# Задачи

Линейные вычисления

1. Вы положили на счет в банке *X* руб. Процентная ставка *Z*% годовых. Определить, сколько денег будет на счету через *Y* лет.

Обратите внимание, в первый год проценты начисляются на сумму вашего вклада, в последующие года проценты начисляются на сумму вклада плюс проценты за прошлые года.

Пример

X = 1000 руб., У = 2 года, Z = 5%

1 год: 1000 + 1000 \* 5% = 1050 руб.

2 год: 1050 + 1050 \* 5% = **1102,5 руб.**

Решение

#include<iostream>

#include<cmath>

#include<Windows.h> //для вывода русского языка в консоли

using namespace std;

int main()

{

 SetConsoleCP(65001); //для вывода русского языка в консоли

 SetConsoleOutputCP(65001); //для вывода русского языка в консоли

 float X,Z;

 int Y;

 cout << "Сумма вклада: "; cin >> X;

 cout << "Срок вклада: "; cin >> Y;

 cout << "Процентная ставка: "; cin >> Z;

 Z = 1+Z/100;

 X = X\*pow(Z,Y);

 cout << "У вас на счету: " << X << " руб." << endl;

 return 0;

}

Условия

1. Вы каждый день ездите на метро. Утром вы едете в школу, а вечером того же дня, обратно из школы, домой. Для того, чтобы немного сэкономить, вы покупаете электронную смарт-карту на *X* поездок. Если на карте осталось ненулевое количество поездок, то турникет пропускает вас и списывает с карты одну поездку. Если же на карте не осталось поездок, то турникет не пропускает вас, и вам нужно купить на этой же станции новую карту на *X* поездок и вновь пройти через турникет.

В связи с тем, что утром метро переполнено, покупать новую карту утром накладно по времени, и можно опоздать в школу. Вам хочется понять: будет ли такой день, что с утра, поехав в школу, вы обнаружите у себя на карточке ноль поездок. Вы больше никуда на метро не ездите и поэтому заходите в метро только на станции около дома и на станции около школы.

Ввод

Слово «School» или «Home» в зависимости от того, где первый куплена карточка на X поездок. Во второй строке содержится натуральное число X, 1 ≤ X ≤ 1000.

Вывод

«Yes», если будет такой день, что дома утром на карточке окажется ноль поездок и «No» в противном случае

Пример

|  |  |
| --- | --- |
| Ввод | Вывод |
| School5 | **Yes** |

Решение

#include<iostream>

#include<Windows.h>

using namespace std;

int main()

{

 SetConsoleCP(65001);

 SetConsoleOutputCP(65001);

 string place;

 int count;

 cin >> place;

 cin >> count;

 if (place == "School" && count % 2 == 1 ||

 place == "Home" && count % 2 == 0)

 cout << "Yes\n";

 else

 cout << "No\n";

 return 0;

}

1. Даны длины трех отрезков. Требуется проверить: могут ли они являться сторонами невырожденного треугольника.

Пример

|  |  |
| --- | --- |
| Ввод | Вывод |
| 3 4 5 | **Yes** |
| 6 6 30 | **No** |

Решение

#include<iostream>

#include<Windows.h>

using namespace std;

int main()

{

 SetConsoleCP(65001);

 SetConsoleOutputCP(65001);

 int A,B,C;

 cin >> A >> B >> C;

 if (A < B+C && B < A+C && C < A+B)

 cout << "Yes\n";

 else

 cout << "No\n";

 return 0;

}

Циклы

1. Вывести на экран таблицу степеней числа *X* от *Y* до *Z* степени включительно.

Пример

X = 3

Y = 2

Z = 5

**3^2 = 9**

**3^3 = 27**

**3^4 = 81**

**3^5 = 243**

Решение

#include<iostream>

#include<cmath>

#include<Windows.h>

using namespace std;

int main()

{

 SetConsoleCP(65001);

 SetConsoleOutputCP(65001);

 int X,Y,Z;

 cout << "Степени какого числа: "; cin >> X;

 cout << "От: "; cin >> Y;

 cout << "До: "; cin >> Z;

 for (int i=Y; i<=Z; i++)

 cout << X << " ^ " << i << " = " << pow(X,i) << endl;

 return 0;

}

1. Последовательность чисел *a1*, *a2*, …, *ai*,… называется Фибоначчиевой, если для всех *i* ≥ 3 верно, что *ai*=*ai-1*+*ai-2*, то есть каждый член последовательности (начиная с третьего) равен сумме двух предыдущих.

Ясно, что, задавая различные числа *a1* и *a2*, мы можем получать различные такие последовательности, и любая Фибоначчиева последовательность однозначно задается двумя своими первыми членами.

Будем решать обратную задачу. Вам будет дано число n и два члена последовательности: *an* и *an+1*. Вам нужно написать программу, которая по их значениям найдет *a1* и *a2*.

Пример

n = 4

an = 3

an+1 = 5

**a1 = 1**

**a2 = 1**

Решение

#include<iostream>

#include<Windows.h>

using namespace std;

int main()

{

 SetConsoleCP(65001);

 SetConsoleOutputCP(65001);

 int a\_n1, a\_n, n;

 cout << "n = "; cin >> n;

 cout << "a[n] = "; cin >> a\_n;

 cout << "a[n+1] = "; cin >> a\_n1;

 for (int i=n; i>=2; i--){

 int c = a\_n1 - a\_n;

 a\_n1 = a\_n;

 a\_n = c;

 }

 cout << "a[1] = " << a\_n << endl;

 cout << "a[2] = " << a\_n1 << endl;

 return 0;

}

Массивы

1. Вывести элементы числового массива, которые больше, чем элементы, стоящие перед ними. Вы должны задать размерность массива, сам массив заполняется случайными числами.

Пример

Дан массив [3, 9, 8, 4, 5, 1]. Следует вывести числа **9** и **5**, так как перед ними стоят соответственно числа 3 и 4, которые меньше их.

Решение

#include<iostream>

#include<cstdlib>

#include<Windows.h>

#include<ctime>

using namespace std;

int main()

{

 SetConsoleCP(65001);

 SetConsoleOutputCP(65001);

 int N;

 cout << "Введите размер массива: "; cin >> N;

 int \*mas = new int[N];

 srand(time(NULL));

 cout << "Наш массив: ";

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

 mas[i] = rand() % 10 + 1;

 cout << mas[i] << " ";

 }

 cout << endl << endl;

 cout << "Элементы, удовлетворяющие условию: ";

 for (int i=1; i<N; i++)

 if (mas[i] > mas[i-1])

 cout << mas[i] << " ";

 cout << endl;

 delete [] mas;

 return 0;

}

1. На основе одномерного массива создать новый массив, содержащий только уникальные элементы.

Пример

Дан массив [3, 1, 3, 5, 1, 7, 5]

Получим **[3,1,5,7]**

Решение

#include<iostream>

#include<cstdlib>

#include<Windows.h>

#include<ctime>

using namespace std;

int main()

{

 SetConsoleCP(65001);

 SetConsoleOutputCP(65001);

 int N;

 cout << "Введите размер массива: "; cin >> N;

 int \*mas = new int[N];

 srand(time(NULL));

 cout << "Наш массив: ";

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

 mas[i] = rand() % 10 + 1;

 cout << mas[i] << " ";

 }

 cout << endl;

 int uniq = 0;

 int \*umas = new int [N];

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

 bool flag = true;

 for (int j=0; j<N; j++)

 if (i!=j && mas[i]==mas[j]){

 flag = false;

 break;

 }

 if (flag) {

 umas[uniq] = mas[i];

 uniq++;

 }

 }

 cout << "Уникальныe: ";

 for (int i=0; i<uniq; i++)

 cout << umas[i] << " ";

 cout << endl;

 delete [] mas;

 delete [] umas;

 return 0;

}

Подпрограммы

1. Написать функцию, вычисляющую площадь треугольника по трем его сторонам.

#include<iostream>

#include<Windows.h>

#include<cmath>

using namespace std;

float Square(float A, float B, float C){

 float p = 0.5\*(A+B+C);

 float S = sqrt(p\*(p-A)\*(p-B)\*(p-C));

 return S;

}

int main()

{

 SetConsoleCP(65001);

 SetConsoleOutputCP(65001);

 float A,B,C;

 cout << "A: "; cin >> A;

 cout << "B: "; cin >> B;

 cout << "C: "; cin >> C;

 cout << "S = " << Square(A,B,C) << endl;

 return 0;

}

1. Написать функцию, вычисляющую наименьшее общее кратное двух чисел.

#include<iostream>

#include<Windows.h>

#include<cmath>

using namespace std;

int SCM(int X, int Y){

 for (int i=max(X,Y); i<=X\*Y; i+=max(X,Y))

 if (i % X == 0 && i % Y == 0)

 return i;

}

int main()

{

 SetConsoleCP(65001);

 SetConsoleOutputCP(65001);

 int X,Y;

 cout << "First X: "; cin >> X;

 cout << "Second Y: "; cin >> Y;

 cout << "SCM(" << X << ", " << Y << ") = " << SCM(X,Y) << endl;

 return 0;

}

1. Написать процедуру, распечатывающую подмассив массива. В качестве параметра передается, с какого индекса начать вывод и сколько элементов вывести.

#include<iostream>

#include<Windows.h>

#include<cstdlib>

#include<ctime>

using namespace std;

void Print(int\* mas, int from, int N){

 for (int i=from; i < from+N; i++)

 cout << mas[i] << " ";

 cout << endl;

}

int main()

{

 SetConsoleCP(65001);

 SetConsoleOutputCP(65001);

 int mas[30];

 srand(time(NULL));

 for (int i=0; i<30; i++)

 mas[i] = rand() % 10 + 1;

 cout << "С 0-го 30 штук (весь): "; Print(mas,0,30);

 cout << "С 5-го 10 штук: "; Print(mas,5,10);

 return 0;

}

Структура

1. Создать структуру квадратное уравнение. Поля: коэффициенты *A*, *B*, *C*. Написать процедуры для заполнения структуры, для ее вывода на экран (в виде *A*x^2+*B*x+*C* = 0), для решения уравнения. Создать массив из 5 квадратных уравнений, заполнить их. Вывести на экран только квадратные уравнения с дискриминантом, не меньшим 0. Решить их.

Пример

1е уравнение

A = 4

B = 1

C = 1

2е уравнение

A = 1

B = -2

C = 1

3е уравнение

A = 1

B = -4

C = 4

4е уравнение

A = 1

B = 5

C = 4

5е уравнение

A = 2

B = 6

C = 10

**Вывод**

**x^2-2x+1 = 0 4-4 = 0**

**x1 = 1, x2 = 1**

**x^2-4x+4 = 0**

**x1 = 2, x2 = 2**

**x^2+5x+4 = 0**

**x1 = -4, x2 = -1**

#include<iostream>

#include<Windows.h>

#include<cmath>

using namespace std;

struct SquareEq{

 int A;

 int B;

 int C;

};

void Print(SquareEq e){

 cout << e.A << "\*x^2 ";

 if (e.B < 0)

 cout << e.B << "\*x ";

 else

 cout << "+ "<< e.B << "\*x ";

 if (e.C < 0)

 cout << e.C << " = 0\n";

 else

 cout << "+ "<< e.C << " = 0\n";

}

void Input(SquareEq& e){

 cout << "A = "; cin >> e.A;

 cout << "B = "; cin >> e.B;

 cout << "C = "; cin >> e.C;

}

float D(SquareEq e){

 return e.B\*e.B - 4\*e.A\*e.C;

}

int main()

{

 SetConsoleCP(65001);

 SetConsoleOutputCP(65001);

 SquareEq mas[5];

 for (int i=0; i<5; i++){

 cout << i+1 << "е уравнение:\n";

 Input(mas[i]);

 }

 for(int i=0; i<5; i++)

 if (D(mas[i]) >= 0){

 float x1 = (-mas[i].B - sqrt(D(mas[i]))) / (2\*mas[i].A);

 float x2 = (-mas[i].B + sqrt(D(mas[i]))) / (2\*mas[i].A);

 Print(mas[i]);

 cout << "x1 = " << x1 << " x2 = " << x2 << endl << endl;

 }

 return 0;

}