

cout << "Input your month: ";

cin >> M;

switch (M)

{

case 1: case 3: case 5: case 7: case 8: case 10: case 12:

if (D > 31)

{

cout << "Your input day is incorrect!";

\_getch();

return 0;

}

case 4: case 6: case 9: case 11:

if (D > 30)

{

cout << "Your input day is incorrect!";

\_getch();

return 0;

}

case 2:

if (D > 28)

{

cout << " ";

\_getch();

return 0;

}

}

switch (M) //считаем номер введенного дня в году

{

case 12: d = d + 30;

case 11: d = d + 31;

case 10: d = d + 30;

case 9: d = d + 31;

case 8: d = d + 31;

case 7: d = d + 30;

case 6: d = d + 31;

case 5: d = d + 30;

case 4: d = d + 31;

case 3: d = d + 28;

case 2: d = d + 31;

case 1: d = d + D;

default:

break;

cout << "Your input month is incorrect!";

}

switch (d % 7)

{

case 1: cout << "Monday" << endl;

break;

case 2: cout << "Tuesday" << endl;

break;

case 3: cout << "Wednesday" << endl;

break;

case 4: cout << "Thursday" << endl;

break;

case 5: cout << "Friday" << endl;

break;

case 6: cout << "Saturday" << endl;

break;

case 0: cout << "Sunday" << endl;

break;

}

\_getch();

}

**Результати:**





**Контрольні питання:**

1. В яких випадках слід використовувати оператор перемикання switch?

* Оператор перемикання використовується , коли необхідно вибрати одну з декількох альтернатив процесу обробки даних інтегрального типу(char, short, int, long, long long) , тобто одну з n можливих дій в залежності від значення ключа(перемикача).

1. Визначити синтаксис оператора перемикання.

* Синтаксис оператора перемикання:

switch(exp)

{

case const-exp-1: {statement; [break;]}

case const-exp-1: {statement; [break;]}

…

case const-exp-n: {statement; [break;]}

[default: {statement; break;}]

}

де exp – вираз, результатом якого може бути значення, що визначає одну з мітокconst-exp; statement – простий або складений оператор ; break – оператор переривання,що завершує виконання оператора switch, але не є обов’язковим ; default – ключове слово, після якого розташовані оператори дій, що виконуються у випадку незбігу значення exp із жодною міткою const-exp(застосовується тільки у випадку повної форми оператора перемикання); default - ключове слово, після якого розташовані оператори, що виконуються у випадку прийняття exp значення, що не збігається з жодною міткою.

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

* Ні, не можуть.

1. Як default впливає на виконання тіла оператора перемикання?

* default – ключове слово, після якого розташовані оператори дій, що виконуються у випадку незбігу значення exp із жодною міткою const-exp(застосовується тільки у випадку повної форми оператора перемикання)

1. Чи можуть гілки case та default бути розміщенні в довільному порядку?

* Так, можуть, так як гілка defaultвиконується останньою після оцінки всіх міток case, незважаючи на їх порядок.

1. Чи може під час виконання програми значення умовного виразу оператора switchспівпасти одразу з декількома різними мітками?

* Ні, не може, так як оператор switch може виконувати тільки операції перевірки строгої рівності.

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

* В якості міток мають бути використані константні значення, які порівнюються на рівність з виразом оператора(exp). Так як константні вирази перевіряються на рівність з exp, то вони мають бути того самого типу даних(інтегрального типу(char, short, int, long, long long)). Константні вирази не можуть мати виклики функцій або змінні.

1. Навіщо в операторі switch використовувати оператор переривання break?

* Якщо оператор break є відсутнім наприкінці операторів виконуваної case- гілки, то будуть почергово виконані всі наступні оператори до break або до кінця оператора switch для всіх case- гілок незалежно від значення їх міток const-exp.

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

* Оператор перемикання краще використовувати, коли є n(альтернатив) варіантів дії, а оператор if - коли необхідно обрати один з двох варіантів дій. Також треба пам’ятати, що switch не може мати дві гілки однакового значення і може виконувати тільки операції перевірки строгої рівності, в той час, коли оператор if може обчислювати логічні вирази і нестрогі рівності. Також, завдяки оператору if можна перевіряти данні будь-якого типу, на відміну від switch.

1. Чи може оператор перемикання switch мати вкладений оператор умови if, ще один вкладений оператор перемикання switch? Відповідь пояснити.

* Так, може. Адже, за форматом:

….

case const-exp-1: {statement }; [break;]

….statement – простий або складений оператор, а умовний оператор та сам оператор перемикання задовольняють цю умову формату.