

**ТЕМА: ВВЕДЕНИЕ В ЯЗЫК
ПРОГРАММИРОВАНИЯ «C++»****Домашнее задание 1****ЗАДАНИЕ 1**

Пользователь вводит с клавиатуры целое шестизначное число. Написать программу, которая определяет, является ли введенное число — счастливым (Счастливым считается шестизначное число, у которого сумма первых 3 цифр равна сумме вторых трех цифр). Если пользователь ввел не шестизначное число — сообщение об ошибке.

Подсказка 1

Если у Вас возникли сложности с выполнением данного задания, нажмите кнопку «Подсказка». С полным решением задания вы сможете ознакомиться, нажав на кнопку «Решение»

Решение 1**ЗАДАНИЕ 2**

Пользователь вводит четырехзначное число. Необходимо поменять в этом числе 1 и 2 цифры, а также 3 и 4 цифры. Если пользователь вводит не четырехзначное число — вывести сообщение об ошибке.

Подсказка 2

Если у Вас возникли сложности с выполнением данного задания, нажмите кнопку «Подсказка». С полным решением задания вы сможете ознакомиться, нажав на кнопку «Решение»

Решение 2**ЗАДАНИЕ 3**

Пользователь вводит с клавиатуры 7 целых чисел. Напишите программу, которая определяет максимальное из этих 7 чисел.

Подсказка 3

Если у Вас возникли сложности с выполнением данного задания, нажмите кнопку «Подсказка». С полным решением задания вы сможете ознакомиться, нажав на кнопку «Решение»

Решение 3

ЗАДАНИЕ 4

Грузовой самолет должен пролететь с грузом из пункта А в пункт С через пункт В. Емкость бака для топлива у самолета — 300 литров. Потребление топлива на 1 км в зависимости от веса груза у самолета следующее:

- до 500 кг — 1 литров/км;
- до 1000 кг — 4 литров/км;
- до 1500 кг — 7 литров/км;
- до 2000 кг — 9 литров/км;
- более 2000 кг — самолет не поднимает.

Пользователь вводит расстояние между пунктами А и В, и расстояние между пунктами В и С, а также вес груза. Программа должна рассчитать какое минимальное количество топлива необходимо для дозаправки самолету в пункте В, чтобы долететь из пункта А в пункт С. В случае невозможности преодолеть любое из расстояний — программа должна вывести сообщение о невозможности полета по введенному маршруту.

Финальное задание рассчитано на самостоятельное решение. Здесь не будет подсказок и готового алгоритма. Вам необходимо применить все практические навыки, полученные из предыдущих заданий.

ПОДСКАЗКА К ЗАДАНИЮ 1

1. Любое шестизначное число (число, состоящее из 6 разрядов) можно представить как сумму его цифр, умноженных на 10 в степени, которая соответствует разряду (позиции цифры в числе справа налево, начиная с 0-вой позиции). Например, число $254238 = 2 \cdot 10^5 + 5 \cdot 10^4 + 4 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 8 \cdot 10^0$.
2. Выделение цифры из числа последовательно (столько раз сколько цифр в числе) можно выполнять также справа налево, т. е. от нулевой позиции до последней (пятой, т. к. номера позиций начинаются с 0).
3. Для выделения последней цифры на каждом шаге (на первом шаге это все число, на втором — оставшиеся первые 5 цифр) можно использовать операцию получения остатка от деления числа на 10.
4. Каким арифметическим оператором реализуется операция получения остатка от деления?
5. Для выделения оставшейся части числа можно использовать операцию целочисленного деления числа на 10.
6. Каким арифметическим оператором реализуется операция целочисленного деления?
7. Для определения, ввел ли пользователь 6-значное число, можно применить к введенному числу операцию целочисленного деления числа на 100000 (разряд первой цифры 6-значного числа, 10^5), если результат меньше 1, то число содержит меньше, чем 6 цифр, если результат больше 9, то число содержит больше, чем 6 цифр.

8. Каким оператором условного ветвления реализуется представленная в пункте 7 ситуация, когда нужно последовательно проверить несколько условий (выполниться может только одно из них) и предусмотреть альтернативный шаг в случае, когда они не выполняются?

ПОДСКАЗКА К ЗАДАНИЮ 2

1. Любое четырехзначное число (число, состоящее из 4 разрядов) можно представить как сумму его цифр, умноженных на 10 в степени, которая соответствует разряду (позиции) цифры в числе справа налево, начиная с нулевой позиции). Например, число $4238 = 4 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 8 \cdot 10^0$.
2. Выделение цифры из числа последовательно (столько раз сколько цифр в числе) можно выполнять также справа налево, т. е. от 0-вой позиции до последней (третьей, т. к. номера позиций начинаются с 0).
3. Для выделения последней цифры на каждом шаге (на первом шаге это все число, на втором — оставшиеся первые 3 цифры) можно использовать операцию получения остатка от деления числа на 10.
4. Каким арифметическим оператором реализуется операция получения остатка от деления?
5. Для выделения оставшейся части числа можно использовать операцию целочисленного деления числа на 10.
6. Каким арифметическим оператором реализуется операция целочисленного деления?
7. Для определения, ввел ли пользователь 4-значное число, можно применить к введенному числу операцию целочисленного деления числа на 1000 (разряд первой цифры 4-значного числа, 10^3), если результат меньше 1, то число содержит меньше, чем 6 цифр, если результат больше 9, то число содержит больше, чем 6 цифр.

8. Каким оператором условного ветвления реализуется представленная в пункте 7 ситуация, когда нужно последовательно проверить несколько условий (выполниться может только одно из них) и предусмотреть альтернативный шаг в случае, когда они не выполняются?
9. Для того, чтобы собрать новое число из цифр после их выделения и перестановки, нужно применить подход, представленный в подсказке 1, поместив нужные цифры в нужные позиции. Например, для числа 4238 решение будет $2 \cdot 10^3 + 4 \cdot 10^2 + 8 \cdot 10^1 + 3 \cdot 10^0$.

ПОДСКАЗКА К ЗАДАНИЮ 3

1. Для нахождения максимального из нескольких чисел нужно последовательно сравнить их парами, т. е. найденный максимум из двух первых чисел сравнивается с третьим числом и т. д.
2. В случае, если следующее сравниваемое число больше максимума, необходимо переопределить максимум .
3. Сколько условных блоков нужно реализовать?
4. Будут ли альтернативные части (*else*) в других условных блоках, кроме первого?
5. Какой из операторов условного ветвления нужно выбрать для сравнения *if- else if* или последовательность из нескольких *if*?

РЕШЕНИЕ ЗАДАНИЯ 1**Описание решения**

После ввода информации пользователем (целого числа) необходимо проверить, является ли оно 6-значным, если нет, то вывести сообщение об ошибке, если введенное число содержит именно 6 цифр, то начать его обработку — выделение цифр числа справа налево (от меньшего разряда к большему).

Для выполнения проверки на 6 цифр в числе можно применить к введенному числу операцию целочисленного деления числа на **100000** (разряд первой цифры 6-значного числа, 10^5):

Например, пользователь ввел число 12345.

$$12345 / 100000 = 0$$

(/ — оператор целочисленного деления), т.е. число содержит менее 6 цифр (5 в нашем примере), т.к. полученный результат меньше 1.

Пусть пользователь ввел 1234567:

$$1234567 / 100000 = 12,$$

т. е. число содержит более 6 цифр (7 в нашем примере), т. к. полученный результат больше 9.

Пусть пользователь ввел **123123**:

$$123123 / 100000 = 1,$$

т.е. число содержит четко 6 цифр (т.к. полученный результат находится в диапазоне от 1 до 9 включительно (или обе представленные выше ошибочные ситуации не выполнились, т.е. корректный вариант — это альтернатива ошибочных ситуаций).

Рассмотрим процесс выделения цифр из числа 123123.

На первом шаге к числу применяем операцию остатка от деления на 10, результат — последняя (шестая) цифра:

$$123123 \% 10 = 3.$$

Для получения оставшейся (для дальнейшей обработки) части числа (первых 5 цифр) выполняем операцию целочисленного деления на 10:

$$123123 / 10 = 12312.$$

Повторяем первый шаг над числом 12312 и получаем предпоследнюю (пятую) цифру:

$$12312 \% 10 = 2.$$

Для получения оставшейся части числа (первых 4 цифр) повторно выполняем над числом 12312 операцию целочисленного деления на 10:

$$12312 / 10 = 1231.$$

Продолжаем обработку по этому же принципу:

$$1231 \% 10 = 1 \text{ — четвертая цифра.}$$

$$1231 / 10 = 123$$

$$123 \% 10 = 3 \text{ — третья цифра.}$$

$$123 / 10 = 12$$

$$12 \% 10 = 2 \text{ — вторая цифра.}$$

$$12 / 10 = 1 \text{ — первая цифра.}$$

После получения всех цифр числа отдельно находим сумму первых трех, находим сумму последних трех цифр, сравниваем их.

Если первая сумма равна второй, то число — счастливое, иначе — нет.

Решение

1. Выводим строку в консоль с описанием названия программы (по желанию, данная строка кода может быть пропущена), создаем необходимое число переменных (в данной задаче все они будут целочисленные) — одна переменная для числа пользователя, шесть — для хранения цифр числа, одна — вспомогательная для хранения оставшейся части числа, две — для хранения сумм первых трех и последних трех цифр числа.

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #4.1.1\n\n";
    int userNumber, d1, d2, d3, d4, d5, d6, sum1, sum2, temp;

    return 0;
}
```

2. Выводим в консоль строку с приглашением ввести число и считываем его в переменную

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #4.1/1\n\n";
    int userNumber, d1, d2, d3, d4, newNumber, temp;
```

```
cout << "Please, enter your 6-digit number:\n";  
cin >> userNumber;  
return 0;  
}
```

3. Реализуем проверку введенного числа на 6 цифр, применяя к введенному числу операцию целочисленного деления числа на 100000. Для реализации процедуры сравнения будем использовать оператор условного ветвления `if-else if-else` и логические операторы сравнения `>` и `<`. Вначале реализуем проверку случая «результат целочисленного деления на 100000 меньше 1»:

```
if (userNumber / 100000 < 1)  
{  
    cout << "Your number includes less than 6 digits!\n";  
}
```

Потом реализуем проверку случая «результат целочисленного деления на 100000 больше 9»:

```
else if (userNumber / 100000 > 9)  
{  
    cout << "Your number includes more than 6 digits!\n";  
}
```

Случай «пользователь ввел 6-значное число», это альтернатива двум первым случаям, представляющая собой блок `else`.

```
else
{
    cout << "Your number includes exactly 6 digits!\n";
}
```

Использование конструкции `else if` вместо еще одного `if` необходимо, чтобы исключить проверку каждого из условий, если одно из них уже выполнилось. Т. е. в случае, если число уже содержит менее 6 цифр, нет необходимости проверять, не содержит ли оно более 6 цифр.

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #4.1/1\n\n";
    int userNumber, d1, d2, d3, d4, d5, d6, sum1,
        sum2, temp;
    cout << "Please, enter your 6-digit number:\n";
    cin >> userNumber;

    if (userNumber / 100000 < 1)
    {
        cout << "Your number includes less than "
                "6 digits!\n";
    }
    else if (userNumber / 100000 > 9)
    {
        cout << "Your number includes more than "
                "6 digits!\n";
    }
}
```

```
else
{
    cout << "Your number includes exactly 6 digits!\n";
}
return 0;
}
```

4. Внутри блока **else** (соответствующего случаю корректного ввода) реализуем процесс выделения цифр из числа.

4.1. Применяем к числу операцию остатка от деления на 10, результат — последняя (шестая) цифра:

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #4.1/1\n\n";
    int userNumber, d1, d2, d3, d4, d5, d6, sum1,
        sum2, temp;
    cout << "Please, enter your 6-digit number:\n";
    cin >> userNumber;
    if (userNumber / 100000 < 1)
    {
        cout << "Your number includes less than "
                "6 digits!\n";
    }
    else if (userNumber / 100000 > 9)
    {
        cout << "Your number includes more than "
                "6 digits!\n";
    }
}
```

```
else
{
    cout << "Your number includes exactly "
           "6 digits!\n";
    d6 = userNumber % 10;
}
return 0;
}
```

4.2. Для получения оставшейся (для дальнейшей обработки) части числа (первых 5 цифр) выполняем операцию целочисленного деления на 10:

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #4.1/1\n\n";
    int userNumber, d1, d2, d3, d4, d5, d6, sum1,
        sum2, temp;
    cout << "Please, enter your 6-digit number:\n";
    cin >> userNumber;
    if (userNumber / 100000 < 1)
    {
        cout << "Your number includes less than "
               "6 digits!\n";
    }
    else if (userNumber / 100000 > 9)
    {
        cout << "Your number includes more than "
               "6 digits!\n";
    }
}
```

```
    }  
    else  
    {  
        cout << "Your number includes exactly "  
                "6 digits!\n";  
        d6 = userNumber % 10;  
        temp = userNumber / 10;  
    }  
  
    return 0;  
}
```

Повторяем шаги 4.1 и 4.2 над оставшейся частью числа и получаем остальные цифры числа, выводим их в консоль.

```
#include <iostream>  
using namespace std;  
  
int main()  
{  
    cout << "Home task #4.1/1\n\n";  
    int userNumber, d1, d2, d3, d4, d5, d6, sum1,  
        sum2, temp;  
    cout << "Please, enter your 6-digit number:\n";  
    cin >> userNumber;  
  
    if (userNumber / 100000 < 1)  
    {  
        cout << "Your number includes less than "  
                "6 digits!\n";  
    }  
    else if (userNumber / 100000 > 9)  
    {
```

```
        cout << "Your number includes more than "
              << "6 digits!\n";
    }
    else
    {
        cout << "Your number includes exactly "
              << "6 digits!\n";
        d6 = userNumber % 10;
        temp = userNumber / 10;
        d5 = temp % 10;
        temp = temp / 10;
        d4 = temp % 10;
        temp = temp / 10;
        d3 = temp % 10;
        temp = temp / 10;
        d2 = temp % 10;
        d1 = temp / 10;
        cout << d1 << " " << d2 << " " << d3;
        cout << " " << d4 << " " << d5 << " " << d6;
    }

    return 0;
}
```

5. Отдельно находим сумму первых трех, находим сумму последних трех цифр, сравниваем их.

Если первая сумма равна второй, то число — счастливой, иначе — нет. Реализуем данную проверку через условную конструкцию **if-else** и логический оператор сравнения **==**.

```
#include <iostream>
using namespace std;
```



```
int main()
{
    cout << "Home task #4.1/1\n\n";
    int userNumber, d1, d2, d3, d4, d5, d6, sum1,
        sum2, temp;
    cout << "Please, enter your 6-digit number:\n";
    cin >> userNumber;

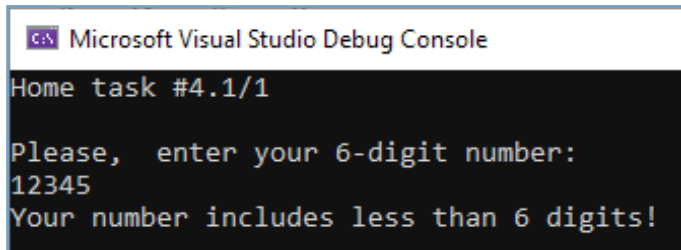
    if (userNumber / 100000 < 1)
    {
        cout << "Your number includes less than "
                "6 digits!\n";
    }
    else if (userNumber / 100000 > 9)
    {
        cout << "Your number includes more than "
                "6 digits!\n";
    }
    else
    {
        cout << "Your number includes exactly "
                "6 digits!\n";

        d6 = userNumber % 10;
        temp = userNumber / 10;
        d5 = temp % 10;
        temp = temp / 10;
        d4 = temp % 10;
        temp = temp / 10;
        d3 = temp % 10;
        temp = temp / 10;
        d2 = temp % 10;
        d1 = temp / 10;
```

```
cout << d1 << " " << d2 << " " << d3 << " ";  
cout << d4 << " " << d5 << " " << d6;  
sum1 = d1 + d2 + d3;  
sum2 = d4 + d5 + d6;  
  
if (sum1 == sum2)  
{  
    cout << "Your number brings you good luck!";  
}  
else  
{  
    cout << "It's a pity!";  
}  
}  
  
return 0;  
}
```

Результаты работы программы (в консоли):

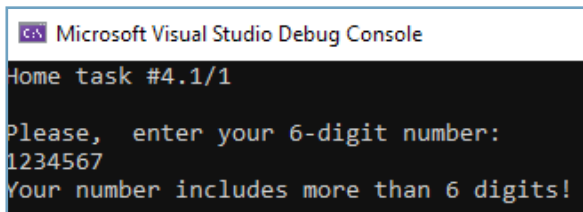
- Тест 1 — пользователь ввел число, содержащее меньше, чем 6 цифр (ошибочная ситуация №1)



```
Microsoft Visual Studio Debug Console  
Home task #4.1/1  
Please, enter your 6-digit number:  
12345  
Your number includes less than 6 digits!
```

Рисунок 1

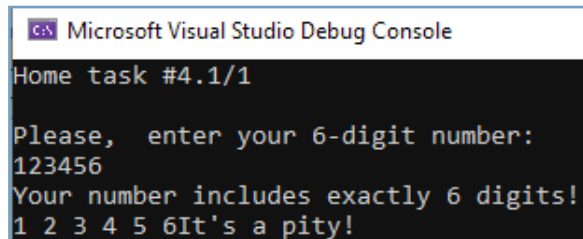
- Тест 2 — пользователь ввел число, содержащее больше, чем 6 цифр (ошибочная ситуация №2)



```
Microsoft Visual Studio Debug Console
Home task #4.1/1
Please, enter your 6-digit number:
1234567
Your number includes more than 6 digits!
```

Рисунок 2

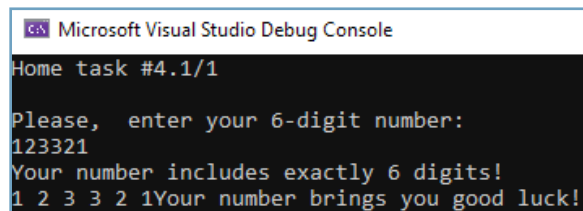
- Тест 3 — пользователь ввел число, содержащее 6 цифр, но число не счастливое



```
Microsoft Visual Studio Debug Console
Home task #4.1/1
Please, enter your 6-digit number:
123456
Your number includes exactly 6 digits!
1 2 3 4 5 6It's a pity!
```

Рисунок 3

- Тест 4 — пользователь ввел число, содержащее 6 цифр, число — счастливое



```
Microsoft Visual Studio Debug Console
Home task #4.1/1
Please, enter your 6-digit number:
123321
Your number includes exactly 6 digits!
1 2 3 3 2 1Your number brings you good luck!
```

Рисунок 4

РЕШЕНИЕ ЗАДАНИЯ 2

Описание решения

После ввода информации пользователем (целого числа) необходимо проверить, является ли оно 4-значным, если нет, то вывести сообщение об ошибке, если введенное число содержит именно 4 цифры, то начать его обработку — выделение цифр числа справа налево (от меньшего разряда к большему).

Для выполнения проверки на 4 цифры в числе можно применить к введенному числу операцию целочисленного деления числа на 1000 (разряд первой цифры 4-значного числа, 10^3):

Например, пользователь ввел число 345.

$$345 / 1000 = 0$$

(/ — оператор целочисленного деления), т. е. число содержит менее 4 цифр (3 в нашем примере), т. к. полученный результат меньше 1

Пусть пользователь ввел 1234567:

$$34567 / 1000 = 34,$$

т. е. число содержит более 4 цифр (5 в нашем примере), т. к. полученный результат больше 9.

Пусть пользователь ввел 3123:

$$3123 / 1000 = 3,$$

т. е. число содержит четко 4 цифры (т. к. полученный результат находится в диапазоне от 1 до 9 включительно (или обе представленные выше ошибочные ситуации не выполнились, т. е. корректный вариант — это альтернатива ошибочных ситуаций).

Рассмотрим процесс выделения цифр из числа 3123.

На первом шаге к числу применяем операцию остатка от деления на 10, результат — последняя (четвертая) цифра:

$$3123 \% 10 = 3.$$

Для получения оставшейся (для дальнейшей обработки) части числа (первых 3 цифр) выполняем операцию целочисленного деления на 10:

$$3123 / 10 = 312.$$

Повторяем первый шаг над числом 312 и получаем предпоследнюю (третью) цифру:

$$312 \% 10 = 2.$$

Для получения оставшейся части числа (первых 2 цифр) повторно выполняем над числом 312 операцию целочисленного деления на 10:

$$312 / 10 = 31.$$

Продолжаем обработку по этому же принципу:

$$31 \% 10 = 1 \text{ — вторая цифра.}$$

$$31 / 10 = 3 \text{ — первая цифра.}$$

После получения всех цифр числа отдельно собираем новое число: вторая цифра $\cdot 10^3$ + первая цифра $\cdot 10^2$ + четвертая цифра $\cdot 10^1$ + третья цифра $\cdot 10^0$ или после упрощения: вторая цифра $\cdot 1000$ + первая цифра $\cdot 100$ + четвертая цифра $\cdot 10$ + третья цифра.

Решение

1. Выводим строку в консоль с описанием названия программы (по желанию, данная строка кода может быть пропущена), создаем необходимое число переменных (в данной задаче

все они будут целочисленные) — одна переменная для числа пользователя, четыре — для хранения цифр числа, одна — вспомогательная для хранения оставшейся части числа, одна — для нового числа (результата).

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #4.1.2\n\n";
    int userNumber, d1, d2, d3, d4, newNumber, temp;
    return 0;
}
```

2. Выводим в консоль строку с приглашением ввести число и считываем его в переменную

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #4.1.2\n\n";
    int userNumber, d1, d2, d3, d4, newNumber, temp;
    cout << "Please, enter your 4-digit number:\n";
    cin >> userNumber;
    return 0;
}
```

3. Реализуем проверку введенного числа на 4 цифры, применяя к введенному числу операцию целочисленного деле-

ния числа на 1000. Для реализации процедуры сравнения будем использовать оператор условного ветвления `if-else if — else` и логические операторы сравнения `>` и `<`.

Вначале реализуем проверку случая «результат целочисленного деления на 1000 меньше 1»:

```
if (userNumber / 1000 < 1)
{
    cout << "Your number includes less than "
          "4 digits!\n";
}
```

Потом реализуем проверку случая «результат целочисленного деления на 1000 больше 9»:

```
else if (userNumber / 1000 > 9)
{
    cout << "Your number includes more than "
          "4 digits!\n";
}
```

Случай «пользователь ввел 4-значное число», это альтернатива двум первым случаям, представляющая собой блок `else`

```
else
{
    cout << "Your number includes exactly "
          "4 digits!\n";
}
```

Использование конструкции `else if` вместо еще одного `if` необходимо, чтобы исключить проверку каждого из условий, если одно из них уже выполнилось. Т. е. в случае, если число уже содержит менее 6 цифр, нет необходимости проверять, не содержит ли оно более 6 цифр.

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #4.1.2\n\n";
    int userNumber, d1, d2, d3, d4, newNumber, temp;
    cout << "Please, enter your 4-digit number:\n";
    cin >> userNumber;
    if (userNumber / 1000 < 1)
    {
        cout << "Your number includes less than "
              "4 digits!\n";
    }
    else if (userNumber / 1000 > 9)
    {
        cout << "Your number includes more than "
              "4 digits!\n";
    }
    else
    {
        cout << "Your number includes exactly "
              "4 digits!\n";
    }

    return 0;
}
```


4. Внутри блока `else` (соответствующего случаю корректного ввода) реализуем процесс выделения цифр из числа.

4.1. Применяем к числу операцию остатка от деления на 10, результат — последняя (четвертая) цифра:

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #4.1.2\n\n";
    int userNumber, d1, d2, d3, d4, newNumber, temp;
    cout << "Please, enter your 4-digit number:\n";
    cin >> userNumber;
    if (userNumber / 1000 < 1)
    {
        cout << "Your number includes less than "
              "4 digits!\n";
    }
    else if (userNumber / 1000 > 9)
    {
        cout << "Your number includes more than "
              "4 digits!\n";
    }
    else
    {
        cout << "Your number includes exactly "
              "4 digits!\n";
        d4 = userNumber % 10;
    }

    return 0;
}
```

4.2. Для получения оставшейся (для дальнейшей обработки) части числа (первых 3 цифр) выполняем операцию целочисленного деления на 10:

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #4.1.2\n\n";
    int userNumber, d1, d2, d3, d4, newNumber, temp;
    cout << "Please, enter your 4-digit number:\n";
    cin >> userNumber;
    if (userNumber / 1000 < 1)
    {
        cout << "Your number includes less than "
                "4 digits!\n";
    }
    else if (userNumber / 1000 > 9)
    {
        cout << "Your number includes more than "
                "4 digits!\n";
    }
    else
    {
        cout << "Your number includes exactly "
                "4 digits!\n";
        d4 = userNumber % 10;
        temp = userNumber / 10;

        return 0;
    }
}
```

Повторяем шаги 4.1 и 4.2 над оставшейся частью числа и получаем остальные цифры числа, выводим их в консоль.

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #4.1.2\n\n";
    int userNumber, d1, d2, d3, d4, newNumber, temp;
    cout << "Please, enter your 4-digit number:\n";
    cin >> userNumber;

    if (userNumber / 1000 < 1)
    {
        cout << "Your number includes less than "
              "4 digits!\n";
    }
    else if (userNumber / 1000 > 9)
    {
        cout << "Your number includes more than "
              "4 digits!\n";
    }
    else
    {
        cout << "Your number includes exactly "
              "4 digits!\n";
        d4 = userNumber % 10;
        temp = userNumber / 10;
        d3 = temp % 10;
        temp = temp / 10;
        d2 = temp % 10;
        d1 = temp / 10;
        cout << "Your number in digits: \n";
    }
}
```

```
        cout << d1 << " " << d2 << " " << d3 << " ";  
        cout << d4 << " Let's modify it. \n";  
    }  
  
    return 0;  
}
```

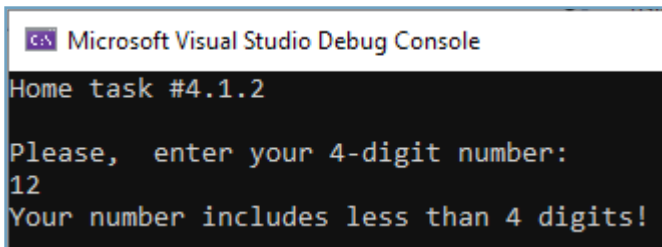
5. Собираем новое число по следующему принципу: **вторая цифра *1000+ первая цифра *100+ четвертая цифра *10+третья цифра** и выводим полученный результат в **консоль**

```
#include <iostream>  
using namespace std;  
  
int main()  
{  
    cout << "Home task #4.1.2\n\n";  
    int userNumber, d1, d2, d3, d4, newNumber, temp;  
    cout << "Please, enter your 4-digit number:\n";  
    cin >> userNumber;  
  
    if (userNumber / 1000 < 1)  
    {  
        cout << "Your number includes less than "  
                "4 digits!\n";  
    }  
    else if (userNumber / 1000 > 9)  
    {  
        cout << "Your number includes more than "  
                "4 digits!\n";  
    }  
}
```

```
else
{
    cout << "Your number includes exactly "
           "4 digits!\n";
    d4 = userNumber % 10;
    temp = userNumber / 10;
    d3 = temp % 10;
    temp = temp / 10;
    d2 = temp % 10;
    d1 = temp / 10;
    cout << "Your number in digits: \n";
    cout << d1 << " " << d2 << " " << d3 << " ";
    cout << d4 << " Let's modify it. \n";
    newNumber = d2 * 1000 + d1 * 100 + d4 *
                10 + d3;
    cout << "Your new number: " << newNumber;
}
return 0;
}
```

Результаты работы программы (в консоли):

- Тест 1 — пользователь ввел число, содержащее меньше, чем 4 цифры (ошибочная ситуация №1)

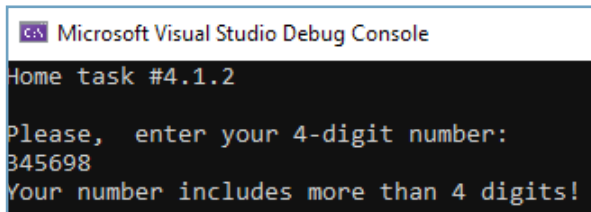


```
Microsoft Visual Studio Debug Console
Home task #4.1.2

Please, enter your 4-digit number:
12
Your number includes less than 4 digits!
```

Рисунок 5

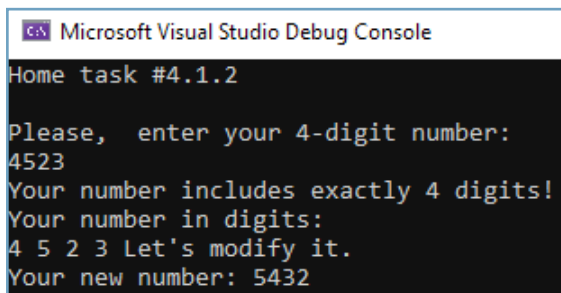
- Тест 2 — пользователь ввел число, содержащее больше, чем 4 цифры (ошибочная ситуация №2)



```
Microsoft Visual Studio Debug Console  
Home task #4.1.2  
Please, enter your 4-digit number:  
845698  
Your number includes more than 4 digits!
```

Рисунок 6

- Тест 3 — пользователь ввел число, содержащее 4 цифры



```
Microsoft Visual Studio Debug Console  
Home task #4.1.2  
Please, enter your 4-digit number:  
4523  
Your number includes exactly 4 digits!  
Your number in digits:  
4 5 2 3 Let's modify it.  
Your new number: 5432
```

Рисунок 7

РЕШЕНИЕ ЗАДАНИЯ 3**Описание решения**

После ввода информации пользователем (семи целых чисел по очереди, одно за другим) необходимо определить начальное значение максимума: если первое число больше второго, то назначить в качестве начального значения максимума первое число, иначе — второе.

Например, пользователь ввел такие числа: 4 5 6 3 7 2 1.

Начальное значение максимума будет 5 (т. к. первое число меньше второго)

На следующем шаге необходимо сравнить полученный максимум (число 5) и следующее число — 6. Число 6 больше текущего максимума, поэтому необходимо переопределить максимум (теперь текущий максимум — это число 6).

Далее сравним текущий максимум (число 6) и следующее число — 3. Число 3 меньше текущего максимума, поэтому значение максимума не изменяется (теперь текущий максимум по-прежнему число 6).

Повторяя данный подход для сравнения четвертого числа и максимума, пятого числа и максимума, шестого числа и максимума, седьмого числа и максимума, получим в результате максимум — число 7.

Решение

1. Выводим строку в консоль с описанием названия программы (по желанию, данная строка кода может быть пропущена), создаем необходимое число переменных (в данной задаче все они будут целочисленные) — семь

переменных для чисел пользователя, одна — для максимума (результата).

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #4.1.3\n\n";
    int d1, d2, d3, d4, d5, d6, d7, maxNumber;
    return 0;
}
```

2. Выводим в консоль строку с приглашением ввести числа и считываем их в переменные

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #4.1.3\n\n";
    int d1, d2, d3, d4, d5, d6, d7, maxNumber;
    cout << "Please, enter your 1st number:\n";
    cin >> d1;
    cout << "Please, enter your 2nd number:\n";
    cin >> d2;
    cout << "Please, enter your 3d number:\n";
    cin >> d3;
    cout << "Please, enter your 4th number:\n";
    cin >> d4;
    cout << "Please, enter your 5th number:\n";
    cin >> d5;
```



```
cout << "Please, enter your 6th number:\n";  
cin >> d6;  
cout << "Please, enter your 7th number:\n";  
cin >> d7;  
  
return 0;  
}
```

3. Определяем начальное значение максимума, сравнивая первое и второе числа. Если первое число больше второго, то начальное значение максимума — это первое число, иначе — второе. Используем оператор условного ветвления `if-else`.

```
#include <iostream>  
using namespace std;  
  
int main()  
{  
    cout << "Home task #4.1.3\n\n";  
    int d1, d2, d3, d4, d5, d6, d7, maxNumber;  
    cout << "Please, enter your 1st number:\n";  
    cin >> d1;  
    cout << "Please, enter your 2nd number:\n";  
    cin >> d2;  
    cout << "Please, enter your 3d number:\n";  
    cin >> d3;  
    cout << "Please, enter your 4th number:\n";  
    cin >> d4;  
    cout << "Please, enter your 5th number:\n";  
    cin >> d5;  
    cout << "Please, enter your 6th number:\n";  
    cin >> d6;  
    cout << "Please, enter your 7th number:\n";  
}
```

```
cin >> d7;  
if (d1 > d2)  
{  
    maxNumber = d1;  
}  
else  
{  
    maxNumber = d2;  
}  
return 0;  
}
```

4. Сравниваем следующее (третье) число с текущим максимумом, если оно больше текущего максимума, то перепределяем его, устанавливая значение из третьего числа. Альтернативный блок (с помощью **else**) реализовывать не нужно, т. к. в случае, когда третье число окажется меньше максимума, то ничего делать не нужно.

```
#include <iostream>  
using namespace std;  
  
int main()  
{  
    cout << "Home task #4.1.3\n\n";  
    int d1, d2, d3, d4, d5, d6, d7, maxNumber;  
    cout << "Please, enter your 1st number:\n";  
    cin >> d1;  
    cout << "Please, enter your 2nd number:\n";  
    cin >> d2;  
    cout << "Please, enter your 3d number:\n";  
    cin >> d3;
```

```
cout << "Please, enter your 4th number:\n";
cin >> d4;
cout << "Please, enter your 5th number:\n";
cin >> d5;
cout << "Please, enter your 6th number:\n";
cin >> d6;
cout << "Please, enter your 7th number:\n";
cin >> d7;
if (d1 > d2)
{
    maxNumber = d1;
}
else
{
    maxNumber = d2;
}
if (d3 > maxNumber)
{
    maxNumber = d3;
}
return 0;
}
```

5. Повторяя данный подход для сравнения четвертого числа и максимума, пятого числа и максимума, шестого числа и максимума, седьмого числа и максимума, получим в результате максимум из семи чисел. Выведем его в консоль.

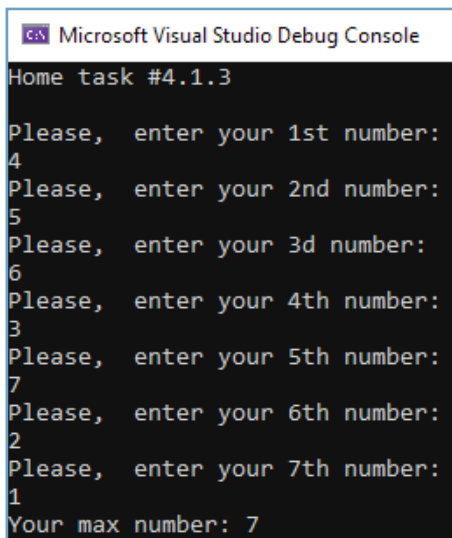
```
#include <iostream>
using namespace std;

int main()
{
```

```
cout << "Home task #4.1.3\n\n";
int d1, d2, d3, d4, d5, d6, d7, maxNumber;
cout << "Please, enter your 1st number:\n";
cin >> d1;
cout << "Please, enter your 2nd number:\n";
cin >> d2;
cout << "Please, enter your 3d number:\n";
cin >> d3;
cout << "Please, enter your 4th number:\n";
cin >> d4;
cout << "Please, enter your 5th number:\n";
cin >> d5;
cout << "Please, enter your 6th number:\n";
cin >> d6;
cout << "Please, enter your 7th number:\n";
cin >> d7;
if (d1 > d2)
{
    maxNumber = d1;
}
else
{
    maxNumber = d2;
}
if (d3 > maxNumber)
{
    maxNumber = d3;
}
if (d4 > maxNumber)
{
    maxNumber = d4;
}
if (d5 > maxNumber)
{
```

```
        maxNumber = d5;
    }
    if (d6 > maxNumber)
    {
        maxNumber = d6;
    }
    if (d7 > maxNumber)
    {
        maxNumber = d7;
    }
    cout << "Your max number: " << maxNumber;
    return 0;
}
```

Результаты работы программы (в консоли):



The screenshot shows the Microsoft Visual Studio Debug Console with the following text:

```
C:\> Microsoft Visual Studio Debug Console
Home task #4.1.3

Please, enter your 1st number:
4
Please, enter your 2nd number:
5
Please, enter your 3d number:
6
Please, enter your 4th number:
3
Please, enter your 5th number:
7
Please, enter your 6th number:
2
Please, enter your 7th number:
1
Your max number: 7
```

Рисунок 8