**Готуємось до олімпіади з інформатики – 2016 – 3**

|  |  |  |
| --- | --- | --- |
| № | Завдання | Програма |
| 1 | Задачі -5  Система – Ejudge  Робота з файлами – input.txt, output.txt | #include "fstream"  using namespace std;  ifstream cin("input.txt");  ofstream cout("output.txt"); |
| 2 | Обчислювальні алгоритми   * Цілі числа та операції – int, long long, /, % * Дійсні числа – float, double, sqrt, pow (math.h) | Задача 1. Час в секундах подати гг хх сс  #include <iostream>  using namespace std;  int main()  { long long t,g,h,s;  cin >>t;  s=t%60;  t=t/60;  h=t%60;  t=t/60;  g=t%60;  cout << g<<" "<<h<<" "<<s<<endl;  return 0;  }  Задача 2. За введеними координатами протилежних вершин прямокутника зі сторонами паралельними осям координат обчислити його площу, периметр і радіус описаного кола  #include <iostream>  #include <math.h>  using namespace std;  int main()  { double x1,y1,x2,y2,p,s,r,a,b;  cin >>x1>>y1>>x2>>y2;  a=fabs(x1-x2);  b=fabs(y1-y2);  s=a\*b;  p=2\*(a+b);  r=sqrt(pow(a,2)+pow(b,2))/2;  cout.precision(3);  cout <<fixed;  cout <<s<<endl;  cout <<p<<endl;  cout <<r<<endl;  return 0;  } |
| 3 | Структура розгалуження і цикулу | Задача 3. Кількість нулів  За даним числу n визначте, якою кількістю нулів закінчується десяткова запис числа n! .  #include "iostream" using namespace std; int main() {  long long n,f; cin>>n; if (n<5){cout<<0;} if (n>=5 && n<10) cout<<1; if (n>=10 && n<15)cout<<2; if (n>=15 && n<20)cout<<3; if (n>=20 && n<25)cout<<4; if (n>=25 && n<30)cout<<6; if (n>=30 && n<35)cout<<7; if (n>=35 && n<40)cout<<8; if (n>=40 && n<45)cout<<9; if (n>=45 && n<50)cout<<10; if (n>=50 && n<55)cout<<12;  }  #include "iostream" using namespace std; int main() { long long n,f,i,d,s; cin>>n;  d=1;s=0; for(i=1;i<=100;i++) {         d=d\*5;         s=s+n/d; } } |
| 4 | Лінійний масив | Задача 4. Кількість максимальних елементів в масиві  #include <iostream>  #include <math.h>  using namespace std;  int a[10000],n;  int main()  { int max,nmax,k;  cin >>n;  for (int i=0;i<n;i++) cin>>a[i];  max=a[0];nmax=0;  for (int i=0;i<n;i++)  if(a[i]>max){max=a[i];nmax=i;}  k=0;  for (int i=0;i<n;i++)  if(a[i]==max)k++;  cout <<k<<endl;  return 0;  }  #include <iostream>  #include <math.h>  using namespace std;  int a[10000],n;  int main()  { int max,nmax,k,j;  cin >>n;  for (int i=0;i<n;i++) cin>>a[i];  for (int i=0;i<n-1;i++)  for (int j=0;j<n-1;j++)  if(a[j]>a[j+1])swap(a[j],a[j+1]);  j=n-1;k=1;  while (a[j]==a[j-1] && j>0){k++;j--;}  cout <<k<<endl;  return 0;  } |

Задача 5. Домашня робота з математики (100 балів)

|  |  |
| --- | --- |
| Максимальний час роботи на одному тесті: | 2 секунди |
| Максимальний об’єм пам’яті: | 256 Мб |
| Ім’я вхідного файлу: | input.txt |
| Ім’я вихідного файлу: | output.txt |

Сашко не дуже любить робити домашні завдання, але на попередньому уроці математики Петро Павлович задав учням *n* різних завдань. Причому, робити деякі домашні завдання можна було лише після того, як виконано інші.

Для кожного завдання Сашко визначив, скільки хвилин йому потрібно, щоб його виконати. Після цього Сашко зрозумів, що виконати всі завдання він точно не встигне. Отож він вирішив зробити всі завдання окрім одного – за одне невиконане завдання вчитель нічого не скаже.

Тепер Сашку слід вибрати, яке завдання не виконувати. Допоможіть Сашку вибрати завдання, яке можна не виконувати, щоб інші завдання виконати якомога швидше.

Формат вхідного файлу

Перший рядок вхідного файлу містить цілі числа *n* и *m* — кількість завдань і кількість залежностей між завданнями (1 *≤ n ≤* 100, 0 *≤ m ≤* 1000). Другий рядок містить *n* цілих чисел: *t*1*, t*2*, . . . , tn*. Число *ti* означає кількість хвилин, необхідних Сашку для виконання *i*-го завдання (1 *≤ ti ≤* 1000). Далі іде *m* рядків, кожен з яких містить два цілих числа. Числа *a* и *b* означають, що

завдання *a* слід виконати раніше аніж завдання *b*. Гарантується, що всі завдання можна виконати.

Формат вихідного файлу

Вивести одне число – мінімальну кількість хвилин, необхідних Сашку для виконання всіх завдань крім одного.

Наприклад

|  |  |
| --- | --- |
| **Вхідні дані** | **Вихідні дані** |
| 5 5  1 2 3 4 5  1 2  5 3  1 3  3 4  2 4 | 11 |

В даному прикладі Сашко може не виконувати **четверте** завдання. Всі інші завдання він виконає за 11 хвилин.

|  |  |  |
| --- | --- | --- |
| #include "fstream"  using namespace std;  int a[1000],b[1001],c[1001];  ifstream cin("input.txt");  ofstream cout("output.txt");  int main()  {  long long n,m,max,nmax,s=0,f2;  cin>>n>>m;  for (int i=1; i<=n;i++) {cin>>a[i];s=s+a[i];}  for (int i=1; i<=m;i++) cin>>b[i]>>c[i];  int f=1;  while (f)  {max=0;nmax=0;  for (int i=1; i<=n;i++) if (a[i]>max){max=a[i];nmax=i;}  f2=0;  for (int i=1; i<=m;i++)  if(b[i]==nmax)f2=1;  if(f2)a[nmax]=0; else f=0;  }  cout<<s-max<<endl;  } | | |
| 5 | Прямокутна таблиця | Задача 6. **На квадратному аркуші паперу в клітинку розміром NхM клітинок намальовано кілька прямокутників. Кожен прямокутник складається з цілих клітинок, різні прямокутники накладаються один на одного.**  **Необхідно написати програму, яка рахує площу покриту цими прямокутниками.**  Вхідні дані  **В першому рядку N, розмір масиву, в наступних n рядків масиву, в кожному з яких написані через пробіл n елементів масиву: A [елемент I, J] = 1, якщо клітина [I, J] належить будь-якому прямокутника, і A [I, J ] = 0, в іншому випадку.**  Вихідні дані  **Необхідно вивести єдине число - кількість прямокутників.**  #include <iostream>  using namespace std;  int a[100][100];  int main()  {  int n,m;  cin>>n>>m;  for (int i=0;i<n;i++)  for (int j=0;j<m;j++)  cin>>a[i][j];  int k=0;  for (int i=0;i<n;i++)  for (int j=0;j<m;j++)  if (a[i][j]==1)k++;  cout<<k<<endl;  return 0;  }  Задача 7. **"**Прямокутники**" На квадратному аркуші паперу в клітинку розміром NхN клітин намальовано кілька прямокутників. Кожен прямокутник складається з цілих клітинок, різні прямокутники не накладаються один на одного і не дотикаються.**  **Необхідно написати програму, яка рахує число цих прямокутників.**  Вхідні дані  **В першому рядку N, розмір масиву, в наступних n рядків масиву, в кожному з яких написані через пробіл n елементів масиву: A [елемент I, J] = 1, якщо клітина [I, J] належить будь-якому прямокутника, і A [I, J ] = 0, в іншому випадку.**  Вихідні дані  **Необхідно вивести єдине число - кількість прямокутників.**  #include <fstream>  using namespace std;  ifstream cin("input.txt");  ofstream cout("output.txt");  int a[100][100];  int main()  {  int n;  cin>>n;  for (int i=0;i<n;i++)  for (int j=0;j<n;j++)  cin>>a[i][j];  int k=0;  for (int i=0;i<n;i++)  for (int j=0;j<n;j++)  if (a[i][j]==1 && a[i+1][j]==0 &&a[i][j+1]==0 && a[i+1][j+1]==0)k++;  cout<<k<<endl;  return 0;  } |
| 6 | Рядки  Задача 8. Підрахувати кількість цифри 0 в рядку  #include <iostream>  #include <string.h>  using namespace std;  char a[10000];  int main()  {  cin>>a;  int n=strlen(a);  int k=0;  for (int i=0;i<n;i++)  if(a[i]=='0')k++;  cout<<k<<endl;  return 0;  } | Задача 9. В тесті залишити всі голосні букви інші замінити “\_” та порахувати їх кількість  #include <fstream>  using namespace std;  ifstream cin("input.txt");  ofstream cout("output.txt");  char a[1000];  int main()  {  int ka=0, ke=0,ki=0, ko=0, ku=0, ky=0;  while (!cin.eof())  {cin>>a;  for(int i=0;i<strlen(a);i++)  {if(a[i]=='a')ka++; else  if(a[i]=='e')ke++; else  if(a[i]=='i')ki++;else  if(a[i]=='o')ko++; else  if(a[i]=='u')ku++;else  if(a[i]=='y')ky++;else  a[i]='\_';  }  cout<<a<<endl;  }  cout<<ka<<" "<<ke<<" "<<ki<<" "<<ko<<" "<<ku<<" "<<ky<<endl;  return 0;  }  #include "fstream"  #include "string.h"  #include "string"  using namespace std;  ifstream cin("input.txt");  ofstream cout("output.txt");  string a[1000];  int main()  {int n=0;  int ka=0, ke=0,ki=0, ko=0, ku=0, ky=0;  while (!cin.eof())  {getline(cin,a[n]);  for(int i=0;i<a[n].length();i++)  {if(a[n][i]=='a')ka++; else  if(a[n][i]=='e')ke++; else  if(a[n][i]=='i')ki++;else  if(a[n][i]=='o')ko++; else  if(a[n][i]=='u')ku++;else  if(a[n][i]=='y')ky++;else  a[n][i]='\_';  }  n++;  }  for(int i=0;i<n;i++)  cout<<a[i]<<endl;  cout<<ka<<" "<<ke<<" "<<ki<<" "<<ko<<" "<<ku<<" "<<ky<<endl;  return 0;  } |
| 7 | Динамічне програмування | Задача 10. Квиток |

**Задача 4. «Квитки»** (30 балів)

**Ім’я файлу програми: BILET.\***

**Ім’я вхідного файлу: input.txt**

**Ім’я вихідного файлу: output.txt**

**Максимальний час роботи на одному тесті: 5с**

За квитками на прем'єру нового мюзиклу вишикувалася черга з *N* людей, кожний з яких хоче купити 1 квиток. На всю чергу працювала тільки одна каса, тому продаж квитків йшов дуже повільно, приводячи людей черги у відчай. Найкмітливіші швидко відмітили, що, як правило, декілька квитків в одні руки касир продає швидше, ніж коли ці ж квитки продаються поодинці. Тому вони запропонували декільком людям, які стоять підряд віддавати гроші першому з них, щоб він купив квитки на всіх.

Проте для боротьби із спекулянтами касир продавала не більше 3-х квитків в одні руки, тому домовитися таким чином між собою могли лише 2 або 3 підряд вартих людини.

Відомо, що на продаж *i-ій* людині з черги одного квитка касир витрачає Ai секунд, на продаж двох квитків — *Bi* секунд, трьох квитків — *Ci* секунд. Напишіть програму, яка підрахує мінімальний час, за який могли бути продані квитки для всіх людей черги.

Зверніть увагу, що квитки на групу людей, що об'єдналися, завжди купує перший з них. Також ніхто в цілях прискорення не купує зайвих квитків (тобто квитків, які нікому не потрібні).

Формат вхідних даних

У вхідному файлі записано спочатку число N — кількість покупців в черзі (1≤N≤5000). Далі йде N трійок натуральних чисел Ai*,* Bi*,* Ci*.* Кожне з цих чисел не перевищує 3600. Люди в черзі нумеруються починаючи від каси.

Формат вихідних даних

У вихідний файл виведіть одне число — мінімальний час в секундах, за яке могли бути обслужені всі покупці.

Приклади файлів

|  |  |  |  |
| --- | --- | --- | --- |
| **input.txt** | **output.txt** | **input.txt** | **output.txt** |
| 5  5 10 15  2 10 15  5 5 5  20 20 1  20 1 1 | 12 | 2  3 4 5  1 1 1 | 4 |

*N*=5

|  |  |  |  |
| --- | --- | --- | --- |
| ***i*** | ***Ai*** | ***Bi*** | ***Ci*** |
| 1 | 5 | 10 | 15 |
| 2 | 2 | 10 | 15 |
| 3 | 5 | 5 | 5 |
| 4 | 20 | 20 | 1 |
| 5 | 20 | 1 | 1 |

D[i]= min ( D[i-1]+*Ai*, D[i-2]+ *Bi-1*, D[i-3]+ *Ci-2* )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| D[1] | D[2] | D[3] | D[4] | D[5] |
| 5 | 7 | 5 | 6 | **12 – відповідь завдання** |

d[0]= 0;

d[1]= а[1];

d[2]= Мінімальне(а[1]+a[2], b[1]);

Для i від 3 до n пц

d[i]= Мінімальне(d[i-1]+ а[i],Мінімальне(d[i-2]+ b[i-1], d[i-3]+ с[i-2]));

#include<iostream>

using namespace std;

int main()

{

int n,i,a[5000],b[5000],c[5000],d[5000];

cin>>n;

for(i=1;i<=n;i++) cin>>a[i]>>b[i]>>c[i];

d[0]= 0; d[1]= a[1]; d[2]= min(a[1]+a[2],b[1]);

for(i=3;i<=n;i++) d[i]=min(d[i-1]+a[i],min(d[i-2]+b[i-1],d[i-3]+c[i-2]));

cout<<d[n]<<endl;

}

|  |  |  |
| --- | --- | --- |
| 8 | Еврестичні методи  cin>>n; . . . cout<<n;  cin>>n; . . . cout<<”yes”; |  |
| 9 |  |  |

<http://134.249.159.199/cgi-bin/new-client?contest_id=23>

Логін school2016-1 . . . school2016-10 (пароль - 1)

<http://nvk26.lutsk.ua/cgi-bin/new-client?contest_id=24>

Логін school2016-1 . . . school2016-10 (пароль - 1)

ІІ етап Всеукраїнської учнівської олімпіади з інформатики (м.Луцьк) 2015-2016н.р. - <http://134.249.159.199/cgi-bin/new-client?contest_id=21>

Логін school2016-1 . . . school2016-10 (пароль - 1)

<http://134.249.159.199/cgi-bin/new-client?contest_id=22>

school1-school10 (пароль - 1)

<http://134.249.159.199/cgi-bin/new-client?contest_id=14>

Логін school2016-1 . . . school2016-10 (пароль - 1)