1. Системи числення

**Переведення чисел з однієї системи числення в іншу**

|  |  |  |
| --- | --- | --- |
| **Десяткова** | **Двійкова** | **Відмітка про виконання** |
| 0 | 0 |  |
| 2 | 10 |  |
| 5 | 101 |  |
| 10 | 1010 |  |
| 32 | 100000 |  |
| 98 | 1100010 |  |
| 1024 | 1000000000 |  |
| 6783 | 1101001111111 |  |
| 98321 | 11000000000010001 |  |
| 2000000 | 111101000010010000000 |  |
| 1073741824 | 1000000000000000000000000000000 |  |
| 5000000000 | 100101010000001011111001000000000 |  |

**Переведення чисел в різних системах числення**

На приклад, якщо потрібно перемножити числа 101 і 1001 в двійковій системі, то він спочатку ці числа переводить в десяткову систему таким чином :

(101)2=1\*22+0\*21+1\*20=4+0+1=5

(1001)2=1\*23+0\*22+0\*21+1\*20=8+0+0+1=9

Після чого множення чисел 5 і 9 Вася з легкістю виконує в десятковій системі числення і отримує число 45. Далі виконує переведення з десяткової системи числення в двійкову. Для цього потрібно ділити число 45 на 2 ( порядок системи числення ), запам'ятовує залишки від ділення, до тих пір поки в результаті не залишиться число 0:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 45 | 2 |  |  |  |  |
| 44 | 22 | 2 |  |  |  |
| 1 | 22 | 11 | 2 |  |  |
|  | 0 | 10 | 5 | 2 |  |
|  |  | 1 | 4 | 2 | 2 |
|  |  |  | 1 | 0 | 1 |

Відповідь складається з одержаних залишків від ділення шляхом їх запису в зворотному  порядку . Таким чином одержуємо результат : (101101)2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **10** | **2** | **4** | **8** | **16** |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 10 | 2 | 2 | 2 |
| 3 | 11 | 3 | 3 | 3 |
| 4 | 100 | 10 | 4 | 4 |
| 5 | 101 | 11 | 5 | 5 |
| 6 | 110 | 12 | 6 | 6 |
| 7 | 111 | 13 | 7 | 7 |
| 8 | 1000 | 20 | 10 | 8 |
| 9 | 1001 | 21 | 11 | 9 |
| 10 | 1010 | 22 | 12 | A |
| 11 | 1011 | 23 | 13 | B |
| 12 | 1100 | 30 | 14 | C |
| 13 | 1101 | 31 | 15 | D |
| 14 | 1110 | 32 | 16 | E |
| 15 | 1111 | 33 | 17 | F |

1. Перестановки

|  |  |
| --- | --- |
| #include <iostream>  #include <algorithm>  using namespace std;  int main()  {  int n,a[100];  cin >> n;  for (int i=0;i<n;i++)a[i]=i+1;  for (int i=0;i<n;i++)cout << a[i] << " ";cout<< endl;    while (next\_permutation(a,a+n))  {for (int i=0;i<n;i++)cout << a[i] << " ";cout<< endl;};    return 0;  } | #include <iostream>  #include <vector>  #include <algorithm>  using namespace std;  int main()  { int n;  cin >> n;  vector <int> a;  for (int i=0;i<n;i++)a.push\_back(i+1);  for (int i=0;i<n;i++)cout << a[i] << " ";cout<< endl;  while (next\_permutation(a.begin(),a.end()))  {for (int i=0;i<n;i++)cout << a[i] << " ";cout<< endl;};  } |

1. Площа многокутника (опуклість)

|  |  |
| --- | --- |
| 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;   } |
| Площа многокутника | #include "iostream"  #include "math.h"  using namespace std;  int main()  {double x[100],y[100],s,p; int i,n;  cin>>n;  for(i=1;i<=n;i++)cin>>x[i]>>y[i];  x[n+1]=x[1];y[n+1]=y[1]; s=0; p=0;  for(i=1;i<=n;i++)  {s=s+x[i]\*y[i+1]-x[i+1]\*y[i];  p=p+sqrt(pow(x[i]-x[i+1],2)+pow(y[i]-y[i+1],2));  }  s=0.5\*abs(s);  cout<<s<<endl;  cout<<p<<endl;  return 0;} |

1. Цікавий код

|  |  |
| --- | --- |
| Зчитування рядка з пропусками (тип char) | cin.getline(str,sizeof(str)); |
| Зчитування до кінця рядка | while (cin.peek()!='\n')  { n++;  cin>>a[n];  } |
| Зчитування до кінця файлу | while (!cin.eof())  { m++;  cin>>b[m];  } |
| Сортування | * #include<algorithm> * sort(a,a+n); |
| 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);   } |
| 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);  } |
| 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;  } |
| 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) * d[i][j] = min(d[i][j], d[i][k] + d[k][j]); |

<https://www.e-olymp.com/uk/problems/1001>

<https://www.e-olymp.com/uk/problems/1784>

<https://www.e-olymp.com/uk/problems/7820>

<https://www.e-olymp.com/uk/problems/7817>

<https://www.e-olymp.com/uk/problems/7818>

<https://www.e-olymp.com/uk/problems/7809>

<https://www.e-olymp.com/uk/problems/7802>

<https://www.e-olymp.com/uk/problems/7330>

<https://www.e-olymp.com/uk/problems/7401>

<https://www.e-olymp.com/uk/problems/7402>

<https://www.e-olymp.com/uk/problems/7403>

<https://www.e-olymp.com/uk/problems/7404>

<https://www.e-olymp.com/uk/problems/7405>

<https://www.e-olymp.com/uk/problems/7406>

<https://www.e-olymp.com/uk/problems/7407> -----------------

<https://www.e-olymp.com/uk/problems/7408>

<https://www.e-olymp.com/uk/problems/7409>

<https://www.e-olymp.com/uk/problems/7410>

<https://www.e-olymp.com/uk/problems/7411>

<https://www.e-olymp.com/uk/problems/7412>