

Примеры решения задач на использование условных операторов

Пример 1.1. Дано двузначное число A . Если сумма его цифр кратна X и вторая цифра кратна X , уменьшить все его цифры в два раза; если сумма его цифр кратна X , а вторая цифра не кратна X , увеличить все его цифры в два раза (по модулю 10); если сумма его цифр не кратна X , а вторая цифра кратна X , увеличить все его цифры на 5 (по модулю 10); если сумма его цифр не кратна X и вторая цифра не кратна X , уменьшить все его цифры на 5 (по модулю). На экран вывести новое число, а не набор цифр.

По модулю 10 означает взять остаток от деления на 10.

Решение.

Поскольку надо определить цифры, то все переменные будут типа `int`.

Определим цифры числа A : первая — $b = A/10$, вторая — $c = A\%10$.

Так как при увеличении цифр они могут стать двузначными, необходимо взять остаток от деления на 10. При уменьшении цифр на какое-то число, они могут стать отрицательными, поэтому необходимо взять по модулю.

Алгоритм выглядит следующим образом:

1. Сначала необходимо проверить, являются ли введенные числа A и X , соответственно, двузначным и однозначным числом. Если нет, завершить программу.
2. Отделить цифры (b, c) , найти их сумму $S = b + c$.
3. Если $S\%X == 0$
 - а. Если $c\%X == 0$, все цифры уменьшаем в два раза.
 - б. Иначе, все цифры увеличиваем в два раза (по модулю 10).
4. Иначе (сумма не кратна X)
 - а. Если $c\%X == 0$, все цифры увеличиваем на 5 (по модулю 10).
 - б. Иначе, все цифры уменьшаем на 5 (по модулю).
5. Создаем новое число $Y = 10 * b_{new} + c_{new}$ и выводим его на экран.

Листинг 1.1.

```
1 //изменить цифры двузначного числа
2 #include <iostream>
3 #include <cmath>
4 using namespace std;
5
6 int main(){
7     setlocale (LC_ALL, "RUS");
8     int A, X;
9     cout << "Введите двузначное число\n";
10    cin >> A;
11    cout << "Введите однозначное число\n";
12    cin >> X;
13    //проверка на корректный ввод
14    if (A < 10 || A > 99 || X <= 0 || X > 9) cout << "Неправильно введены данные\n";
15    else{ //все хорошо
16        int b = A / 10;
17        int c = A % 10;
18        int S = b + c;
19        int bn, cn, Y;
20
21        if (S % X == 0){ //сумма кратна X
22            if (c % X == 0){ //вторая цифра кратна X
23                bn = b / 2;
24                cn = c / 2;
25            }
26            else { //первая цифра не кратна X
27                bn = b * 2 % 10;
28                cn = c * 2 % 10;
29            }
30        }
31        else{ //сумма не кратна X
32            if (c % X == 0){ //вторая цифра кратна X
33                bn = (b + 5) % 10;
34                cn = (c + 5) % 10;
35            }
36            else { //первая цифра не кратна X
37                bn = abs(b - 5);
38                cn = abs(c - 5);
```

```

39     }
40 }
41
42 Y = 10 * bn + cn; //создаем новое число
43 cout << Y << endl;
44 }
45 system("pause");
46 return 0;
47 }

```

Результат работы программы:

A	X	Результат
28	2	14
28	5	46
25	5	70
25	4	30
345	5	Неправильные данные
24	18	Неправильные данные

□

Пример 1.2. Вычислить значение функции $F(x) = \frac{\sqrt{x^2 - 16} + \ln(8 - x)}{\sqrt{x^2 + 10x + 25}}$.

В случае, если функция неопределена, вывести сообщение об этом (должно быть выведено: «Деление на ноль», «Корень из отрицательного числа», «Логарифм отрицательного числа»).

Решение.

Используемые переменные должны быть типа `float`.

Функция может быть неопределена в случае:

1. Корня из отрицательного числа: $x^2 - 16 = (x - 4)(x + 4)$. Чтобы получилось отрицательное число, одна из скобок должна быть отрицательной, другая положительной: $x - 4 > 0$ && $x + 4 < 0$ решения не имеет; $x - 4 < 0$ && $x + 4 > 0$ имеет решение: $-4 < x < 4$.

В случае корня в знаменателе $x^2 + 10x + 25 = (x + 5)^2$ — всегда положительное число.

2. Логарифма отрицательного числа: $8 - x \leq 0 \rightarrow x \geq 8$.
3. Деления на ноль: $x + 5 = 0 \rightarrow x = -5$.

Листинг 1.2.

```
1 //найти значение выражения
2 #include <iostream>
3 #include <cmath>
4 using namespace std;
5
6 int main(){
7     setlocale (LC_ALL, "RUS");
8     float X;
9     cout << "x = ";
10    cin >> X;
11    if (X > -4 && X < 4) cout << "Корень из отрицательного числа\n";
12    else if (X >= 8) cout << "Логарифм отрицательного числа\n";
13    else if (X == -5) cout << "Деление на ноль\n";
14    else {
15        float Y = (sqrt(X * X - 16) + log (8 - X))/(X + 5);
16        cout << "F = " << Y << endl;
17    }
18    system("pause");
19    return 0;
20 }
```

Результат работы программы:

X	Результат
5	0.409861
-2	Корень из отрицательного числа
10	Логарифм отрицательного числа
-5	Деление на ноль

□

Пример 1.3. С 1 января 2000 года прошло t месяцев. Вывести на экран название текущего месяца.

РЕШЕНИЕ.

Лучше воспользоваться оператором `switch`, поэтому переменная должна быть типа `int`.

Сначала надо найти остаток от деления на 12, чтобы определить текущий месяц. По результату выбирать. Если остаток равен 0, текущий месяц — январь, 11 — декабрь. `default` не нужен, так как результатом операции взятия остатка от деления может быть только диапазон $[0..11]$.

Листинг 1.3.

```

1 //найти название текущего месяца
2 #include <iostream>
3 #include <cmath>
4 using namespace std;
5
6 int main(){
7     setlocale (LC_ALL, "RUS");
8     int m;
9     cout << " m = ";
10    cin >> m;
11
12    m %= 12; //сколько прошло месяцев (отделяем кл-во прошедших лет)
13    switch (m){
14        case 0 : cout << "Январь\n"; break;
15        case 1 : cout << "Февраль\n"; break;
16        case 2 : cout << "Март\n"; break;
17        case 3 : cout << "Апрель\n"; break;
18        case 4 : cout << "Май\n"; break;
19        case 5 : cout << "Июнь\n"; break;
20        case 6 : cout << "Июль\n"; break;
21        case 7 : cout << "Август\n"; break;
22        case 8 : cout << "Сентябрь\n"; break;
23        case 9 : cout << "Октябрь\n"; break;
24        case 10 : cout << "Ноябрь\n"; break;
25        case 11 : cout << "Декабрь\n"; break;
26    }
27    system("pause");
28    return 0;
29 }

```

Результат работы программы:

m	Результат
45	Октябрь
8	сентябрь
60	Январь

□

Пример 1.4. Дана окружность с координатами центра $O(X, Y)$ и радиусом R и прямая $Ax + By + C = 0$. Найти их взаимное расположение: не пересекаются, пересекаются

(найти точки пересечения), касаются (найти точку пересечения).

РЕШЕНИЕ.

Уравнение окружности: $(x - X_c)^2 + (y - Y_c)^2 = R^2$. Следовательно, надо решить систему:

$$\begin{cases} (x - X_c)^2 + (y - Y_c)^2 = R^2 \\ Ax + By + C = 0. \end{cases}$$

Если корней нет, то прямая и окружность не пересекаются, если один корень — они касаются, если 2 корня — пересекаются.

$$\begin{cases} (\frac{By + C}{A} + X_c)^2 + (y - Y_c)^2 = R^2 \\ x = -\frac{By + C}{A}. \end{cases}$$

Если $A \neq 0$, то решаем первое уравнение. Раскрывая скобки, получаем:

$$(\frac{B^2}{A^2} + 1)y^2 + (\frac{2B(C + X_c)}{A} - 2Y_c)y + (\frac{(C + X_c)^2}{A} + Y_c^2 + R^2) = 0$$

Переобозначаем:

$$\begin{aligned} A_1 &= \frac{B^2}{A^2} + 1 \\ B_1 &= \frac{2B(C + X_c)}{A} - 2Y_c \\ C_1 &= \frac{(C + X_c)^2}{A} + Y_c^2 + R^2 \end{aligned}$$

В итоге получаем обычное квадратное уравнение $A_1x^2 + B_1x + C_1 = 0$ и находим его корни.

Если $A = 0$, система имеет следующий вид:

$$\begin{cases} (x - X_c)^2 + (y - Y_c)^2 = R^2 \\ By + C = 0. \end{cases}$$

Соответственно, если $B \neq 0$,

$$\begin{cases} (x - X_c)^2 + (-\frac{C}{B} - Y_c)^2 = R^2 \\ y = -\frac{C}{B}. \end{cases}$$

Тогда $x = X_c \pm \sqrt{R^2 - (\frac{C}{B} + Y_c)^2}$. Определяем знак корня.

Если $A = 0$ и $B = 0$, второе уравнение не является уравнением прямой.

Листинг 1.4.

```

1 //взаимное расположение окружности и прямой
2 #include <iostream>
3 #include <cmath>
4 using namespace std;
5
6 int main(){
7     setlocale (LC_ALL, "RUS");
8     float A,B,C, Xc, Yc, R;
9     cout << " Введите коэффциенты прямой\n";
10    cin >> A >> B >> C;
11    cout << " Введите координаты центра и радиус окружности\n";
12    cin >> Xc >> Yc >> R;
13
14    if (A != 0){ //  $Ax + By + C = 0$ 
15        float A1 = B * B / (A * A) + 1; //переобозначаем (см. пример)
16        float B1 = 2 * B * (C + Xc) / A - 2 * Yc;
17        float C1 = (C + Xc)*(C + Xc) / A + Yc * Yc + R * R;
18
19        float D = B1*B1 - 4 * A1 * C1; //  $A1x^2 + B1x + C1$ 
20
21        if (D < 0) cout << "Прямая и окружность не пересекаются\n";
22        else if (D == 0){
23            cout << "Прямая и окружность касаются\n";
24            float y = -B1 / (2 * A1); //из квадратного уравнения
25            float x = - (B*y + C) / A; // из системы (см. пример)
26            cout << "Координаты точки касания " << x << " " << y << endl;
27        }
28        else{
29            cout << "Прямая и окружность пересекаются\n";
30            float y1 = (-B1 - D) / (2 * A1); //из квадратного уравнения
31            float x1 = - (B*y1 + C) / A; // из системы (см. пример)
32            cout << "Координаты первой точки пересечения " << x1 << " " << y1 <<
                endl;
33            float y2 = (-B1 + D) / (2 * A1); //из квадратного уравнения
34            float x2 = - (B*y2 + C) / A; // из системы (см. пример)
35            cout << "Координаты первой точки пересечения " << x2 << " " << y2 <<
                endl;
36        }

```

```

37 }
38 else if (B != 0){ //By + C = 0
39     float D = R * R - (C / B + Yc)*(C / B + Yc); //из системы (см. пример)
40     if (D < 0) cout << "Прямая и окружность не пересекаются\n";
41     else if (D == 0){
42         cout << "Прямая и окружность касаются\n";
43         float y = -C / B; //из уравнения прямой
44         float x = Xc; // из системы (см. пример)
45         cout << "Координаты точки касания " << x << " " << y << endl;
46     }
47     else{
48         cout << "Прямая и окружность пересекаются\n";
49         float y1 = -C / B; //из уравнения прямой
50         float x1 = Xc - D; // из системы (см. пример)
51         cout << "Координаты первой точки пересечения " << x1 << " " << y1 <<
            endl;
52         float y2 = -C / B; //из уравнения прямой
53         float x2 = Xc + D; // из системы (см. пример)
54         cout << "Координаты первой точки пересечения " << x2 << " " << y2 <<
            endl;
55     }
56 }
57 else cout << "Это не прямая\n"; //C = 0
58 system("pause");
59 return 0;
60 }

```

□

Примеры решения задач на использование циклов

Пример 2.1. Дано целое число A . Вывести простое число, следующее за A .

РЕШЕНИЕ.

Для того, чтобы определить является ли число X простым, необходимо перебрать все числа от 2 до \sqrt{X} . Если не встретилось ни одного делителя, то число является простым.

В нашей задаче будет два вложенных цикла.

Внешний увеличивает на единицу заданное число, до тех пор пока не встретится простое. Условие выхода из цикла — найденное простое число. Лучше всего использовать цикл с постусловием, так как, введенное число может оказаться простым, следовательно, лучше сначала определить, является ли это число простым, а потом увеличивать.

Внутренний цикл определяет, является ли текущее число простым.

Листинг 2.1.

```
1 //Дано число. Найти следующее за ним простое число.
2 #include<iostream>
3 #include<cmath>
4 using namespace std;
5
6 int main(){
7     int A;
8     cout << " A = ";
9     cin >> A;
10    bool fl;
11    do{
12        fl = true; //предполагаем что число простое
13        for(int i = 2; i <= sqrt((float) A); i++) //проверяем на простоту
14            if (A % i == 0){
15                fl = false; //встретили делитель, число не простое
16                break; //прекращаем цикл
17            }
18        A++;
19    }
```

```

20     while (!fl);
21     cout << "--A << endl;
22     return 0;
23 }

```

□

Пример 2.2. Вывести на экран значения функции $y = \frac{\sqrt{x-1}}{\sqrt{x^2-4x+4}}$ в диапазоне $[a, b]$ с шагом h . В случае, если функция неопределена, вывести сообщение об этом.

РЕШЕНИЕ.

Так как задача математическая, переменные должны быть типа `float` или `double`.

При вводе данных необходимо проверить, что диапазон введен верно, в противном случае поменять местами значения переменных a и b (строка 17 Листинга 2.2).

В данном случае функция неопределена, если

1. Выражение под корнем в числителе отрицательно: $x - 1 < 0$.
2. Выражение под корнем в знаменателе равно нулю, так как это квадрат разности, он всегда положителен: $x^2 - 4x + 4 = (x - 2)^2 \rightarrow x = 2$.

Листинг 2.2.

```

1 //вычислить значения функции в диапазоне [a,b] с шагом h
2 #include<iostream>
3 #include<math.h>
4 using namespace std;
5
6 int main(){
7     setlocale(LC_ALL, "rus");
8     float a, b, h;
9     //ввод данных
10    cout << "a=";
11    cin >> a;
12    cout << "b=";
13    cin >> b;
14    cout << "h=";
15    cin >> h;
16    //проверка корректности или замена:

```

```
17  if (a > b) swap (a,b);
18  for (float x = a; x < b + h; x += h){//цикл [a,b]
19      if (x < 1) cout << "Корень из отрицательного числа\n";
20      else if (x == 2) cout << "Деление на ноль\n";
21      else{
22          float y = sqrt(x - 1)/sqrt(x*x - 4*x + 4);
23          cout << "x = " << x << " y = " << y << endl; //вывод
24      }
25  }
26  return 0;
27 }
```

Результат работы программы при $a = 0$, $b = 3$, $h = 0.5$:

x	Результат
0	Корень из отрицательного числа
0.5	Корень из отрицательного числа
1	0
1.5	1.41421
2	Деление на ноль
2.5	2.44949
3	1.41421

□

Пример 2.3. Напечатать следующую таблицу, используя вложенные циклы:

```
4
4
5 5
4 3
6 6 6
4 3 2
7 7 7 7
4 3 2 1
8 8 8 8 8
4 3 2 1 0
```

РЕШЕНИЕ.

Данная таблица состоит из 5 блоков, каждый блок содержит две строки. Внешний цикл изменяется от 1 до 5, внутренний — от 1 до i .

Нечетные строки:

```

4
5 5
6 6 6
7 7 7 7
8 8 8 8 8

```

печатаются по правилу: `cout << 3 + i << ""`;

Четные строки:

```

4
4 3
4 3 2
4 3 2 1
4 3 2 1 0

```

печатаются по правилу: `cout << 5 - j << ""`;

Листинг 2.3.

```

1 //напечатать таблицу
2 #include<iostream>
3 using namespace std;
4
5 int main(){
6     const int N = 5; //число блоков
7     for (int i = 1; i <= N; i++){//блоки
8         for (int j = 1; j <= i; j++) //нечетные строки
9             cout << 3 + i << " ";
10        cout << endl;
11        for (int j = 1; j <= i; j++) //четные строки
12            cout << 5 - j << " ";
13        cout << endl;
14    }
15    return 0;
16 }

```

□