Инженерное дело - программирование, заключительный этап, 11 класс

Задача 1 (5 баллов)

Условие

Целые числа называются взаимно простыми, если не имеют никаких общих делителей, кроме единицы.
Для двух заданных натуральных чисел требуется проверить, являются ли они взаимно простыми.
Входные данные: через пробел записаны два натуральных числа, каждое не превышает 10^6.
Выходные данные: слово YES (заглавными буквами), если числа взаимно просты, или наименьший общий делитель двух заданных чисел (за исключением единицы) в противном случае.

Пример

Входные данные	Выходные данные
5 14	YES
6 15	3

Проверочные тесты

Входные данные	Ожидаемый результат
5 14	YES
6 15	3
7 7	7
15 25	5
15 27	3
30 75	3

Пример решения

```
#include <iostream>

using namespace std;

int main()
{
	int a, b;
	cin >> a >> b;
	for (int i=2; i<=a && i<=b; i++)
	{
	if (a%i == 0 && b%i == 0)
	{
		cout << i;
		return 0;
	}
	}
	cout << "YES";
	return 0;
}
```

Задача 2 (8 баллов)

Условие

Для заданного натурального числа и диапазона систем счисления найти такую с/с, в которой в записи данного числа можно получить наибольшую среди всех остальных записей цифру.
Входные данные: через пробел записаны 3 натуральных числа: N - заданное число в 10-й с/с, не превышающее 10^6, C_1 и C_2 (каждое в диапазоне от 2 до 16) - начало и конец диапазона систем счисления, который требуется исследовать. C_2 не может быть меньше C_1.
Выходные данные: основание системы счисления, в которой заданное число имеет наибольшую цифру в своей записи.
Если подходящих систем счисления несколько - вывести ту, чьё основание наибольшее.

Пример

Входные данные	Выходные данные	Примечание
51 2 16	13	$51_{10} = 3C_{13}$, C - наибольшая цифра среди записей числа в заданном диапазоне с/с
16 11 16	11	$16_{10} = 15_{11}$, 5 - наибольшая цифра среди записей числа в заданном диапазоне с/с

Проверочные тесты

Входные данные	Ожидаемый результат
51 2 16	13
16 11 16	11
1 2 16	16
65535 2 16	16
10 13 13	13
47250 2 16	15
47250 2 14	13
265720 2 3	3

Пример решения

```
#include <iostream>

using namespace std;

int main()
{
	int n, c1, c2;
	int m=0, mc=2;
	cin >> n >> c1 >> c2;
	for (int i=c1; i<=c2; i++)
	{
	int a = n;
	while (a>0)
	{
		if(a%i >= m)
		{
			m = a%i;
			mc=i;
		}
		a/=i;
	}
	}
	cout << mc;
	return 0;
}
```

Задача 3 (10 баллов)

Условие

Для двух заданных слов требуется определить, можно ли составить второе слово из букв первого.
Входные данные: в одну строку через пробел записаны два слова, состоящие из строчных латинских букв. Длина каждого слова не превышает 20 символов.
Выходные данные: 0, если второе слово можно составить из букв первого, либо количество букв, которых не хватает в первом слове для получения второго.

Пример

Входные данные	Выходные данные
abab abba	0
abcac abba	1

Проверочные тесты

Входные данные	Ожидаемый результат
abab abba	0
abcac abba	1
aaa a	0
a aaa	2
abccba aabbccaabbcc	6
yesterday tomorrow	6
aabbccdd abcabc	0
opqrst tostrsqqpo	4

Пример решения

```
def f(x,y):
      k = 0
      for i in range(len(y)):
      if y[i] in x:
            x = x.replace(y[i],'1',1)
            k += 1
      if k==len(y):
      return 0
      else:
      return (len(y)-k)

a = input()
a = a.split(' ')
a1 = a[0]
a2 = a[1]
print(f(a1,a2))
```

Задача 4 (12 баллов)

Условие

Требуется выполнить обход элементов квадратной матрицы нечётного размера по следующему принципу: во вложенном квадрате максимального размера, повёрнутом на 45 градусов относительно основной матрицы, выбрать все элементы, считывая их по часовой стрелке от внешнего контура к середине. В каждом контуре выбор элементов следует начинать с самого верхнего.

Входные данные: в первой строке записано натуральное нечётное число N, не превышающее 19. В последующих N строках записаны строки матрицы, значения в которых (каждое - целое число, по модулю не превышающее 1000) разделены пробелами.

Выходные данные: вывести в одну строку элементы матрицы по заданному правилу.

Пример

Входные данные	Выходные данные
5 11 12 **13** 14 15 21 **22 23 24** 25 **31 32 33 34 35** 41 **42 43 44** 45 51 52 **53** 54 55	13 24 35 44 53 42 31 22 23 34 43 32 33

Проверочные тесты

Входные данные	Ожидаемый результат
5 11 12 13 14 15 21 22 23 24 25 31 32 33 34 35 41 42 43 44 45 51 52 53 54 55	13 24 35 44 53 42 31 22 23 34 43 32 33
3 1 2 3 4 5 6 7 8 9	2 6 8 4 5
1 1	1
3 -1 -2 -3 -4 -5 -6 -7 -8 -9	-2 -6 -8 -4 -5
3 1000 -1000 1000 -1000 1000 -1000 1000 -1000 1000	-1000 -1000 -1000 -1000 1000
9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9	5 6 7 8 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 7 6 5 4 3 2 3 4 5 6 7 6 5 4 3 4 5 6 5 4 5

15 1 0 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 1
19 999 998 997 996 995 994 993 992 991 990 989 988 987 986 985 984 983 982 981 980 979 978 977 976 975 974 973 972 971 970 969 968 967 966 965 964 963 962 961 960 959 958 957 956 955 954 953 952 951 950 949 948 947 946 945 944 943 942 941 940 939 938 937 936 935 934 933 932 931 930 929 928 927 926 925 924 923 922 921 920 919 918 917 916 915 914 913 912 911 910 909 908 907 906 905 904 903 902 901 900 899 898 897 896 895 894 893 892 891 890 889 888 887 886 885 884 883 882 881 880 879 878 877 876 875 874 873 872 871 870 869 868 867 866 865 864 863 862 861 860 859 858 857 856 855 854 853 852 851 850 849 848 847 846 845 844 843 842 841 840 839 838 837 836 835 834 833 832 831 830 829 828 827 826 825 824 823 822 821 820 819 818 817 816 815 814 813 812 811 810 809 808 807 806 805 804 803 802 801 800 799 798 797 796 795 794 793 792 791 790 789 788 787 786 785 784 783 782 781 780 779 778 777 776 775 774 773 772 771 770 769 768 767 766 765 764 763 762 761 760 759 758 757 756 755 754 753 752 751 750 749 748 747 746 745 744 743 742 741 740 739 738 737 736 735 734 733 732 731 730 729 728 727 726 725 724 723 722 721 720 719 718 717 716 715 714 713 712 711 710 709 708 707 706 705 704 703 702 701 700 699 698 697 696 695 694 693 692 691 690 689 688 687 686 685 684 683 682 681 680 679 678 677 676 675 674 673 672 671 670 669 668 667 666 665 664 663 662 661 660 659 658 657 656 655 654 653 652 651 650 649 648 647 646 645 644 643 642 641 640 639	990 970 950 930 910 890 870 850 830 810 792 774 756 738 720 702 684 666 648 668 688 708 728 748 768 788 808 828 846 864 882 900 918 936 954 972 971 951 931 911 891 871 851 831 811 793 775 757 739 721 703 685 667 687 707 727 747 767 787 807 827 845 863 881 899 917 935 953 952 932 912 892 872 852 832 812 794 776 758 740 722 704 686 706 726 746 766 786 806 826 844 862 880 898 916 934 933 913 893 873 853 833 813 795 777 759 741 723 705 725 745 765 785 805 825 843 861 879 897 915 914 894 874 854 834 814 796 778 760 742 724 744 764 784 804 824 842 860 878 896 895 875 855 835 815 797 779 761 743 763 783 803 823 841 859 877 876 856 836 816 798 780 762 782 802 822 840 858 857 837 817 799 781 801 821 839 838 818 800 820 819

Пример решения

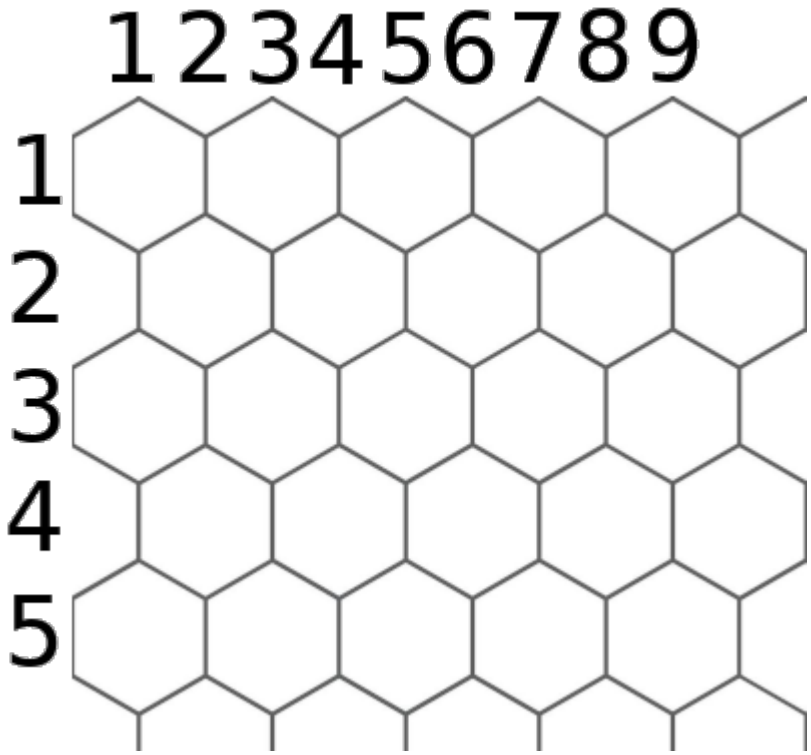
```
#include <iostream>

using namespace std;
int a[1000][1000];
int n;
int f(int p)
{

      int i,j;
      for (i=p,j=n/2; j<=n-1-p; i++, j++)
      {
      cout << a[i][j] << " ";
      }
      for (i=n/2+1, j=n-2-p; i<n-p; i++, j--)
      {
      // cout << "trrty";
      cout << a[i][j] << " ";
      }
      for (i=n-p-2, j=n/2-1; j>=p; i--, j--)
      {
      cout << a[i][j] << " ";
      }
      for (i=n/2-1, j=p+1; i>p; i--, j++)
      {
      cout << a[i][j] << " ";
      }
}
int main()
{
      cin >> n;
      for (int i=0; i<n; i++)
      for (int j=0; j<n; j++) cin >> a[i][j];

      for (int i=0; i<n/2; i++)
      {
      f(i);
      }
      cout << a[n/2][n/2];
      return 0;
}
```

Задача 5 (15 баллов)

Условие

Вася услышал от преподавателя кружка робототехники, в котором он занимается, что скоро состоятся городские соревнования. Одно из заданий, которое предстоит выполнить его команде - собрать робота, который сможет найти кратчайший путь из одной ячейки в другую на поле, являющемся шестиугольной решёткой.

Пример решётки изображён ниже:



Ячейки решётки нумеруются так, как изображено на схеме, по строкам и столбцам.

Робот из ячейки, в которой он расположен в данный момент, может перемещаться в одну из соседних шести ячеек.

Известно, что на поле могут находиться препятствия - ячейки, в которые робот попадать не может.

Помогите Васе составить алгоритм определения кратчайшего пути из одной заданной ячейки в другую и определить длину этого пути.

Входные данные: в первой строке записан размер поля - количество строк и столбцов решётки через пробел. Оба числа лежат в диапазоне от 1 до 10. Во второй строке записано неотрицательное целое K, не превышающее 20 - количество препятствий. Далее в K строках через пробел записаны координаты препятствий - номера строки и столбца каждой

"недостижимой" ячейки. Затем в последней строке через пробел записаны 4 числа - номера строки и столбца исходной и конечной позиции робота.

Выходные данные: одно число - длина кратчайшего пути, который роботу предстоит проделать, либо слово NO (заглавными буквами), если такой путь отсутствует.

Пример

Входные данные	Выходные данные
5 5 0 1 3 3 5	2
5 6 1 2 4 2 2 2 6	3

Проверочные тесты

Входные данные	Ожидаемый результат
5 5 0 1 3 3 5	2
5 6 1 2 4 2 2 2 6	3
1 1 0 1 1 1 1	0
4 4 2 2 2 2 4 4 2 1 1	NO

2 8 2 1 5 2 6 2 2 1 7	NO
10 10 5 3 1 3 3 3 5 2 6 1 5 2 4 6 10	NO
10 10 2 3 5 2 6 2 4 6 10	6
10 10 3 3 5 2 6 3 3 2 4 6 10	7
9 9 1 2 2 1 1 5 1	5
9 9 4 1 3 2 4 2 6 1 7 1 1 1 5	NO

Пример решения

```
#include <bits/stdc++.h>

typedef long long ll;
typedef long double ld;

using namespace std;

ll n, m;
```

```
set <pair <ll, ll>> bad;
map <pair <ll, ll>, ll> dist;

void solution() {
	ll k; cin >> n >> m >> k;
	for (ll i = 0; i < k; i++) {
	ll x, y; cin >> x >> y;
	bad.insert({x, y});
	}
	ll a, b, c, d;
	cin >> a >> b >> c >> d;
	dist[{a, b}] = 0;
	deque <pair <ll, ll>> q = {{a, b}};
	while (!q.empty()) {
	pair <ll, ll> t = q.front();
	ll x = t.first, y = t.second;
	q.pop_front();
	vector <pair <ll, ll>> vec = {{x, y - 2}, {x, y + 2}, {x - 1, y - 1},
{x - 1, y + 1}, {x + 1, y - 1}, {x + 1, y + 1}};
	for (pair <ll, ll> p : vec) {
		ll i = p.first, j = p.second;
		if (1 <= i && i <= n && 1 <= j && j <= m) {
			if (bad.find({i, j}) == bad.end() && dist.find({i, j}) ==
dist.end()) {
				dist[{i, j}] = dist[{x, y}] + 1;
				q.push_back({i, j});
			}
		}
	}
	}
	if (dist.find({c, d}) != dist.end())
	cout << dist[{c, d}];
	else
	cout << "NO";
}


int main() {
	ll T = 1;
	// cin >> T;
	while (T--) solution();
	return 0;
}
```