

## Задача 1:

/\* Задание:

Три команды программистов вышли в финал международного чемпионата по спортивному программированию AMC ICPC - 2017. Каждая команда решила задачу в свое определенное время, победившей считается команда, решившая задачу раньше всех.

Вам нужно написать программу, которая найдет время, в которое команда победителей решила задачу, а также время отставания последней проигравшей команды от команды победителей.

\*/

```
#include "pch.h"
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(){
```

```
    int a, T, R, i;
```

```
    T = 100000;
```

```
    R = 0;
```

```
    for (i = 1; i < 4; i++){
```

```
        cin >> a;
```

```
        if (a < T)
```

```
            T = a;
```

```
        if (a > R)
```

```
            R = a;
```

```
    }
```

```
    cout << T << ' ' << R - T << endl;
```

```
}
```

The screenshot shows the Microsoft Visual Studio IDE with a C++ file named Task1.cpp. The code is as follows:

```
1  /* ... */
10 #include "pch.h"
11 #include <iostream>
12
13 using namespace std;
14
15 int main(){
16     int a, T, R, i;
17     T = 100000;
18     R = 0;
19     for (i = 1; i < 4; i++){
20         cin >> a;
21         if (a < T)
22             T = a;
23         if (a > R)
24             R = a;
25     }
26     cout << T << ' ' << R - T << endl;
27 }
```

The Debug Console window shows the following output:

```
55410
22445
12234
12234 43176
C:\Users\User\Desktop\Проекты\INF2018-1\Debug\INF2018-1.exe (процесс 7436) завер
шает работу с кодом 0.
Чтобы закрыть это окно, нажмите любую клавишу:
```

## Задача 2:

/\* Задание

Задано целое число N. В ячейку памяти вычислительной машины записывается его двоичное представление (размер ячейки - 1 байт). Над содержимым ячейки выполняется преобразование: циклический сдвиг влево на одну позицию. Все цифры двоичного числа сдвигаются влево на одну позицию, при этом старший бит переходит в младший. Преобразование повторяется восемь раз (после последнего преобразования число становится исходным).

Напишите программу, которая по введенному натуральному десятичному числу N, вычисляет наибольшее из десятичных чисел полученных в процессе преобразования.

Например: 201 представлено в ячейке памяти как 11001001. После первого сдвига влево получаем число 10010011, это соответствует десятичному числу 147.

\*/

```
#include "pch.h"
#include <iostream>
using namespace std;

int main(){
    int N, max, i, k;
    int a[8];
    int b[8];
    cin >> N;
    max = 0;
    for (i = 0; i <= 7; i++) {
        a[7-i] = N % 2;
        b[7-i] = a[7-i];
        N = (N - a[7-i]) / 2;
    }
    for (k = 1; k <= 8; k++) {
        N = 0;
        for (i = 1; i <= 6; i++)
            a[7 - i] = b[8 - i];
        a[0] = b[1];
        a[7] = b[0];
        for (i = 0; i <= 7; i++)
            N = N + a[7 - i] * pow(2, i);
        if (N > max)
            max = N;
        for (i = 0; i <= 7; i++) {
            b[i] = a[i];
        }
    }
    cout << max << endl;
}
```

The screenshot shows the Microsoft Visual Studio IDE with the code from the previous block. The debug console window is open, displaying the output of the program. The output consists of two lines: 86 and 202. Below the output, the console shows the file path and process ID: C:\Users\User\Desktop\Проекты\INF2018-2\Debug\INF2018-2.exe (процесс 10112). A message at the bottom of the console says: завершает работу с кодом 0. Чтобы закрыть это окно, нажмите любую клавишу.

### Задача 3:

/\* Задание

Группа археологов занималась раскопками древнего города. В результате раскопок поверхность стала неровной. Требуется определить, сколько ям образовалось на участке. Ямой считается место, уровень которого ниже граничащих с четырех сторон участков (слева, справа, сверху, снизу).

Участок представляет собой прямоугольник размерами N\*M.

Напишите программу, которая посчитает количество ям на участке.

\*/

```
#include "pch.h"
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(){
```

```
    int N, M, i, k, Result;
```

```
    int a[10][10];
```

```
    cin >> N >> M;
```

```
    Result = 0;
```

```
    for (i = 0; i < N; i++) {
```

```
        for (k = 0; k < M; k++) {
```

```
            cin >> a[k][i];
```

```
        }
```

```
    }
```

```
    for (i = 1; i < N - 1; i++) {
```

```
        for (k = 1; k < M - 1; k++) {
```

```
            if ((a[k][i] < a[k-1][i]) and (a[k][i] < a[k+1][i]) and (a[k][i] < a[k][i-1]) and (a[k][i] < a[k][i+1])) {
```

```
                Result = Result + 1;
```

```
            }
```

```
        }
```

```
    }
```

```
    cout << Result << endl;
```

```
}
```

The screenshot shows the Microsoft Visual Studio IDE with a C++ project named 'INF2018-3'. The code editor displays the solution for the 'Task 3' problem, which involves counting pits in a 2D grid. The code reads the grid dimensions N and M, then the grid elements. It iterates through the grid, checking each cell (excluding the first and last rows and columns) to see if it is a pit (lower than all four neighbors). The result is printed at the end.

The debug console window shows the following output:

```
10 10
56 1000 249 379 126 565 171 78 687 715
643 516 579 299 163 202 315 448 309 843
373 603 988 322 864 653 620 160 208 80
583 752 594 811 806 609 315 649 301 958
26 697 276 583 121 210 412 308 117 408
795 5 746 210 262 20 433 574 542 479
346 374 444 365 626 405 215 199 692 577
780 704 531 984 226 620 551 598 257 289
29 497 940 329 78 328 233 906 4 261
350 381 653 976 144 569 477 294 258 804
13
C:\Users\User\Desktop\Проекты\Debug\INF2018-3.exe (процесс 9348) завершает работу с кодом 0.
Чтобы закрыть это окно, нажмите любую клавишу:
```

## Задача 5:

/\* Задание

В комнате требуется покрыть пол листами ламината. Пол имеет прямоугольную форму, размерами  $M \times N$ . Доска ламината тоже имеет прямоугольную форму, размерами  $a \times b$ . Ламинат укладывается вдоль длинной стороны комнаты. Требуется определить количество досок ламината для покрытия пола.

Указание. При укладке ламината на пол, не всегда вмещается целое число досок, таким образом будет необходимо использовать еще часть доски ламината. Оставшуюся часть доски мы будем использовать только в том случае, если остаток больше либо равен половине доски, если же осталось меньше половины листа ламината, то она выбрасывается.

\*/

```
#include "pch.h"
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {  
    int M, N, a, b, k, l, n, m, r1, r2, R;  
    cin >> M >> N >> a >> b;  
    k = M / a;  
    l = N / b;  
    n = 0;  
    m = 0;  
    r1 = 0;  
    r2 = 0;  
    if ((M - a * k) != 0) {  
        while ((n * (M - a * k)) <= (a / 2))  
            n++;  
        r2 = l / n;  
        if ((l - r2 * n) != 0)  
            r2 = r2 + 1;  
    }  
    if ((N - b * l) != 0) {  
        while ((m * (N - b * l)) <= (b / 2))  
            m++;  
        r1 = k / m;  
        if ((k - r1 * m) != 0)  
            r1 = r1 + 1;  
    }  
    R = k * l + r1 + r2;  
    if (((M - a * k) != 0) and ((N - b * l) != 0))  
        R = R + 1;  
    cout << R << endl;  
}
```

The screenshot shows the Microsoft Visual Studio IDE with the C++ code from the previous block. The code is displayed in the main editor window. A separate window titled "Консоль отладки Microsoft Visual Studio" (Microsoft Visual Studio Debug Console) is open, showing the output of the program. The output consists of the numbers 30, 20, 7, 2, and 45, each on a new line. Below the numbers, a message indicates that the program has finished execution and prompts the user to press any key to close the window.