



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
КАФЕДРА МЕДИЧНОЇ КІБЕРНЕТИКИ ТА ТЕЛЕМЕДИЦИНИ

Лабораторна робота № 5

з дисципліни «Алгоритмізація та програмування»
на тему: "Алгоритми послідовності (лінійної) структури"
Варіант 16

Виконал:

Студент групи БС-32

Хоміцевич Микита

Перевірила:

викл.

Матвійчук А.О.

Задание

16	Вводиться число студентів $n \leq 30$. Вивести фразу «У нашій групі n студентів», погодивши слово «студент» з числом n . Передбачити ситуацію неправильного введення вихідних даних.
----	---

☐ Лабораторна робота виконана без зауважень

☐ Лабораторна робота має зауваження:

☐ присутні зауваження до блок-схем:

☐ виконані не за стандартом

☐ не відповідають коду

☐ присутній код

☐ інші зауваження:

☐ присутні зауваження до коду:

☐ задача завдання вирішена хибно

☐ код програми не компілюється

☐ використано глобальні змінні

☐ недостатня декомпозиція на функціонуванні користувача

☐ статичні змінні при роботі з масивами

☐ оформлення коду

☐ присутні зайві символи «{» та «}»

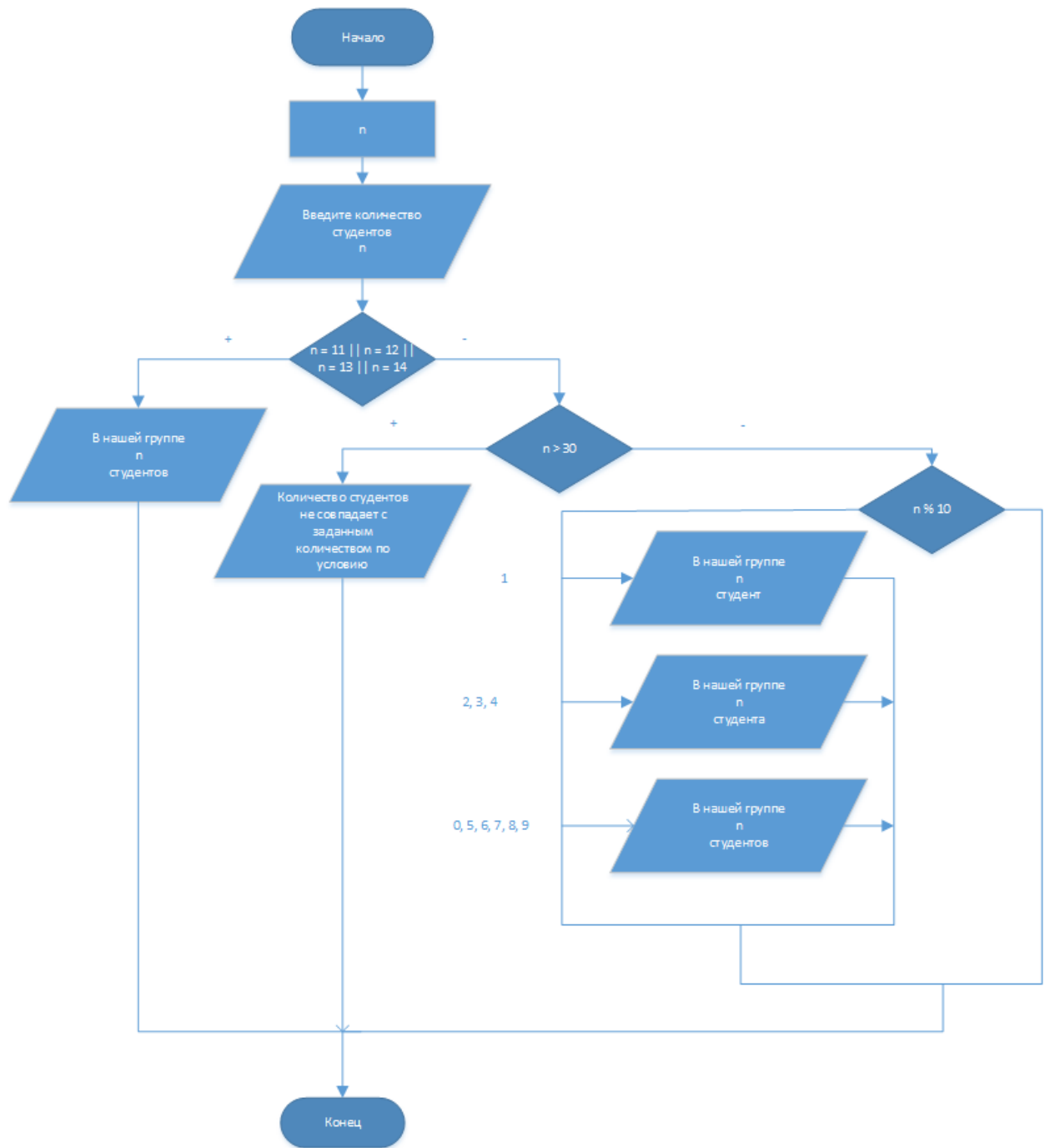
☐ інші зауваження:

☐ невірні відповіді на запитання:

☐ №1 ☐ №2 ☐ №3

☐ мають інші зауваження:

Блок – схема



Программный код

```
#include <iostream>

#include <conio.h>

#include <locale.h>

using namespace std;

int main() {

    setlocale(LC_ALL, "RUS");

    int n;

    cout << "Введите количество студентов n=";

    cin >> n;

    if (n == 11 || n == 12 || n == 13 || n == 14) {

        cout << "В нашей группе " << n << " студентов" << endl;

    }

    else if (n > 30) {

        cout << "Количество студентов не совпадает с заданным количеством по условию"
<< endl;

    }

    else {

        switch (n % 10) {

            case 1:

                {

                    cout << "В нашей группе ";

                    cout << n;

                    cout << " студент";

                    break;

                }

            case 2: case 3: case 4:

                {

                    cout << "В нашей группе ";

                    cout << n;
```

```
        cout << " студента";  
        break;  
    }  
  
    case 0: case 5: case 6: case 7: case 8: case 9:  
    {  
        cout << "В нашей группе ";  
        cout << n;  
        cout << " студентов";  
        break;  
    }  
}  
  
_getch();  
}
```

Скриншот работы программы



Ответы на контрольные вопросы

1. В каких случаях следует использовать оператор переключения switch?

Оператор перемикаання використовується, коли необхідно вибрати одну з декількох альтернатив процесу обробки даних, тобто одну з n можливих дій в залежності від значення ключа (перемикача). *Switch* є щось на зразок узагальнення оператора *if* (вибір одного з двох варіантів) на n варіантів (альтернатив) дій.

2. Определите синтаксис оператора переключения

Оператор перемикаання має скорочену (без *default*) або повну (з *default*) форми. Формат оператора перемикаання:

```
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, результатом якого може бути значення, що визначає одну з міток *constant-expression*; *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 *expression* із жодною міткою *constant-expression* (застосовується тільки у випадку повної форми оператора перемикаання).

3. Як default влияет на использование тела оператора переключения?

При использовании ветки *default*, если наше условие не совпадает ни с одной из указанных выше веток *case*, тогда будет выбрана для выполнения ветка *default* и у нас всегда будет выдаваться заданный в программе ответ (при любых задаваемых параметрах)

4. Могут ли ветки case и default быть расположены в свободном порядке?

Да, могут. В доказательство вот пример кода:

```
#include <iostream>

using namespace std;

int main() {
    int m;

    cin >> m;

    switch(m)
```

```
{
    default : {cout << "default chain has maked earlier!"
    << endl; break;}

    case 1: {cout <<" This is a case chain "<< endl;
    break;}

}
}
```

5. Могут ли быть условные выражения вариантов одного оператора switch одинаковыми?

Нет. Вот пример нерабочего кода:

```
#include <iostream>
using namespace std;
int main() {
    int m;
    cin >> m;
    switch(m)
    {
```

```
        default : {cout << "default chain has maked earlier!"
        << endl; break;}

        case 1: {cout <<" This is a case chain " << endl;
        break;}

        case 1: {cout << "This is a case 2, like a case 1" <<
        endl; break;}

    }
}
```

6. Что может быть использовано в виде меток в операторе переключения switch?

Числовые и символьные значения. Строки не могут быть метками в case ветках.

7. Когда лучше использовать условный оператор, а когда оператор переключения?

Это зависит от многих факторах и, так же, стиля написания самого программиста. Но если не брать в учет сложных условных цепочек, состоящих из “if() {...;} else if() {...;} else {...;}”, то можно точно сказать, что **switch** гораздо удобнее использовать при больших разбегах каких-либо условий, которые можно кратко математически описать.

Switch структуру, к сожалению нельзя применить для какого либо рода логических сравнений, и здесь однозначно выигрывает **if, else** структура!