**Готуємось до олімпіади з інформатики**

1. Сума арифметичної прогресії

|  |  |
| --- | --- |
| int s=0;  for(int i=a1;i<=an;i=i+d)s=s+i; | S_n=\sum_{i=1}^n a_i ={a_1+a_n \over 2}n={2a_1 + d(n-1) \over 2}n.  int s=(a1+an)\*n/2; |

1. Числа Фібоначі

|  |  |  |
| --- | --- | --- |
| int a1(1),a2(1);  for (int i=3;i<=n;i++)  {a3=a1+a1;  a1=a2;  a2=a3;  } | int a[100];  a[1]=1; a[2]=1;  for (int i=3;i<=n;i++)  a[i]=a[-1]+a[i-2]; | #include "stdafx.h"  #include <iostream>  #include <math.h>  using namespace std;  int main(){  double n,f;  cin>>n;  f=1/sqrt(5.0)\*(pow((1+sqrt(5.0))/2.0,n)+pow((1-sqrt(5.0))/2.0,n));  **cout.precision(0);**  **cout<<fixed<<f<<endl;**  return 0;  } |

1. Алгоритм Евкліда

|  |
| --- |
| #include "stdafx.h"  #include <iostream>  using namespace std;  int main(){  int a,b,a1,b1,t;  cin>>a>>b;  a1=a;b1=b;  while (b!=0)  {t=a%b;  a=b;  b=t; }  int nsd=a;  int nsk=a1\*b1/nsd;  cout<<nsd<<endl;  cout<<nsk<<endl;  return 0;  } |

1. Прості числа (решето Ератосфена)

|  |  |
| --- | --- |
| #include "iostream"  #include "math.h"  using namespace std;  int main()  {int n,p;  cin>>n;  for(int i=2;i<=n; i++)  {p=0;  for(int j=2;j<=int(sqrt(double (i)));j++)  if (i%j ==0)p=1;  if (p==0) cout<<i<<endl;  }  return 0;  } | #include "iostream"  #include "math.h"  using namespace std;  int a[10000];  int main()  {int n,p,i,j;  cin>>n;  for(i=1;i<=n; i++)a[i]=i;  a[1]=0;i=1;  while (i<=n/2)  {  while (a[i]==0)i++;  j=i+a[i];  while (j<=n)  {a[j]=0;  j=j+a[i];}  i++;  }  for(i=1;i<=n;i++)  if (a[i]!=0 ) cout<<a[i]<<" ";  } |

1. Зчитування та порівняння рядків та символів

|  |  |  |
| --- | --- | --- |
| char a;  cin>>a;  if (a=='A') cout<<"ok"<<endl;else cout<<"no"<<endl; | #include “string.h”  char a[1000],b[1000];  cin.getline(a,sizeof(a));  cin.getline(b,sizeof(b));  if (strcmp(a,b)==0)cout<<”ok”<<endl;  {Менше нуля a менше b  Нуль a рівне b  Більше нуля a більше b  } | #include <fstream>  #include <string>  using namespace std;  int main(){  ifstream cin(“input.txt”);  ofstream cout(“output.txt”);  string s1, s2;  getline (cin, s1);  getline (cin, s2);  cout<<s1<<endl;  cout<<s2<<endl;  if (strcmp(s1.c\_str(),s2.c\_str())==0)  cout << "Ok..\n";    return 0;  } |

1. Швидке сортування

|  |
| --- |
| #include<iostream>  #include<algorithm>  #include<vector>  using namespace std;  int main()  {int n,i;  cin>>n; vector<int> a(n);  for (i=0; i<n; i++)cin>>a[i];  sort(a.begin(),a.end());  for (i=0; i<n-1; i++)cout<<a[i]<<" ";  cout<<a[n-1]<<"\n";  return 0;  } |

1. Повний перебір

|  |  |
| --- | --- |
| #include <fstream>  #include <iostream>  #include <math.h>  #include <vector>  #include <algorithm>  #include <time.h>  using namespace std;  vector <int> a;  // ifstream cin("input.dat");  //ofstream cout("output.ans");    void printper(int n);  int main()  {  int n;  cin >> n;  for (int i=1;i<=n;i++){a.push\_back(i);  } | printper(n);    clock\_t start = clock();  while (next\_permutation(a.begin(),a.end())){  // printper(n);  };  // час перебору  long double r=(clock() - start)\*1. / CLOCKS\_PER\_SEC;  cout<<"times work = "<<r<<endl;  return 0;  }  void printper(int n)  {  for (int i=0;i<n-1;i++){  cout << a[i] << " ";  }  cout << a[n-1] << endl;  } |

1. Векторний добуток

|  |
| --- |
| #include<fstream>  using namespace std;  ifstream cin("input.txt");  ofstream cout("output.txt");  int main()  {  int i,n,m,k=0,x[10001],y[10001],a[10001],b[10001],v[10001];  cin>>n>>m;  for(i=1;i<=n;i++) cin>>x[i]>>y[i];  for(i=1;i<=n-1;i++)  {  a[i]=x[i+1]-x[i];  b[i]=y[i+1]-y[i];  }  for(i=1;i<=n-2;i++){  v[i]=a[i]\*b[i+1]-a[i+1]\*b[i];  if(v[i]>0) k++;}  cout<<k\*m<<endl;  } |

1. Повний обхід дерева

|  |
| --- |
| #include "iostream"  using namespace std;  int c[100],n;  void p(int i,int v)  {c[i]=v;  if(i==n)  {for(int j=1;j<=n;j++)  cout<<c[j];cout<<endl;}  else{p(i+1,0);p(i+1,1);}}  int \_tmain()  {cin>>n;  p(1,0);  p(1,1);  } |

1. Рекурсивний пошук шляху в графі

|  |
| --- |
| #include "iostream"  using namespace std;  int a[100][100],c[100],n;  void p(int i, int v)  {c[i]=v;  if (v==n || i>n)  {for(int j=1;j<=n;j++)  cout<<c[j]<<" ";  cout<<endl;  }  else  for(int j=1;j<=n;j++)  if(a[v][j]>=1) p(i+1,j);  }  int main()  {  cin>>n;  for(int i=1;i<=n;i++)  for (int j=1;j<=n;j++)cin>>a[i][j];  p(1,1);  } |

1. Побудова остового дерева

|  |  |
| --- | --- |
| #include <iostream>  #include <math.h>  using namespace std;  int main()  {int n,int p=0;  int a[n][n];  int x[1000],y[1000],kol\_ver[1000],v[1000];  int k,i,j,vi,vj,min,s;  int ver[1000][3];  int f;  for(i=0;i<n;i++)  for (j=0;j<n;j++)  cin>>a[i][j];  k=0; v[k]=p;s=0; | while (k<n-1) {  min=100000;  for (i=0;i<=k;i++)  for(j=0;j<n;j++)  if (a[v[i]][j]<min) {min=a[v[i]][j];vi=v[i];vj=j;}  f=1;  for (i=0;i<=k;i++)if (vj==v[i])f=0;  if (f==1) {k=k+1;  ver[k][1]=vi;  ver[k][2]=vj;  v[k]=vj;  kol\_ver[vj]=kol\_ver[vj]+1;  kol\_ver[vi]=kol\_ver[vi]+1;  s=s+a[vi][vj];  }  a[vi][vj]=1e30;a[vj][vi]=1e30;  }    cout<<s<<endl;  for(i=1;i<n;i++) cout<<ver[i][1]<<' '<<ver[i][2]<<endl;  return 0;  } |

1. Пошук мінімального шляху (Флойд)

|  |
| --- |
| for (int k=0; k<n; ++k)  for (int i=0; i<n; ++i)  for (int j=0; j<n; ++j)  if (d[i][j]< d[i][k] + d[k][j])  d[i][j] = d[i][j]; else d[i][j]=d[i][k] + d[k][j]; |

**ІІI етап Всеукраїнської учнівської олімпіади з інформатики (м.Луцьк) 2012-2013н.р**

<http://176.31.28.231/cgi-bin/new-client?contest_id=11>

**ІІI етапВсеукраїнськоїучнівськоїолімпіадизінформатики (м.Луцьк) 2013-2014н.р**

<http://www.e-olimp.com.ua/problems/7234>

<http://www.e-olimp.com.ua/ua/problems/7235>

<http://www.e-olimp.com.ua/ua/problems/7236>

<http://www.e-olimp.com.ua/ua/problems/7237>

<http://www.e-olimp.com.ua/ua/problems/7238>

<http://www.e-olimp.com.ua/ua/problems/7239>

<http://www.e-olimp.com.ua/ua/problems/7240>

<http://www.e-olimp.com.ua/ua/problems/7241>

<http://www.e-olimp.com.ua/ua/problems/7242>

<http://www.e-olimp.com.ua/ua/problems/7243>