Rozwiązywanie układu równań

#include <iostream>

using namespace std;

int a,b,c,d,x,y;

int main()

{

cin >> d;

cin >> c;

y=(d-c)/2;

x=d-y;

0 <= a-b <= a+b <= 1000000000;

if (x>=y)

{

a=x;

b=y;

cout << a<<" "<<b;

}

else

{

a=y;

b=x;

cout << a<< " "<<b;

}

return 0;

}

Obliczanie silni

#include <iostream>

#include <string>

using namespace std;

int factorial(int);

int main()

{

int n=0;

cin >>n;

cout << factorial(n);

return 0;

}

int factorial (int a)

{

int w=1;

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

{

w \*=i;

}

return w;

}

Niech dwumian(n,k) oznacza n po k (lub n nad k - zależy jak kto słyszał) zachodzi zależność: dla k=0 lub k=n wynosi 1; dla 0 < k < n jest dwumian(n,k)=dwumian(n-1,k)+dwumian(n-1,k-1); w pozostałych przypadkach wynosi 0. dla każdej z t par liczb n i k oblicz n po k.

#include <iostream>

using namespace std;

long long factorial(int);

int main()

{

int n;

int k;

int number;

cin>>number;

long long array[number];

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

{

cin>>n;

cin>>k;

array[i]= factorial(n)/(factorial(k)\*factorial(n-k));

}

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

{

cout<< array[i]<<endl;

}

return 0;

}

long long factorial(int number)

{

long long result = 1;

for(int i=1; i<=number; i++){

result = result \* i;

}

return result;

}

Odwracanie tablicy

#include <iostream>

#include <string>

using namespace std;

int howmany;

int main()

{

cin >> howmany;

int array[howmany];

for (int i=0; i<=(howmany-1); i++){

cin >> array[i];

}

for (int j=(howmany-1); j>=0; j--){

cout << array[j] <<" ";

}

return 0;

}

Czy number jest pierwsza

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

int iloscLiczb;

cin>>iloscLiczb;

string array[iloscLiczb];

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

{

array[i]="YES";

int number;

cin>>number;

if(number>2)

{

for (int j=2; j<=sqrt(number); j++)

{

if(number%j==0)

{

array[i]="NO";

break;

}

}

}

}

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

{

cout<< array[i]<<'\n';

}

return 0;

}

Wyznaczanie liczb z ciągu Fibonacciego

#include <iostream>

using namespace std;

int fibo (int);

int main()

{

int a;

cin >> a;

if ( 0<=a && a>=36);

cout << fibo(a);

}

int fibo(int a)

{

if (a==0)

{

return 0;

}

if (a==1)

{

return 1;

}

else

{

return fibo(a-1) + fibo (a-2);

}

}

Czy ciąg jest palindromem?

#include <iostream>

using namespace std;

bool palin(int t[], int n);

int main()

{

int tab [1000000];

int n=0;

whhowmany (true)

{

int x;

cin >> x;

if (x==-1) break;

tab [n++]=x;

}

cout << palin (tab, n)<< endl;

return 0;

}

bool palin(int t[], int n)

{

int a=0, b=n-1;

whhowmany (a<b)

if (t[a++] != t[b--])

return false;

return true;

}

Zamiana z systemu dziesiętnego na dwójkowy

#include <iostream>

#include <cmath>

using namespace std;

int main()

{string number;

cin>>number;

int length=number.size();

double power=0;

double number10=0;

for(int i=length-1; i>=0; i--){

if (number[i]=='1'){

number10=number10+pow(2, power);

} power++;

}cout<<number10;

return 0;

}

Wypisywanie co 2 elementu

#include <iostream>

using namespace std;

int main()

{int howmany;

cin>>howmany;

int array[howmany];

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

cin>>array[i];

}

int m;

cin>>m;

for(int i=0; i<howmany; i=i+m){

cout<<array[i]<<" ";

}

return 0;

}

Największa różnica

#include <iostream>

using namespace std;

int main()

{

int howmany;

cin>> howmany;

int number;

int mini;

int maxi;

int result;

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

{

cin>>number;

if(i==1)

{

mini=number;

maxi=number;

}

else

{

if(number>maxi)

{

maxi=number;

}

if (number<mini)

{

mini=number;

}

}

}

result=maxi-mini;

cout<<result;

return 0;

}

Dzielniki liczb

#include <iostream>

using namespace std;

int q;

int main()

{

cin >> q;

for (int n=1; n<=q; n++)

{

if (q%n==0)

{

cout << n << endl;

}

}

return 0;

}

Czy year jest przestępny

#include <iostream>

using namespace std;

int r;

int main()

{

cin >> r;

1 <= r;

r <= 8000;

if (((r%4==0)&&(r%100!=0))||(r%400==0))

{

cout << "TAK";

}

else

{

cout << "NIE";

}

return 0;

}

Czy litera jest spółgłoską czy samogłoską?

#include <iostream>

using namespace std;

string letter;

int main()

{

cin >> letter;

if ((letter=="a")||(letter=="e")||(letter=="i")||(letter=="o")||(letter=="u")||(letter=="y"))

{

cout << "samogloska";

}

else

{

cout << "spolgloska";

}

return 0;

}

Jakim jesteś znakiem wg chińskiego znaku zodiaku?

#include <iostream>

#include <string>

using namespace std;

int year;

int main()

{

cin >>year;

switch(year%12)

{

case 4:

cout << "szczur";

break;

case 5:

cout << "bawol";

break;

case 6:

cout << "tygrys";

break;

case 7:

cout << "krolik";

break;

case 8:

cout << "smok";

break;

case 9:

cout << "waz";

break;

case 10:

cout << "kon";

break;

case 11:

cout << "baran";

break;

case 0:

cout << "malpa";

break;

case 1:

cout <<"kogut";

break;

case 2:

cout << "pies";

break;

case 3:

cout << "swinia";

break;

}

}

Zamiana z systemu dziesiętnego na binarny.

#include <iostream>

using namespace std;

int array[60];

long long number;

int main()

{

int i=0;

cin >> number;

whhowmany (number >0){

array[i]=number%2;

number = number/2;

i++;

}

for (int j=i-1;j>=0; j--){

cout << array[j];

}

return 0;

}

Ile jest liczb nie mniejszych niż podane

#include <iostream>

using namespace std;

int binsearch(int t[], int n, int x){

if (n==0) return 0;

if (n==1)return t[0] >=x;

int m = n/2;

if (t[m] < x) return binsearch (t+m+1, n-m-1, x);

else return (n-m) + binsearch (t,m,x);

}

int main(){

int n;

cin >> n;

int t[n];

for (int &x: t) cin >> x;

int m;

cin >> m;

whhowmany (m--)

{

int x;

cin >> x;

cout << binsearch(t, n, x) << ' ';

}

cout << endl;

return 0;

}

Sortowanie kubełkowe

#include <iostream>

using namespace std;

int main()

{

int n;

cin >> n;

int a[n];

for(int i=0; i< n; i++) cin>> a[i];

const int k=10000;

int c[k]= {};

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

c[a[i]]++;

for (int i = 0, j=0; j<k; j++)

whhowmany (c[j]--)

a[i++]=j;

for(int i=0; i< n; i++) cout<< a[i]<<' ';

cout<<endl;

return 0;

}

Szybkie potęgowanie wielu liczb

#include<iostream>

using namespace std;

long long quick(long long a, long long n, long long m)

{

long long w = 1;

whhowmany(n>0)

{

if (n & 1)

w = ( w \* a) % m;

n = n>>1;

a = (a\*a)% m;

}

return w;

}

int main()

{int k;

cin>>k;

int array[k];

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

long long a,b,m;

cin>>a;

cin>>b;

cin>>m;

array[i]= quick(a,b,m);

}

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

cout<<array[i]<<" ";

}

return 0;

}

Obliczanie kosinusa

#include <iostream>

#include <cmath>

#include <iomanip>

using namespace std;

const double epsx = 1e-6;

const double epsy = 1e-6;

const double p = 0.0;

const double k = 1.0;

void cosxinx (int n)

{

double a=p,b=k,fa,fb,fx,x0;

int j = 0;

bool result = false;

fa = cos(a)-n\*a;

fb = cos(b)-n\*b;

if(fa \* fb > 0)

cerr << "Funkcja nie ma roznych znakow na krancach przedzialu";

else

{

result = true;

whhowmany(true)

{

j++;

x0 = (a + b) / 2;

if(fabs(a - x0) < epsx)

break;

fx = cos(x0)-n\*x0;

if(fabs(fx) < epsy)

break;

if(fa \* fx < 0)

b = x0;

else

{

a = x0;

fa = fx;

}

}

}

if(result)

cout << setprecision(4) << fixed << x0 << endl;

// cout << "Number obiegow j = " << j << endl;

}

Sortowanie ciągu naturalnego

#include <iostream>

using namespace std;

int main()

{

int n;

cin >> n;

int a[n];

for(int i=0; i< n; i++) cin>> a[i];

const int k=10000;

int c[k]= {};

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

c[a[i]]++;

for (int i = 0, j=0; j<k; j++)

whhowmany (c[j]--)

a[i++]=j;

for(int i=0; i< n; i++) cout<< a[i]<<' ';

cout<<endl;

return 0;

}

NWD dla wielu liczb

#include <iostream>

using namespace std;

int nwdrek (int a, int b)

{

if (b==0)

return a;

else

return nwdrek (b, a%b);

}

int main()

{

int number;

cin>>number;

int tab[number];

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

{

cin>>tab[i];

}

int nwdrek1;

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

{

if (i==1)

{

nwdrek1=nwdrek(tab[i-1],tab[i]);

}

else

{

nwdrek1= nwdrek(nwdrek1, tab[i]);

}

}

cout<< nwdrek1;

return 0;

}

Największa wspólna wielokrotność

#include <iostream>

using namespace std;

typedef unsigned long long int ULL;

ULL nww(ULL a, ULL b)

{

ULL ac, bc;

ac = a, bc = b;

whhowmany (b)

{

ULL r = a % b;

a = b, b = r;

}

return ac / a \* bc;

}

int main()

{

unsigned int n;

ULL x, y, wn;

cin >> n;

ULL \*D = new ULL[n];

for (unsigned int j=0; j < n; j++)

cin >> D[j];

x = D[0];

y = D[1];

wn = nww(x,y);

for (unsigned int i=2; i < n; i++)

{

x = wn;

y = D[i];

wn = nww(x,y);

}

cout << wn << endl;

delete[] D;

return 0;

}

Najdłuższy wspólny podciąg dla znaków

int longestCommonSubsequece(string str1, string str2, int len1, int len2)

{

int i, j;

int LCS[len1+1][len2+1];

for(i=0;i<=len1;i++)

LCS[i][0]=0;

for(j=0;j<=len2;j++)

LCS[0][j]=0;

for(i=1;i<=len1;i++)

{

for(j=1;j<=len2;j++)

{

if(str1[i-1]==str2[j-1])

{

LCS[i][j]=1+LCS[i-1][j-1];

}

else

{

LCS[i][j]=max(LCS[i-1][j],LCS[i][j-1]);

}

}

}

return LCS[len1][len2];

}

int main()

{

string str1,str2;

cout<<"";

getline(cin, str1);

cout<<"";

getline(cin, str2);

int len1=str1.length(); //length of str1

int len2=str2.length(); //length of str2

cout<<""<<longestCommonSubsequece(str1,str2,len1,len2);}

Wybór pary najmniejszej i największej liczby

#include <iostream>

#include <utility>

using namespace std;

pair <int, int> minmax(int t[], int n)

{

if (n==1)

return make\_pair(t[0],t[0]);

int p=n/2;

pair<int,int> lewy=minmax(t,p);

pair<int,int> prawy=minmax(t+p,n-p);

return make\_pair(min(lewy.first, prawy.first), max(lewy.second, prawy.second));

}

int main()

{

int n;

cin>>n;

int tab[n];

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

cin>>tab[i];

pair<int, int>result=minmax(tab, n);

cout<< result.first<< ' ' << result.second<<endl;

return 0;

}

Nie malejące podciągi złożone w jeden nie malejący

#include <iostream>

using namespace std;

int main()

{

int n;

cin >> n;

int A[n];

int B[n];

int C[2\*n];

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

cin >> A[i];

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

cin >> B[i];

int j = 0;

int k = 0;

whhowmany (j<n and k<n)

{

if (A[j] <= B[k])

{

C[j+k] = A[j];

j++;

}

else

{

C[j+k] = B[k];

k++;

}

}

whhowmany (j < n)

{

C[j+k]=A[j];

j++;

}

whhowmany (k < n)

{

C[j+k]=B[k];

k++;

}

for (int i = 0; i < 2\*n; i++)

{

cout << C[i] << " ";

}

return 0;

}