C1 Question text

Nhập vào một số nguyên n. Kiểm tra xem số đó có là số hoàn hảo không. Biết số hoàn hảo là một số nguyên dương mà tổng các ước nguyên dương chính thức của nó (số nguyên dương bị nó chia hết ngoại trừ nó) bằng chính nó. Ví dụ số 6 là số hoàn hảo vì 6=1+2+3.

Input: Số nguyên n kiểu int được nhập từ bàn phím. Ví dụ: -30

Output: In ra thông báo "n la so hoan hao" hoặc "n khong la so hoan hao".

Ví dụ: "6 la so hoan hao"

Contrain: n kiểu int

**For example:**

| **Input** | **Result** |
| --- | --- |
| 2005 | 2005 khong la so hoan hao |

Answer:(penalty regime: 0 %)

#include<stdio.h>

int main(){

int n;

scanf("%d",&n);

int i, tong=0;

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

if(n%i==0)

tong+=i;

if(tong==n)

printf("%d la so hoan hao",n);

else

printf("%d khong la so hoan hao",n);

}

C2 Question text

Viết chương trình giải phương trình bậc 2 ax2+bx+c =0. Với hệ số a, b, c được nhập từ bàn phím.

Yêu cầu: Xét tất cả các trường hợp.

Input:

a, b, c là các số thực được nhập từ bàn phím

Output:

+ Nếu vô nghiệm ghi VN

+ Nếu vô số nghiệm ghi VSN

+ Nếu có 1 nghiệm ghi giá trị nghiệm với độ chính xác 3 chữ số thập phân

+ Nếu có 2 nghiệm ghi giá trị nghiệm với độ chính xác 3 chữ số thập phân, các nghiệm cách nhau dấu cách. Nghiệm nhỏ hơn đứng trước.

Ví dụ:

Input: 3 5 -8

Output:

-2.670 1.000

Constrains:

a, b, c là các số thực

**For example:**

| **Input** | **Result** |
| --- | --- |
| -20.000 -1.000 4.000 | -0.473 0.423 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<math.h>

int main(){

float a,b,c;

scanf("%f%f%f",&a,&b,&c);

if(a==0){

if(b==0){

if(c==0) printf("VSN");

else printf("VN");

}else printf("%.3f",-c/b);

}else{

float D,x1,x2,x3;

D=b\*b-4\*a\*c;

if(D>0){

x1= (-b-sqrt(D))/(2\*a);

x2= (-b+sqrt(D))/(2\*a);

if(x1<x2) printf("%.3f %.3f", x1,x2);

else printf("%.3f %.3f", x2,x1);

}else if(D==0){

x3=-b/(2\*a);

printf("%.3f", x3);

}else printf("VN");

}

}

C3 Question text

Viết chương trình giải hệ phương trình bậc nhất 2 ẩn

a1x+b1y=c1

a2x+b2y=c2

Input: Các hệ số a1, b1, c1 và a2, b2, c2 cách nhau dấu cách, trên 2 dòng. Ví dụ:

2 3 4

7 8 9

Output: Nếu có nghiệm thì in ra nghiệm x, y cách nhau dấu cách, các nghiệm có độ chính xác 5 chữ số thập phân. Ví dụ: -1.12345 2.12345

Nếu vô nghiệm in ra chuỗi VN

Nếu vô số nghiệm in ra chuỗi VSN

Constraints: các hệ số và các nghiệm kiểu float

**For example:**

| **Input** | **Result** |
| --- | --- |
| -425.000000 -54.000000 -236.000000  -337.000000 143.000000 -306.000000 | 0.63657 -0.63969 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<math.h>

int main(){

float a1,b1,c1,a2,b2,c2;

scanf("%f%f%f", &a1, &b1, &c1);

scanf("%f%f%f", &a2, &b2, &c2);

float d,dx,dy;

d=a1\*b2-b1\*a2;

dx=c1\*b2-b1\*c2;

dy=a1\*c2-c1\*a2;

if(d==0){

if((a1/a2)==(b1/b2)&&(b1/b2)==(c1/c2)&&(c1/c2)==(a1/a2))

printf("VSN");

else printf("VN");

}else printf("%.5f %.5f", dx/d, dy/d);

}

C4 Question text

Viết chương trình nhập vào 2 số nguyên a, b. Tính tổng các số lẻ trong đoạn [a, b]

Input:

a b trên cùng một dòng, cách nhau dấu cách.

Ví dụ: 5 12 hoặc 30 -8

Output:

Tổng các số lẻ tìm được. Ví dụ: 30

Constrains:

các biến kiểu nguyên

**For example:**

| **Input** | **Result** |
| --- | --- |
| 9 1 | 25 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

int main(){

int a,b;

scanf("%d%d",&a,&b);

int i, s=0;

if(a<b){

for(i=a;i<=b;i++){

if(i%2!=0)

s+=i;

}

printf("%d", s);

}else{

for(i=b;i<=a;i++){

if(i%2!=0)

s+=i;

}

printf("%d", s);

}

}

C5 Question text

Nhập vào 3 điểm (x1, y1), (x2, y2), (x3, y3). Với xi, yi là các số thực. Hãy kiểm tra xem 3 điểm trên có thẳng hàng hay không?

Input:

Dòng 1: x1 y1 cách nhau dấu cách

Dòng 2: x2 y2 cách nhau dấu cách

Dòng 3: x3 y3 cách nhau dấu cách

Output:

Nếu 3 điểm thẳng hàng thì in ra chữ YES; ngược lại in ra chữ NO

Constraints:

xi, yi kiểu float

**For example:**

| **Input** | **Result** |
| --- | --- |
| 219.000000 892.000000  -664.000000 244.000000  -781.000000 194.000000 | NO |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<math.h>

int main(){

float x1,y1,x2,y2,x3,y3;

scanf("%f %f",&x1,&y1);

scanf("%f %f",&x2,&y2);

scanf("%f %f",&x3,&y3);

float a,b,c;

a=sqrt((x1-x2)\*(x1-x2)+(y1-y2)\*(y1-y2));

b=sqrt((x1-x3)\*(x1-x3)+(y1-y3)\*(y1-y3));

c=sqrt((x2-x3)\*(x2-x3)+(y2-y3)\*(y2-y3));

if(a+b==c||b+c==a||c+a==b)

printf("YES");

else

printf("NO");

}

Question **1**

Correct

Marked out of 1.00

Flag question

Question text

Viết hàm tìm ước số chung lớn nhất và bội số chung nhỏ nhất của 2 số nguyên dương a, b. Sau đó, viết chương trình nhập vào 2 số nguyên x, y. Kiểm tra xem chúng có nguyên dương không. Nếu có thì in ra usc lớn nhất và bội số chung nhỏ nhất của chúng. Nếu không thì in ra thông báo "Du lieu sai".

Hàm được xây dựng phải có định dạng như sau:

void USC\_BSC(int a, int b, int \*uc, int \*bc){

//nội dung hàm

}

Input: x, y là các số nguyên. Ví dụ: -4 9 hoặc 19 2

Output:

Nếu đầu vào thỏa mãn yêu cầu thì in ra ước số chung nhỏ nhất và bội số chung lớn nhất của m, n trên 2 dòng theo định dạng sau:

Uoc so chung lon nhat cua x va y la m

Boi so chung nho nhat cua x va y la n

Nếu đầu vào không thỏa mãn yêu cầu thì in ra thông báo "Du lieu sai"

Chú ý: x, y là các số nhập vào, m, n là các kết quả tìm được.

Ví dụ, với dữ liệu vào là 7 và 5 thì kết quả in ra là.

Uoc so chung lon nhat cua 7 va 5 la 1

Boi so chung nho nhat cua 7 va 5 la 35

Constrains: x, y kiểu int

**For example:**

| **Input** | **Result** |
| --- | --- |
| 1334 1500 | Uoc so chung lon nhat cua 1334 va 1500 la 2  Boi so chung nho nhat cua 1334 va 1500 la 1000500 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

int UCLN(int a, int b){

return (a%b)?UCLN(b,a%b):b;

}

int BCNN(int a, int b){

return (a\*b)/UCLN(a,b);

}

void USC\_BSC(int a, int b, int \*uc, int \*bc){

\*uc=UCLN(a,b);

\*bc=BCNN(a,b);

}

int main(){

int a,b;

scanf("%d %d",&a,&b);

int uc, bc;

if(a>=0&&b>=0){

USC\_BSC(a,b,&uc,&bc);

printf("Uoc so chung lon nhat cua %d va %d la %d\n",a,b, uc);

printf("Boi so chung nho nhat cua %d va %d la %d", a,b,bc);

}

else

printf("Du lieu sai");

}

Question **2**

Correct

Marked out of 1.00

Flag question

Question text

Viết hàm đổi số nguyên dương n từ thập phân sang hệ nhị phân. Chuỗi nhị phân nằm trên cùng một dòng, cách nhau dấu cách. Ví dụ: 1 0 0 1

Viết chương trình nhập hai số nguyên dương a, b từ bàn phím. Sử dụng hàm trên để đổi số các số nguyên dương từ a đến b sang hệ nhị phân. Trong trường hợp b <= a thì đổi các số nguyên dương từ b đến a sang hệ nhị phân.

Input: a, b trên cùng một dòng, cách nhau dấu cách.

Output:  Mỗi dòng thể hiện như sau (ví dụ với a=93, b=95)

93: 1 0 1 1 1 0 1

94: 1 0 1 1 1 1 0

95: 1 0 1 1 1 1 1

Constrains: các số kiểu int

**For example:**

| **Input** | **Result** |
| --- | --- |
| 104 92 | 92: 1 0 1 1 1 0 0  93: 1 0 1 1 1 0 1  94: 1 0 1 1 1 1 0  95: 1 0 1 1 1 1 1  96: 1 1 0 0 0 0 0  97: 1 1 0 0 0 0 1  98: 1 1 0 0 0 1 0  99: 1 1 0 0 0 1 1  100: 1 1 0 0 1 0 0  101: 1 1 0 0 1 0 1  102: 1 1 0 0 1 1 0  103: 1 1 0 0 1 1 1  104: 1 1 0 1 0 0 0 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void \_10To2(int n){

if(n<=0) return;

\_10To2(n/2);

printf("%d ", n%2);

}

void In(int a, int b){

for(int i=a;i<=b;i++){

printf("%d: ", i);

\_10To2(i);

printf("\n");

}}

int main(){

int a,b;

scanf("%d %d", &a, &b);

if(a<b) In(a,b);

else In(b,a);

return 0;

}

Question **3**

Correct

Marked out of 1.00

Flag question

Question text

Viết hàm tính chu vi và diện tích tam giác khi biết ba cạnh. Nếu 3 số a, b, c không thỏa mãn điều kiện 3 cạnh của tam giác thì chu vi và diện tích nhận giá trị -1.000.

Viết chương trình nhập vào (a1, b1, c1) và (a2, b2, c2) là các số thực. Sử dụng hàm ở trên để tính chu vi và diện tích tam giác tương ứng.

Input:

(a1, b1, c1) và (a2, b2, c2) cách nhau dấu cách và trên 2 dòng khác nhau

Output:

chu vi và diện tích của tam giác cách nhau dấu cách, có độ chính xác 3 chữ số thập phân. Tam giác nào có diện tích lớn hơn thì ở dòng trên.

Constrains: các số thực, độ chính xác 3 chữ số thập phân

**For example:**

| **Input** | **Result** |
| --- | --- |
| -76.000 -83.000 -91.000  397.000 436.000 241.000 | 1074.000 47408.660  -1.000 -1.000 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<math.h>

float CV(float a, float b, float c){

if(a+b<c||a+c<b||c+b<a||a<0||b<0||c<0)

return -1;

return a+b+c;

}

float DT(float a, float b, float c){

if(a+b<c||a+c<b||c+b<a||a<0||b<0||c<0)

return -1;

float p=CV(a,b,c)/2;

return sqrt(p\*(p-a)\*(p-b)\*(p-c));

}

int main(){

float a1,b1,c1,a2,b2,c2;

scanf("%f%f%f", &a1,&b1,&c1);

scanf("%f%f%f", &a2,&b2,&c2);

float p1,p2,s1,s2;

p1=CV(a1,b1,c1);

p2=CV(a2,b2,c2);

s1=DT(a1,b1,c1);

s2=DT(a2,b2,c2);

if(s1>s2){

printf("%.3f %.3f\n", p1, s1);

printf("%.3f %.3f", p2, s2);

}else{

printf("%.3f %.3f\n", p2, s2);

printf("%.3f %.3f", p1, s1);

}

}

Question **4**

Correct

Marked out of 1.00

Flag question

Question text

Viết hàm kiểm tra một số nguyên có là số nguyên tố không. Biết số nguyên tố là số nguyên dương >=2 và chỉ chia hết cho 1 và chính nó.

Viết chương trình nhập vào 2 số nguyên a, b. In ra các số nguyên tố trong đoạn [a, b], nếu a<=b, hoặc ngược lại. Nếu trong đoạn này không có số nguyên tố thì in ra thông báo "Khong co".

Input: a, b là số nguyên cách nhau dấu cách.

Output: các số nguyên tố tìm được viết trên một dòng, cách nhau dấu cách. Hoặc thông báo "Khong co".

Constrains: các số kiểu int

**For example:**

| **Input** | **Result** |
| --- | --- |
| -9 -33 | Khong co |
| 18 28 | 19 23 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<math.h>

int soNguyenTo(int n){

int i;

if(n<2) return 0;

for(i=2;i<=sqrt(n);i++)

if(n%i==0) return 0;

return 1;

}

void in(int a, int b){

int i,dem=0;

for(i=a;i<=b;i++){

if(soNguyenTo(i)){

printf("%d ",i);

dem++;

}}

if(dem==0) printf("Khong co");

}

int main(){

int a,b;

scanf("%d%d",&a,&b);

if(a<=b)

in(a,b);

else

in(b,a);

}

Question **5**

Correct

Marked out of 1.00

Flag question

Question text

Viết hàm nhận đầu vào là số nguyên h khác 0. In ra tam giác cân với chiều cao |h| bằng các ký tự \* theo quy tắc sau. Nếu h>0 thì đáy của tam giác ở phía dưới. Nếu h<0 thì đáy của tam giác ở phía trên.

Ví dụ: h=-5 thì tam giác cần vẽ là

 \*\*\*\*\*\*\*\*\*

  \*\*\*\*\*\*\*

   \*\*\*\*\*

    \*\*\*

     \*

Với h=5 thì tam giác cần vẽ là

    \*

   \*\*\*

  \*\*\*\*\*

 \*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

Viết chương trình nhập vào 3 số nguyên x, y, z. Dùng hàm trên để vẽ các tam giác với chiều cao tương ứng với |x|, |y|, |z|.

Input: 3 số nguyên x, y, z cách nhau dấu cách.

Output:

Dòng 1 in số nguyên x

Từ dòng 2 in ra tam giác với chiều cao |x|

Tiếp theo đến y, z

Constrains: các dữ liệu là kiểu số nguyên

**For example:**

| **Input** | **Result** |
| --- | --- |
| 6 -8 1 | 6  \*  \*\*\*  \*\*\*\*\*  \*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*  -8  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*  \*\*\*\*\*  \*\*\*  \*  1  \* |

Answer:(penalty regime: 0 %

#include<stdio.h>

void tgXuoi(int n){

int i,j,t=n-1;

for(i=1;i<=n;i++){

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

printf(" ");

for(j=1;j<=2\*i-1;j++)

printf("\*");

printf("\n");

t--;

}}

void tgNguoc(int n){

int i,j,t=0;

for(i=n;i>=1;i--){

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

printf(" ");

for(j=1;j<=2\*i-1;j++)

printf("\*");

printf("\n");

t++;

}}

void in(int n){

printf("%d\n",n);

if(n>0)

tgXuoi(n);

else

tgNguoc(n\*(-1));

}

int main(){

int x,y,z;

scanf("%d%d%d",&x,&y,&z);

in(x);

in(y);

in(z);

}

Question **6**

Correct

Marked out of 1.00

Flag question

Question text

Viết hàm giải phương trình bậc nhất ax+b=0, hàm giải phương trình bậc 2 ax2+bx+c=0. Sau đó, sử dụng các hàm trên để:

Viết chương trình giải phương trình bậc 2 ax2+bx+c =0. Với hệ số a, b, c được nhập từ bàn phím.

Yêu cầu: Xét tất cả các trường hợp.

Input:

a, b, c là các số thực được nhập từ bàn phím

Output:

+ Nếu vô nghiệm ghi VN

+ Nếu vô số nghiệm ghi VSN

+ Nếu có 1 nghiệm ghi giá trị nghiệm với độ chính xác 3 chữ số thập phân

+ Nếu có 2 nghiệm ghi giá trị nghiệm với độ chính xác 3 chữ số thập phân, các nghiệm cách nhau dấu cách. Nghiệm nhỏ hơn đứng trước.

Ví dụ:

Input: 3 5 -8

Output:

-2.670 1.000

Constrains:

a, b, c là các số thực

**For example:**

| **Input** | **Result** |
| --- | --- |
| -20.000 -1.000 4.000 | -0.473 0.423 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<math.h>

void PTBN(float a, float b){

if(a==0){

if(b==0)

printf("VSN");

else

printf("VN");

}

else

printf("%.3f",-b/a);

}

void PTBH(float a, float b, float c){

if(a==0){

if(b==0){

if(c==0)

printf("VSN");

else

printf("VN");

}

else

printf("%.3f",-c/b);

}else{

float D;

D=b\*b-4\*a\*c;

if(D>0){

float x1,x2;

x1=(-b-sqrt(D))/(2\*a);

x2=(-b+sqrt(D))/(2\*a);

if(x1<x2)

printf("%.3f %.3f",x1,x2);

else

printf("%.3f %.3f",x2,x1);

}

else if(D==0)

printf("%.3f", -b/(2\*a));

else

printf("VN");

}

}

int main(){

float a,b,c;

scanf("%f%f%f",&a,&b,&c);

PTBH(a,b,c);

}

Question **1**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào 2 ma trận X (có n1 hàng, m1 cột) và Y (có n2 hàng, m2 cột) các số nguyên. Tính tổng 2 ma trận đó và in kết quả ra màn hình. Nếu không tính được tổng 2 ma trận thì in ra thông báo "Du lieu vao sai"

Input:

+ Dòng thứ nhất nhập vào n1 và m1 là số hàng và số cột của ma trận X.

+ Dòng tiếp theo nhập vào các phần tử của ma trận X gồm n1 hàng, m1 cột

+ Dòng tiếp theo nhập vào n2 và m2 là số hàng và số cột của ma trận Y.

+ Dòng tiếp theo nhập vào các phần tử của ma trận Y gồm n2 hàng, m2 cột

Output:

+ Nếu tính được tổng:

- Dòng thứ nhất in ra thông báo "Ma tran tong"

- Dòng thứ hai in ra số hàng và số cột cách nhau dấu cách

- Các dòng tiếp theo in ra các phần tử của ma trận cách nhau dấu cách.

+ Nếu không tính được tổng:

- In ra thông báo "Du lieu vao sai"

Constranins: 1<=n, m<=100, các phần tử trong ma trận là các số nguyên.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 2 8  9 0 19 24 3 8 12 14  5 20 6 2 11 16 20 17  2 8  2 11 16 4 2 3 17 7  21 16 18 20 22 1 21 13 | Ma tran tong  11 11 35 28 5 11 29 21  26 36 24 22 33 17 41 30 |
| 5 6  10 6 16 1 7 1  12 17 21 10 16 23  12 9 9 4 22 7  23 0 13 23 17 4  9 5 8 0 23 4  6 3  23 6 0  13 16 5  5 17 12  11 2 24  3 2 18  21 23 15 | Du lieu vao sai |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(int a[][100], int n, int m)

{

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

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

{

scanf("%d",&a[i][j]);

}

}

void xuat(int a[][100], int n, int m)

{

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

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

printf("%d ",a[i][j]);

printf("\n");

}

}

void tong(int a[][100], int b[][100], int c[][100], int n, int m)

{

int i,j;

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

{

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

{

c[i][j] =a[i][j]+b[i][j];

printf("%d ",c[i][j]);

}

printf("\n");

}

}

int main() {

int a[100][100], na, ma;

int b[100][100], nb, mb;

int c[100][100] ;

scanf("%d%d",&na,&ma);

if(na<=0 || ma<=0)

{

printf("Du lieu vao sai");

return 1;

}

nhap(a,na,ma);

scanf("%d%d",&nb,&mb);

if(nb<=0 || mb<=0)

{

printf("Du lieu vao sai");

return 1;

}

nhap(b,nb,mb);

if(na==nb && ma==mb)

{

printf("Ma tran tong\n");

tong(a,b,c,na,ma);

}

else

{

printf("Du lieu vao sai");

}

}

Question **2**

Correct

Marked out of 1.00

Flag question

Question text

Viết các hàm nhập vào mảng gồm n số thực, tìm vị trí khóa key trong mảng. Nếu có nhiều hơn một phần bằng key thì lấy vị trí nhỏ nhất. Nếu không có khóa key trong mảng thì hàm trả về giá trị -1.

Sử dụng các hàm ở trên viết chương trình nhập một mảng các số thực. Sau đó nhập vào một số thực X, in ra vị trí của X trong mảng nếu tìm thấy. Ngược lại in ra thông báo " X not found".

Input:

+ Dòng thứ nhất nhập vào số nguyên N, 1 <= N <= 1000

+ Dòng thứ hai nhập vào N số thực cách nhau dấu cách, độ chính xác 3 chữ số thập phân.

+ Dòng thứ ba nhập vào số thực X.

Output:

+ In ra thông báo "Phan tu X co vi tri y" hoặc "X not found"

Trong đó: X là khóa cần tìm, y là vị trí của X trong mảng.

Constrains: Các phần tử trong mảng là các số thực, độ chính xác 3 chữ số thập phân.

Examples:

+ Input:

5 //N

302.000 744.000 -302.000 744.000 32.000 //Mảng

320.000 //khóa X

+ Output: 320.000 not found

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  302.000 744.000 -302.000 744.000 32.000  32.000 | Phan tu 32.000 co vi tri 4 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void Nhap(float s[], int n){

int i;

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

scanf("%f", &s[i]);

}

}

void Xuat(float s[], int n, float key){

int i, dem=0;

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

if(s[i]==key){

dem++;

printf("Phan tu %.3f co vi tri %d", key, i);

break;

}

}

if(dem==0){

printf("%.3f not found", key);

}

}

int main(){

int n;

scanf("%d", &n);

float s[n];

Nhap(s,n);

float x;

scanf("%f", &x);

Xuat(s,n,x);

}

Question **3**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một mảng các số nguyên. Sắp xếp mảng đó tăng dần.

Input:

+ Dòng thứ nhất nhập vào số nguyên N, 1 <= N <= 1000

+ Dòng thứ hai nhập vào N số nguyên cách nhau dấu cách.

Output:

+ Dòng thứ nhất in ra số phần tử của mảng (N)

+ Dòng thứ hai in ra các phần tử của mảng (đã được sắp xếp tăng dần) cách nhau dấu cách.

Constrains:

+ Các phần tử trong mảng là các số nguyên

**For example:**

| **Input** | **Result** |
| --- | --- |
| 11  169 224 228 108 212 214 205 145 77 211 241 | 11  77 108 145 169 205 211 212 214 224 228 241 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(int a[], int n)

{

int i;

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

scanf("%d",&a[i]);

}

void sapXep(int a[], int n){

int i,j,tg;

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

for(j=i+1;j<n;j++)

if(a[i]>a[j]){

tg=a[i];

a[i]=a[j];

a[j]=tg;

}

printf("%d ",a[i]);

}

}

int main(){

int a[1000];

int n;

scanf("%d",&n);

nhap(a,n);

printf("%d\n",n);

sapXep(a,n);

}

Question **4**

Correct

Marked out of 1.00

Flag question

Question text

Viết hàm nhập vào mảng gồm n số thực, hàm in mảng đó ra màn hình.

Viết chương trình nhập và in ra một mảng các số thực có sử dụng hàm ở trên.

Input:

Dòng thứ nhất nhập vào số nguyên N, 1 <= N <= 1000

Dòng thứ hai nhập vào N số thực cách nhau dấu cách, độ chính xác 3 chữ số thập phân.

Output:

Dòng thứ nhất in ra thông báo "Mang ban dau"

Dòng thứ 2 in ra số phần tử của mảng (N)

Dòng thứ 3 in ra các phần tử của mảng cách nhau dấu cách, độ chính xác 3 chữ số thập phân.

Constrains:

N kiểu số nguyên

Mảng kiểu số thực

**For example:**

| **Input** | **Result** |
| --- | --- |
| 7  756.000 165.000 263.000 -33.000 548.000 689.000 350.000 | Mang ban dau  7  756.000 165.000 263.000 -33.000 548.000 689.000 350.000 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(float a[], int n){

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

scanf("%f",&a[i]);

}

void xuat(float a[], int n){

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

printf("%.3f ",a[i]);

}

int main(){

float a[1000];

int n;

scanf("%d", &n);

nhap(a,n);

printf("Mang ban dau\n");

printf("%d\n",n);

xuat(a,n);

}

Question **5**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một mảng các số nguyên. Tìm vị trí và giá trị của phần tử chẵn lớn nhất trong mảng. Nếu có nhiều số thỏa mãn thì kết quả là số có số thứ tự nhỏ nhất.

Input: Dòng thứ nhất nhập vào số nguyên N, 1 <= N <= 1000

Dòng thứ hai nhập vào mảng gồm N số nguyên cách nhau dấu cách.

Output:

+ In ra thông báo "So chan lon nhat co vi tri X gia tri Y"//X là vị trí, Y là giá trị của phần tử tìm được.

+ Nếu trong mảng không có số chẵn thì thông báo "Khong co so chan trong mang".

Constrains: Các phần tử trong mảng là các số nguyên.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 13  163 195 75 79 148 148 237 41 175 147 24 122 6 | So chan lon nhat co vi tri 4 gia tri 148 |
| 11  881 2041 1671 2011 465 2483 141 2421 2371 1741 131 | Khong co so chan trong mang |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void Nhap(int s[], int n){

int i;

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

scanf("%d", &s[i]);

}

}

void Xuat(int s[], int n){

int i, vt, dem=0;

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

if(s[i]%2==0){

dem++;

vt=i;

break;

}

}

if(dem==0){

printf("Khong co so chan trong mang");

return;

}

for(i=vt; i<n; i++){

if(s[i]%2==0 && s[i]>s[vt]){

vt=i;

}

}

printf("So chan lon nhat co vi tri %d gia tri %d", vt, s[vt]);

}

int main(){

int n;

scanf("%d", &n);

int s[n];

Nhap(s,n);

Xuat(s,n);

}

Question **6**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một ma trận gồm n hàng, m cột các số nguyên. Sắp xếp từng hàng của ma trận tăng dần. In kết quả ra màn hình.

Input:

- Dòng thứ nhất nhập vào n và m

- Dòng tiếp theo nhập vào các phần tử của ma trận

Output:

- Dòng thứ nhất in ra n và m cách nhau dấu cách

- Dòng tiếp theo in ra ma trận kết quả

Constrains: 1<=n, m<=100, các phần tử trong ma trận là số nguyên.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 6 6  17 23 9 3 0 9  12 2 21 12 23 7  12 2 2 13 7 6  7 24 22 12 11 20  4 5 7 24 2 11  15 23 20 22 20 24 | 6 6  0 3 9 9 17 23  2 7 12 12 21 23  2 2 6 7 12 13  7 11 12 20 22 24  2 4 5 7 11 24  15 20 20 22 23 24 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(int a[][100], int n, int m)

{

int i,j;

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

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

{

scanf("%d",&a[i][j]);

}

}

void in(int a[][100], int n, int m)

{

int i,j;

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

{

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

printf("%d ",a[i][j]);

printf("\n");

}

}

void sapXep(int a[][100], int n, int m)

{

int i,j,k,tg;

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

{

for(j=0;j<m-1;j++)

{

for(k=j+1;k<m;k++)

{

if(a[i][k]<a[i][j])

{

tg=a[i][j];

a[i][j]=a[i][k];

a[i][k]=tg;

}

}

}

}

}

int main()

{

int a[100][100];

int n,m;

scanf("%d%d",&n,&m);

if(n>=1 && m<=100)

{

nhap(a,n,m);

printf("%d %d\n",n,m);

sapXep(a,n,m);

in(a,n,m);

}

}

Question **7**

Correct

Marked out of 1.00

Flag question

Question text

Viết hàm nhập vào mảng gồm n số nguyên, hàm in mảng đó ra màn hình.

Viết chương trình nhập và in ra một mảng các số nguyên có sử dụng hàm ở trên.

Input: Dòng thứ nhất nhập vào số nguyên N, 1 <= N <= 1000

Dòng thứ hai nhập vào N số nguyên cách nhau dấu cách.

Output:

Dòng thứ nhất in ra thông báo "Mang ban dau"

Dòng thứ 2 in ra số phần tử của mảng (N)

Dòng thứ 3 in ra các phần tử của mảng cách nhau dấu cách.

Constrains: Các phần tử trong mảng là các số nguyên

**For example:**

| **Input** | **Result** |
| --- | --- |
| 12  20 11 -4 11 9 24 -15 9 12 1 13 0 | Mang ban dau  12  20 11 -4 11 9 24 -15 9 12 1 13 0 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(int a[], int n){

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

scanf("%d",&a[i]);

}

void xuat(int a[], int n){

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

printf("%d ", a[i]);

}

int main(){

int a[1000];

int n;

scanf("%d", &n);

if(n>=1&&n<=1000){

nhap(a,n);

printf("Mang ban dau\n");

printf("%d\n", n);

xuat(a,n);

}

}

Question **8**

Correct

Marked out of 1.00

Flag question

Question text

Viết các hàm nhập vào mảng gồm n số thực, hàm in mảng đó ra màn hình, sắp xếp mảng số thực tăng dần.

Sử dụng các hàm ở trên viết chương trình nhập vào một mảng gồm N số thực, sắp xếp và in ra kết quả ra màn hình.

Input:

Dòng thứ nhất nhập vào số nguyên N, 1 <= N <= 1000

Dòng thứ hai nhập vào N số thực cách nhau dấu cách, độ chính xác 3 chữ số thập phân.

Output:

+ Dòng thứ nhất in ra số phần tử của mảng (N)

+ Dòng thứ hai in ra các phần tử của mảng đã được sắp xếp, cách nhau dấu cách, độ chính xác 3 chữ số thập phân.

Constrains: Các phần tử trong mảng là các số thực, độ chính xác 3 chữ số thập phân.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  302.000 744.000 -302.000 744.000 32.000 | 5  -302.000 32.000 302.000 744.000 744.000 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(float a[], int n)

{

int i;

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

scanf("%f",&a[i]);

}

void in(float a[], int n)

{

int i;

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

printf("%.3f ",a[i]);

}

void sapXep(float a[], int n)

{

int i,j,tg;

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

{

for(j=i+1;j<n;j++)

if(a[i]>a[j])

{

tg=a[i];

a[i]=a[j];

a[j]=tg;

}

printf("%.3f ",a[i]);

}

}

int main()

{

float a[1000];

int n;

scanf("%d",&n);

nhap(a,n);

printf("%d\n",n);

sapXep(a,n);

}

Question **9**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào 2 ma trận X (có n1 hàng, m1 cột) và Y (có n2 hàng, m2 cột) các số nguyên. Tính tích 2 ma trận X.Y và in kết quả ra màn hình. Nếu không tính được tích 2 ma trận thì in ra thông báo "Du lieu vao sai"

Input:

+ Dòng thứ nhất nhập vào n1 và m1 là số hàng và số cột của ma trận X.

+ Dòng tiếp theo nhập vào các phần tử của ma trận X gồm n1 hàng, m1 cột

+ Dòng tiếp theo nhập vào n2 và m2 là số hàng và số cột của ma trận Y.

+ Dòng tiếp theo nhập vào các phần tử của ma trận Y gồm n2 hàng, m2 cột

Output:

+ Nếu tính được tích:

- Dòng thứ nhất in ra thông báo "Ma tran tich"

- Các dòng tiếp theo in ra các phần tử của ma trận cách nhau dấu cách.

+ Nếu không tính được tích:

- In ra thông báo "Du lieu vao sai"

Constranins: 1<=n, m<=100, các phần tử trong ma trận là các số nguyên.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 2 8  4 0 9 4 8 8 2 4  5 5 1 7 1 1 5 2  8 8  6 1 4 2 3 2 2 1  6 8 5 7 6 1 8 9  2 7 9 5 4 3 1 2  3 3 4 1 1 3 8 7  4 2 7 7 9 3 1 9  8 6 5 0 2 8 6 0  2 4 8 6 5 0 9 0  0 6 1 3 8 9 3 4 | Ma tran tich  154 175 229 137 182 171 135 138  105 113 136 100 108 68 165 118 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(int a[][100], int n, int m)

{

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

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

{

scanf("%d",&a[i][j]);

}

}

void xuat(int a[][100], int n, int m)

{

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

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

printf("%d ",a[i][j]);

printf("\n");

}

}

void tich(int a[][100], int b[][100], int c[][100], int n, int m, int k)

{

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

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

{

for(int l=0; l<m; l++)

{

c[i][j]+=a[i][l]\*b[l][j];

}

}

}

int main() {

int a[100][100], na, ma;

int b[100][100], nb, mb;

int c[100][100] ={0};

scanf("%d%d",&na,&ma);

nhap(a,na,ma);

scanf("%d%d",&nb,&mb);

nhap(b,nb,mb);

if(ma != nb)

printf("Du lieu vao sai\n");

else

{

tich(a,b,c,na,ma,mb);

printf("Ma tran tich\n");

xuat(c,na,mb);

}

}

Question **10**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một mảng các số nguyên. Tìm vị trí và giá trị của phần tử lẻ nhỏ nhất trong mảng. Nếu có nhiều số thỏa mãn thì kết quả là số có số thứ tự nhỏ nhất.

Input: Dòng thứ nhất nhập vào số nguyên N, 1 <= N <= 1000

Dòng thứ hai nhập vào mảng gồm N số nguyên cách nhau dấu cách.

Output:

+ In ra thông báo "So le nho nhat co vi tri X gia tri Y"//X là vị trí, Y là giá trị của phần tử tìm được.

+ Nếu trong mảng không có số lẻ thì thông báo "Khong co so le trong mang".

Constrains: Các phần tử trong mảng là các số nguyên.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 10  171 90 222 241 139 86 157 118 184 206 | So le nho nhat co vi tri 4 gia tri 139 |
| 11  88 204 16 20 46 248 14 242 22 174 132 | Khong co so le trong mang |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(int a[], int n ){

int i;

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

scanf("%d",&a[i]);

}

void tMin(int a[], int n){

int i,vt,dem=0;

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

if(a[i]%2==1){

dem++;

vt=i;

}

if(dem==0)

printf("Khong co so le trong mang");

else{

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

if(a[i]%2==1 && a[i]<a[vt]) vt=i;

}

printf("So le nho nhat co vi tri %d gia tri %d",vt,a[vt]);

}

}

int main(){

int a[1000];

int n;

scanf("%d",&n);

nhap(a,n);

tMin(a,n);

}

Question **11**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập và in ra ma trận gồm n hàng, m cột các số nguyên.

Input: Dòng thứ nhất nhập vào n và m là số hàng và số cột của ma trận.

Dòng tiếp theo nhập vào các phần tử của ma trận gồm n hàng, m cột

Output:

Dòng thứ nhất in ra thông báo "Ma tran ban dau"

Dòng thứ hai in ra số hàng và số cột cách nhau dấu cách

Các dòng tiếp theo in ra các phần tử của ma trận cách nhau dấu cách.

Constranins: 1<=n, m<=100, các phần tử trong ma trận là các số nguyên.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 6 18  13 14 2 10 21 18 21 19 2 1 4 20 20 10 19 14 7 19  15 19 8 0 21 5 17 5 13 5 18 15 4 9 12 23 20 24  24 5 22 18 7 0 17 12 23 15 19 13 21 13 17 3 8 24  1 8 0 6 9 20 18 16 23 23 12 10 6 5 22 21 6 1  21 24 8 19 20 15 18 17 8 1 6 3 5 7 11 24 19 13  6 2 2 4 22 3 5 14 14 22 16 18 1 12 7 9 20 24 | Ma tran ban dau  6 18  13 14 2 10 21 18 21 19 2 1 4 20 20 10 19 14 7 19  15 19 8 0 21 5 17 5 13 5 18 15 4 9 12 23 20 24  24 5 22 18 7 0 17 12 23 15 19 13 21 13 17 3 8 24  1 8 0 6 9 20 18 16 23 23 12 10 6 5 22 21 6 1  21 24 8 19 20 15 18 17 8 1 6 3 5 7 11 24 19 13  6 2 2 4 22 3 5 14 14 22 16 18 1 12 7 9 20 24 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(int a[][100], int n, int m){

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

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

scanf("%d",&a[i][j]);

}

void xuat(int a[][100], int n, int m){

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

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

printf("%d ",a[i][j]);

printf("\n");}

}

int main(){

int a[100][100];

int n,m;

scanf("%d%d",&n,&m);

nhap(a,n,m);

printf("Ma tran ban dau\n");

printf("%d %d\n", n, m);

xuat(a,n,m);

}

Question **12**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một ma trận gồm n hàng, m cột các số thực. In ra vị trí của phần tử lớn nhất trong ma trận. Nếu có nhiều phần tử lớn nhất thì in ra phần tử có chỉ số hàng, chỉ số cột nhỏ nhất.

Input:

- Dòng thứ nhất nhập vào n và m

- Dòng tiếp theo nhập vào các phần tử của ma trận

Output: In ra thông báo "Phan tu lon nhat co chi so hang x, chi so cot y, gia tri Z"

Trong đó: x là chỉ số hàng, y là chỉ số cột và Z là giá trị của phần tử lớn nhất tìm được.

Constrains: 1<=n, m<=100, các phần tử trong ma trận là số thực có 2 chữ số thập phân.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 6 4  7.67 -2.33 -1.33 -6.67  2.00 9.00 -9.67 -8.33  7.00 11.00 -6.67 -9.33  2.67 4.67 -2.67 6.00  11.00 -5.33 -6.00 8.67  -7.67 -1.67 -5.00 11.00 | Phan tu lon nhat co chi so hang 2, chi so cot 1, gia tri 11.00 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(float a[][100], int sh, int sc)

{

int i,j;

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

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

scanf("%f",&a[i][j]);

}

void tMax(float a[][100], int sh, int sc)

{

int i,j,vth=0,vtc=0;

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

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

if(a[i][j]>a[vth][vtc])

{

vth=i;

vtc=j;

}

printf("Phan tu lon nhat co chi so hang %d, chi so cot %d, gia tri %.2f",vth,vtc,a[vth][vtc]);

}

int main()

{

float a[100][100];

int n,m;

scanf("%d%d",&n,&m);

nhap(a,n,m);

tMax(a,n,m);

}

Question **13**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một mảng các số nguyên. Sắp xếp tăng dần các phần tử lẻ trong mảng. Các phần tử chẵn không thay đổi vị trí ban đầu.

Input:

+ Dòng thứ nhất nhập vào số nguyên N, 1 <= N <= 1000

+ Dòng thứ hai nhập vào N số nguyên cách nhau dấu cách.

Output:

+ Dòng thứ nhất in ra số phần tử của mảng (N)

+ Dòng thứ hai in ra các phần tử của mảng kết quả cách nhau dấu cách.

Constrains:

+ Các phần tử trong mảng là các số nguyên

**For example:**

| **Input** | **Result** |
| --- | --- |
| 11  169 228 108 212 214 205 145 31 77 211 241 | 11  31 228 108 212 214 77 145 169 205 211 241 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(int a[], int n){

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

scanf("%d",&a[i]);

}

void xuat(int a[], int n){

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

printf("%d ", a[i]);

}

void hoanvi(int &b,int &c){

int tg=b;

b=c;

c=tg;

}

void sapxep(int a[], int n){

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

for(int j=i+1;j<n;j++){

if(a[i]>a[j] && a[i]%2!=0 && a[j]%2!=0){

hoanvi(a[i],a[j]);

}

}

}

int main()

{

int a[100];

int n;

scanf("%d", &n);

if(n>=1 && n<=1000){

nhap(a,n);

printf("%d\n", n);

sapxep(a,n);

xuat(a,n);

}

}

Question **14**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một mảng các số nguyên. Sắp xếp tăng dần các phần tử chẵn trong mảng. Các phần tử lẻ không thay đổi vị trí ban đầu.

Input:

+ Dòng thứ nhất nhập vào số nguyên N, 1 <= N <= 1000

+ Dòng thứ hai nhập vào N số nguyên cách nhau dấu cách.

Output:

+ Dòng thứ nhất in ra số phần tử của mảng (N)

+ Dòng thứ hai in ra các phần tử của mảng kết quả cách nhau dấu cách.

Constrains:

+ Các phần tử trong mảng là các số nguyên

**For example:**

| **Input** | **Result** |
| --- | --- |
| 11  169 224 228 108 212 205 214 145 77 211 241 | 11  169 108 212 214 224 205 228 145 77 211 241 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(int a[], int n){

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

scanf("%d",&a[i]);

}

void xuat(int a[], int n){

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

printf("%d ", a[i]);

}

void hoanvi(int &c,int &d)

{

int tg1=c;

c=d;

d=tg1;

}

void sapxep(int a[], int n){

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

for(int j=i+1;j<n;j++){

if(a[i]>a[j] && a[i]%2==0 && a[j]%2==0){

hoanvi(a[i],a[j]);

}

}

}

int main()

{

int a[100];

int n;

scanf("%d", &n);

if(n>=1 && n<=1000){

nhap(a,n);

printf("%d\n", n);

sapxep(a,n);

xuat(a,n);

}

}

Question **15**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một ma trận gồm n hàng, m cột các số thực. In ra vị trí của phần tử nhỏ nhất trong ma trận. Nếu có nhiều phần tử nhỏ nhất thì in ra phần tử có chỉ số hàng, chỉ số cột nhỏ nhất.

Input:

- Dòng thứ nhất nhập vào n và m

- Dòng tiếp theo nhập vào các phần tử của ma trận

Output: In ra 3 số x y Z cách nhau dấu cách.

Trong đó: x là chỉ số hàng, y là chỉ số cột và Z là giá trị của phần tử nhỏ nhất tìm được.

Constrains: 1<=n, m<=100, các phần tử trong ma trận là số thực có 2 chữ số thập phân.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5 4  2.67 -4.33 -1.33 -10.00  -3.67 -4.67 -10.00 10.67  3.67 10.00 -1.33 -4.00  3.67 1.00 -6.67 -5.00  7.00 -4.00 -0.33 -0.67 | 0 3 -10.00 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(float a[][100], int sh, int sc)

{

int i,j;

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

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

scanf("%f",&a[i][j]);

}

void tMin(float a[][100], int sh, int sc)

{

int i,j,vth=0,vtc=0;

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

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

if(a[i][j]<a[vth][vtc])

{

vth=i;

vtc=j;

}

printf("%d %d %.2f",vth,vtc,a[vth][vtc]);

}

int main()

{

float a[100][100];

int n,m;

scanf("%d%d",&n,&m);

nhap(a,n,m);

tMin(a,n,m);

}

Question **16**

Correct

Marked out of 1.00

Flag question

Question text

Viết các hàm nhập vào mảng gồm n số thực, hàm tìm vị trí phần tử lớn nhất trong mảng. Nếu có nhiều hơn một phần tử lớn nhất thì lấy phần tử có vị trí nhỏ nhất.

Sử dụng các hàm ở trên viết chương trình nhập một mảng các số thực. Sau đó in ra vị trí và giá trị phần tử lớn nhất trong mảng.

Input:

+ Dòng thứ nhất nhập vào số nguyên N, 1 <= N <= 1000

+ Dòng thứ hai nhập vào N số thực cách nhau dấu cách, độ chính xác 3 chữ số thập phân.

Output:

+ In ra thông báo "Phan tu lon nhat co vi tri x, co gia tri y"

Trong đó: x, y là vị trí và giá trị của phần tử lớn nhất trong mảng.

Constrains: Các phần tử trong mảng là các số thực, độ chính xác 3 chữ số thập phân.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  302.000 744.000 -302.000 744.000 32.000 | Phan tu lon nhat co vi tri 1, co gia tri 744.000 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(float a[], int n)

{

int i;

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

scanf("%f",&a[i]);

}

void tMax(float a[], int n)

{

int i,vt=0;

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

if(a[i]>a[vt])

vt=i;

printf("Phan tu lon nhat co vi tri %d, co gia tri %.3f",vt,a[vt]);

}

int main()

{

float a[1000];

int n;

scanf("%d",&n);

nhap(a,n);

tMax(a,n);

}

Question **17**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một ma trận gồm n hàng, m cột các số nguyên. In ra vị trí của phần tử lớn nhất của từng hàng trong ma trận. Nếu trong hàng có nhiều phần tử lớn nhất thì in ra phần tử có chỉ số cột nhỏ nhất.

Input:

- Dòng thứ nhất nhập vào n và m

- Dòng tiếp theo nhập vào các phần tử của ma trận

Output: gồm n dòng, mỗi dòng in ra 3 giá trị x y Z cách nhau dấu cách

Trong đó: x là chỉ số hàng, y là chỉ số cột và Z là giá trị của phần tử lớn nhất tìm được.

Constrains: 1<=n, m<=100, các phần tử trong ma trận là số nguyên.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 6 21  5 0 18 19 12 19 4 7 0 9 20 22 13 11 7 4 0 23 16 19 11  6 17 9 10 2 18 23 4 6 8 7 20 18 13 13 3 5 3 2 17 5  4 20 16 10 5 10 10 16 11 24 12 13 9 3 19 4 23 0 14 14 9  20 23 15 13 17 2 17 1 11 24 12 24 0 5 0 14 8 4 0 11 16  21 23 24 0 10 24 4 4 8 18 22 18 13 15 24 15 17 4 10 10 14  24 11 14 1 9 14 3 13 7 15 22 19 11 21 13 14 1 1 15 21 23 | 0 17 23  1 6 23  2 9 24  3 9 24  4 2 24  5 0 24 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void Nhap(int s[][200], int n, int m)

{

int i, j;

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

{

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

{

scanf("%d", &s[i][j]);

}

}

}

void Xuat(int s[][200], int n, int m)

{

int i, j;

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

{

int vt=0;

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

{

if(s[i][j]>s[i][vt])

{

vt=j;

}

}

printf("%d %d %d\n", i, vt, s[i][vt]);

}

}

int main()

{

int n, m;

scanf("%d%d", &n, &m);

int s[n][200];

Nhap(s,n,m);

Xuat(s,n,m);

}

Question **18**

Correct

Marked out of 1.00

Flag question

Question text

Viết các hàm nhập vào mảng gồm n số thực, in ra mảng, sắp xếp mảng số thực tăng dần.

Viết chương trình (sử dụng các hàm trên) thực hiện nhập, sắp xếp mảng và in kết quả ra màn hình.

Input:

+ Dòng thứ nhất nhập vào số nguyên N, 1 <= N <= 1000

+ Dòng thứ hai nhập vào N số thực cách nhau dấu cách, độ chính xác 3 chữ số thập phân.

Output:

+ Dòng thứ nhất in ra thông báo "Mang sap xep"

+ Dòng thứ hai in ra các phần tử của mảng đã được sắp xếp, cách nhau dấu cách, độ chính xác 3 chữ số thập phân.

Constrains: Các phần tử trong mảng là các số thực, độ chính xác 3 chữ số thập phân.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  302.000 744.000 -302.000 744.000 32.000 | Mang sap xep  -302.000 32.000 302.000 744.000 744.000 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

void nhap(float a[], int n)

{

int i;

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

scanf("%f",&a[i]);

}

void in(float a[], int n)

{

int i;

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

printf("%f ",a[i]);

}

void sapXep(float a[], int n)

{

int i,j;

int tg;

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

{

for(j=i+1;j<n;j++)

if(a[i]>a[j])

{

tg=a[i];

a[i]=a[j];

a[j]=tg;

}

printf("%.3f ",a[i]);

}

}

int main()

{

float a[1000];

int n;

scanf("%d",&n);

nhap(a,n);

printf("Mang sap xep\n");

sapXep(a,n);

}

Question **1**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào danh sách gồm n hình tròn trong mặt phẳng. In danh sách đó ra màn hình. Biết thông tin của hình tròn gồm mã hình tròn, bán kính, tọa độ tâm (x, y).

Input:

+ Dòng thứ nhất nhập vào số nguyên dương n

+ Dòng tiếp theo nhập vào thông tin của từng hình tròn. Thứ tự nhập vào là mã, bán kính, tâm x, tâm y.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach hinh tron"

+ Dòng thứ 2 in ra số lượng các hình tròn (n)

+ Các dòng tiếp theo in ra thông tin các hình tròn. Thứ tự in ra là mã, tọa độ x, tọa độ y, bán kính. Các thông tin cách nhau dấu cách. Giá trị số thực có độ chính xác 3 chữ số thập phân.

Constrains:

+ 1<=n<=1000

+ bán kính là số thực, tọa độ tâm là các số nguyên

+ Số pi =3.14159

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  78 19.667 962 464  5 15.333 281 827  61 30.667 995 942  27 12.333 391 604 | Danh sach hinh tron  4  78 962 464 19.667  5 281 827 15.333  61 995 942 30.667  27 391 604 12.333 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

typedef struct

{

char maHT[2];

float r;

int x,y;

} htr;

void nhap(htr a[], int n)

{

int i;

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

{

scanf("%s",a[i].maHT);

scanf("%f",&a[i].r);

scanf("%d",&a[i].x);

scanf("%d",&a[i].y);

}

}

void xuat(htr a[], int n)

{

int i;

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

{

printf("%s ",a[i].maHT);

printf("%d ",a[i].x);

printf("%d ",a[i].y);

printf("%.3f \n",a[i].r);

}

}

int main()

{

htr a[1000];

int n;

scanf("%d",&n);

if(n>=1 && n<=1000)

{

nhap(a,n);

printf("Danh sach hinh tron\n");

printf("%d\n",n);

xuat(a,n);

}

}

Question **2**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n thí sinh thi đại học (n<=100). In danh sách đó ra màn hình. Biết thông tin của một thí sinh gồm: tên, số báo danh, điểm môn 1, điểm môn 2, điểm môn 3.

Input:

+ Dòng thứ nhất nhập vào n là số thí sinh

+ Dòng tiếp theo nhập vào thông tin của các thí sinh theo thứ tự tên, số báo danh, điểm môn 1, điểm môn 2, điểm môn 3.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach thi sinh"

+ Dòng thứ hai in ra số thí sinh trong danh sách

+ Các dòng tiếp theo, mỗi dòng in ra thông tin của từng thí sinh. Các thông tin cách nhau dấu cách (thứ tự: tên, số báo danh, điểm môn 1, điểm môn 2, điểm môn 3), điểm tổng in ở cuối mỗi dòng.

Constrains: 1<=n<=100, điểm các môn và tổng điểm có độ chính xác 2 chữ số thập phân, số báo danh là số nguyên dương, tên thí sinh không chứa dấu cách.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  Anh 78 10.00 1.00 0.00  Vinh 5 7.00 5.00 8.00  Nhung 61 7.00 3.00 7.00  Trang 27 2.00 7.00 7.00 | Danh sach thi sinh  4  Anh 78 10.00 1.00 0.00 11.00  Vinh 5 7.00 5.00 8.00 20.00  Nhung 61 7.00 3.00 7.00 17.00  Trang 27 2.00 7.00 7.00 16.00 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

typedef struct

{

char ten[30];

int sbd;

float diem1, diem2, diem3;

}ths;

float tong(ths x)

{

return (x.diem1+x.diem2+x.diem3);

}

void nhap(ths a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].sbd);

scanf("%f",&a[i].diem1);

scanf("%f",&a[i].diem2);

scanf("%f",&a[i].diem3);

}

}

void xuat(ths a[], int n)

{

int i;

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

printf("%s ",a[i].ten);

printf("%d ",a[i].sbd);

printf("%.2f ",a[i].diem1);

printf("%.2f ",a[i].diem2);

printf("%.2f ",a[i].diem3);

printf("%.2f \n",tong(a[i]));

}

}

int main()

{

ths a[100];

int n;

scanf("%d",&n);

if(n>=1 && n<=100)

{

nhap(a,n);

printf("Danh sach thi sinh\n");

printf("%d\n",n);

xuat(a,n);

}

}

Question **3**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n sinh viên (n<=100). In danh sách đó ra màn hình. Biết thông tin của một sinh viên gồm: tên, mã sv, điểm Toán, điểm Triết, điểm LT C

Input:

+ Dòng thứ nhất nhập vào n là số sinh viên

+ Dòng tiếp theo nhập vào thông tin của các sinh viên theo thứ tự tên, mã, điểm Toán, điểm Triết, điểm LT C.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach sinh vien"

+ Dòng thứ hai in ra số sinh viên trong danh sách

+ Các dòng tiếp theo, mỗi dòng in ra thông tin của từng sinh viên. Các thông tin cách nhau dấu cách (thứ tự tên, mã, điểm Toán, điểm Triết, điểm LT C), điểm trung bình in ở cuối mỗi dòng.

Constrains: 1<=n<=100, điểm các môn và điểm trung bình có độ chính xác 2 chữ số thập phân, mã sinh viên là các số nguyên dương, tên sinh viên không chứa dấu cách.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  Anh 78 10.00 1.00 0.00  Vinh 5 7.00 5.00 8.00  Nhung 61 7.00 3.00 7.00  Trang 27 2.00 7.00 7.00 | Danh sach sinh vien  4  Anh 78 10.00 1.00 0.00 3.67  Vinh 5 7.00 5.00 8.00 6.67  Nhung 61 7.00 3.00 7.00 5.67  Trang 27 2.00 7.00 7.00 5.33 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

typedef struct

{

char ten[30];

int MSV;

float Toan,Triet,LTC;

}SV;

float TB(SV a){

return ((a.Toan+a.Triet+a.LTC)/3);

}

void nhap(SV a[], int n){

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

scanf("%s",a[i].ten);

scanf("%d",&a[i].MSV);

scanf("%f %f %f",&a[i].Toan, &a[i].Triet, &a[i].LTC);

}

}

void xuat(SV a[], int n){

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

printf("%s ",a[i].ten);

printf("%d ",a[i].MSV);

printf("%.2f %.2f %.2f ",a[i].Toan,a[i].Triet,a[i].LTC);

printf("%.2f\n", TB(a[i]));

}

}

int main(){

SV a[100];

int n;

scanf("%d", &n);

nhap(a,n);

printf("Danh sach sinh vien\n");

printf("%d\n",n);

xuat(a,n);

}

Question **4**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n nhân viên (n<=100). In danh sách đó ra màn hình. Biết thông tin của một nhân viên gồm: tên, mã nv, hệ số lương, phụ cấp

Input:

+ Dòng thứ nhất nhập vào n là số nhân viên

+ Dòng tiếp theo nhập vào thông tin của các nhân viên theo thứ tự tên, mã, hệ số lương, phụ cấp.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach nhan vien"

+ Dòng thứ hai in ra số nhân viên trong danh sách

+ Các dòng tiếp theo, mỗi dòng in ra thông tin của từng nhân viên. Các thông tin cách nhau dấu cách, lương tháng in ở cuối mỗi dòng. Lương tháng được tính bằng hệ số lương \* 2000000 + phụ cấp.

Constrains: 1<=n<=100, mã nhân viên, phụ cấp là các số nguyên dương, tên nhân viên không chứa dấu cách, các thông tin còn lại là số thực có độ chính xác 2 chữ số thập phân.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  Anh 78 2.34 1000000  Vinh 5 3.3 500000  Nhung 61 2.67 700000  Trang 27 4.32 250000 | Danh sach nhan vien  4  Anh 78 2.34 1000000 5680000.00  Vinh 5 3.30 500000 7100000.00  Nhung 61 2.67 700000 6040000.00  Trang 27 4.32 250000 8890000.00 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

typedef struct

{

char ten[30];

int maNV;

float hsLuong;

int phuCap;

}nv;

void nhap(nv a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].maNV);

scanf("%f",&a[i].hsLuong);

scanf("%d",&a[i].phuCap);

}

}

float luong(nv x)

{

return ((x.hsLuong\*2000000)+x.phuCap);

}

void xuat(nv a[], int n)

{

int i;

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

{

printf("%s ",a[i].ten);

printf("%d ",a[i].maNV);

printf("%.2f ",a[i].hsLuong);

printf("%d ",a[i].phuCap);

printf("%.2f\n",luong(a[i]));

}

}

int main()

{

nv a[100];

int n;

scanf("%d",&n);

if(n>=1 && n<=100)

{

nhap(a,n);

printf("Danh sach nhan vien\n");

printf("%d\n",n);

xuat(a,n);

}

}

Question **1**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n sinh viên (n<=100). Sắp xếp danh sách đó theo điểm trung bình (thứ tự giảm dần) và in danh sách đã sắp ra màn hình. Biết thông tin của một sinh viên gồm: tên, mã sv, điểm Toán, điểm Triết, điểm LT C

Input:

+ Dòng thứ nhất nhập vào n là số sinh viên

+ Dòng tiếp theo nhập vào thông tin của các sinh viên theo thứ tự tên, mã, điểm Toán, điểm Triết, điểm LT C.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach sinh vien"

+ Dòng thứ hai in ra số sinh viên trong danh sách

+ Các dòng tiếp theo, mỗi dòng in ra thông tin của từng sinh viên. Các thông tin cách nhau dấu cách, điểm trung bình in ở cuối mỗi dòng.

Constrains: 1<=n<=100, điểm các môn và điểm trung bình có độ chính xác 2 chữ số thập phân, mã sinh viên là các số nguyên dương, tên sinh viên không chứa dấu cách.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  Huy 87 5.00 5.00 4.00  Lan 2 7.00 2.00 7.00  Anh 25 7.00 0.00 10.00  Van 35 0.00 10.00 9.00  Trang 76 2.00 9.00 1.00 | Danh sach sinh vien  5  Van 35 0.00 10.00 9.00 6.33  Anh 25 7.00 0.00 10.00 5.67  Lan 2 7.00 2.00 7.00 5.33  Huy 87 5.00 5.00 4.00 4.67  Trang 76 2.00 9.00 1.00 4.00 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

typedef struct

{

char ten[30];

int maSV;

float diemT,diemTR,diemLTC;

} sv;

void nhap(sv a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].maSV);

scanf("%f",&a[i].diemT);

scanf("%f",&a[i].diemTR);

scanf("%f",&a[i].diemLTC);

}

}

float dtb(sv x)

{

return ((x.diemT+x.diemTR+x.diemLTC)/3);

}

void sapXep(sv a[], int n)

{

int i,j;

sv tg;

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

for(j=i+1;j<n;j++)

if(dtb(a[i])<dtb(a[j]))

{

tg=a[i];

a[i]=a[j];

a[j]=tg;

}

printf("%s ",a[i].ten);

printf("%d ",a[i].maSV);

printf("%.2f ",a[i].diemT);

printf("%.2f ",a[i].diemTR);

printf("%.2f ",a[i].diemLTC);

printf("%.2f\n",dtb(a[i]));

}

}

int main()

{

sv a[100];

int n;

scanf("%d",&n);

if(n>=1 && n<=100)

{

nhap(a,n);

printf("Danh sach sinh vien\n");

printf("%d\n",n);

sapXep(a,n);

}

}

Question **2**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n nhân viên (n<=100). Sắp xếp danh sách nhân viên theo lương tháng giảm dần và in kết quả ra màn hình.  Biết thông tin của một nhân viên gồm: tên nhân viên, mã nv, hệ số lương, phụ cấp. Lương tháng được tính bằng hệ số lương \* 2000000 + phụ cấp.

Input:

+ Dòng thứ nhất nhập vào n là số nhân viên

+ Dòng tiếp theo nhập vào thông tin của các nhân viên theo thứ tự tên, mã, hệ số lương, phụ cấp.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach nhan vien"

+ Dòng thứ hai in ra số nhân viên trong danh sách

+ Các dòng tiếp theo, mỗi dòng in ra thông tin của từng nhân viên. Các thông tin cách nhau dấu cách, lương tháng in ở cuối mỗi dòng.

Constrains: 1<=n<=100, mã nhân viên, phụ cấp là các số nguyên dương, tên nhân viên không chứa dấu cách, các thông tin còn lại là số thực có độ chính xác 2 chữ số thập phân.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  Anh 78 2.34 1000000  Vinh 5 3.3 500000  Nhung 61 2.67 700000  Trang 27 4.32 250000 | Danh sach nhan vien  4  Trang 27 4.32 250000 8890000.00  Vinh 5 3.30 500000 7100000.00  Nhung 61 2.67 700000 6040000.00  Anh 78 2.34 1000000 5680000.00 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

typedef struct

{

char ten[30];

int maNV;

float hsLuong;

int phuCap;

}nv;

float luong(nv x)

{

return ((x.hsLuong\*2000000)+x.phuCap);

}

void nhap(nv a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].maNV);

scanf("%f",&a[i].hsLuong);

scanf("%d",&a[i].phuCap);

}

}

void sapXep(nv a[], int n)

{

int i,j;

nv tg;

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

for(j=i+1;j<n;j++)

if(luong(a[i])<luong(a[j]))

{

tg=a[i];

a[i]=a[j];

a[j]=tg;

}

printf("%s ",a[i].ten);

printf("%d ",a[i].maNV);

printf("%.2f ",a[i].hsLuong);

printf("%d ",a[i].phuCap);

printf("%.2f \n",luong(a[i]));

}}

int main()

{

nv a[100];

int n;

scanf("%d",&n);

if(n>=1 && n<=100)

{

nhap(a,n);

printf("Danh sach nhan vien\n");

printf("%d\n",n);

sapXep(a,n);

}

}

Question **3**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào danh sách gồm n hình tròn trong mặt phẳng. Nhập vào một điểm A (xa, ya). Cho biết điểm A nằm trong, nằm trên, hay nằm ngoài các hình tròn đã nhập vào. Biết thông tin của hình tròn gồm mã hình tròn, bán kính, tọa độ tâm (x, y).

Input:

+ Dòng thứ nhất nhập vào số nguyên dương n

+ Dòng tiếp theo nhập vào thông tin của từng hình tròn. Thứ tự nhập vào là mã, bán kính, tâm x, tâm y.

+ Dòng cuối cùng nhập vào xa, ya cách nhau dấu cách.

Output:

+ Gồm n dòng, mỗi dòng in ra thông báo "Diem (xa, ya) nam trong/tren/ngoai hinh tron tam (x, y) ban kinh r"

+ Trong đó: (xa, ya) là tọa độ điểm A, (x, y) và r là tọa độ tâm và bán kính các hình tròn trong danh sách.

Constrains:

+ 1<=n<=1000.

+ bán kính là số thực với độ chính xác 3 chữ số thập phân, tọa độ là các số nguyên.

+ Số pi =3.14159.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 2  33 224.667 664 141  33 9560.139 664 141  44 62 | Diem (44, 62) nam ngoai hinh tron tam (664, 141) ban kinh 224.667  Diem (44, 62) nam trong hinh tron tam (664, 141) ban kinh 9560.139 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<math.h>

typedef struct

{

int id;

float r, x, y;

} ht;

void Nhap(ht a[], int n)

{

int i;

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

{

scanf("%d", &a[i].id);

scanf("%f", &a[i].r);

scanf("%f", &a[i].x);

scanf("%f", &a[i].y);

}

}

float KC(float x, float y,float xa, float ya)

{

return sqrt(pow((x-xa),2)+pow((y-ya),2));

}

void Xuat(ht a[], int n, float xa, float ya)

{

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

{

if(KC(a[i].x,a[i].y,xa,ya)<a[i].r)

printf("Diem (%.0f, %.0f) nam trong hinh tron tam (%.0f, %.0f) ban kinh %.3f\n",

xa, ya, a[i].x, a[i].y, a[i].r);

else if(KC(a[i].x,a[i].y,xa,ya)>a[i].r)

printf("Diem (%.0f, %.0f) nam ngoai hinh tron tam (%.0f, %.0f) ban kinh %.3f\n",

xa, ya, a[i].x, a[i].y, a[i].r);

else

printf("Diem (%.0f, %.0f) nam tren hinh tron tam (%.0f, %.0f) ban kinh %.3f\n",

xa, ya, a[i].x, a[i].y, a[i].r);

}

}

int main()

{

int n;

scanf("%d", &n);

ht a[n];

float x, y;

Nhap(a,n);

scanf("%f%f",&x,&y);

Xuat(a,n,x,y);

}

Question **4**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n nhân viên (n<=100). Tìm nhân viên có lương lớn nhất. Nếu có nhiều người có lương lớn nhất bằng nhau thì kết quả là người ở đầu danh sách. Biết thông tin của một nhân viên gồm: tên, mã nv, hệ số lương, phụ cấp

Input:

+ Dòng thứ nhất nhập vào n là số nhân viên

+ Dòng tiếp theo nhập vào thông tin của các nhân viên theo thứ tự tên, mã, hệ số lương, phụ cấp.

Output:

+ Dòng thứ nhất in ra thông báo "Nhan vien co luong lon nhat"

+ Dòng thứ hai in ra thông tin của nhân viên tìm được. Các thông tin cách nhau dấu cách, lương tháng in ở cuối mỗi dòng. Lương tháng được tính bằng hệ số lương \* 2000000 + phụ cấp.

Constrains: 1<=n<=100, mã nhân viên, phụ cấp là các số nguyên dương, tên nhân viên không chứa dấu cách, các thông tin còn lại là số thực có độ chính xác 2 chữ số thập phân.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  Anh 78 2.34 1000000  Vinh 5 4.32 250000  Nhung 61 2.67 700000  Trang 27 4.32 250000 | Nhan vien co luong lon nhat  Vinh 5 4.32 250000 8890000.00 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

typedef struct

{

char ten[30];

int maNV;

float hsLuong;

int phuCap;

} nv;

void nhap(nv a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].maNV);

scanf("%f",&a[i].hsLuong);

scanf("%d",&a[i].phuCap);

}

}

float luong(nv x)

{

return ((x.hsLuong\*2000000)+x.phuCap);

}

void lMax(nv a[], int n)

{

int i;

nv max=a[0];

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

{

if(luong(max)<luong(a[i]))

{

max=a[i];

}

}

printf("%s ",max.ten);

printf("%d ",max.maNV);

printf("%.2f ",max.hsLuong);

printf("%d ",max.phuCap);

printf("%.2f ",luong(max));

}

int main()

{

nv a[100];

int n;

scanf("%d",&n);

if(n>=1 && n<=100)

{

nhap(a,n);

printf("Nhan vien co luong lon nhat\n");

lMax(a,n);

}

}

Question **5**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n sinh viên (n<=100). Tìm sinh viên có điểm trung bình thấp nhất. Nếu có nhiều người bằng điểm thì kết quả là người ở cuối danh sách. Biết thông tin của một sinh viên gồm: tên, mã sv, điểm Toán, điểm Triết, điểm LT C

Input:

+ Dòng thứ nhất nhập vào n là số sinh viên

+ Dòng tiếp theo nhập vào thông tin của các sinh viên theo thứ tự tên, mã, điểm Toán, điểm Triết, điểm LT C.

Output:

+ Dòng thứ nhất in ra thông báo "So thu tu cua sv: X "//X là số thứ tự của sv tìm dc

+ Dòng thứ hai in ra thông tin của từng sinh viên. Các thông tin cách nhau dấu cách, điểm trung bình in ở cuối mỗi dòng.

Constrains: 1<=n<=100, điểm các môn và điểm trung bình có độ chính xác 2 chữ số thập phân, mã sinh viên là các số nguyên dương, tên sinh viên không chứa dấu cách.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  Anh 78 10.00 1.00 0.00  Vinh 5 7.00 5.00 8.00  Nhung 61 10.00 1.00 0.00  Trang 27 2.00 7.00 7.00 | So thu tu cua sv: 2  Nhung 61 10.00 1.00 0.00 3.67 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<string.h>

typedef struct

{

char ten[30];

int maSV;

float diemT;

float diemTR;

float diemLTC;

}sv;

void nhap(sv a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].maSV);

scanf("%f",&a[i].diemT);

scanf("%f",&a[i].diemTR);

scanf("%f",&a[i].diemLTC);

}

}

float dtb(sv x)

{

return ((x.diemT+x.diemTR+x.diemLTC)/3);

}

void dtbMin(sv a[], int n)

{

int i,vt=0;

sv min=a[0];

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

{

if(dtb(min)>=dtb(a[i]))

{

min=a[i];

vt=i;

}

//if(strcmp(a[i].ten,a[j].ten)<0)

}

printf("So thu tu cua sv: %d\n",vt);

printf("%s ",min.ten);

printf("%d ",min.maSV);

printf("%.2f ",min.diemT);

printf("%.2f ",min.diemTR);

printf("%.2f ",min.diemLTC);

printf("%.2f ",dtb(min));

}

int main()

{

sv a[100];

int n;

scanf("%d",&n);

nhap(a,n);

dtbMin(a,n);

}

Question **6**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n thí sinh thi đại học (n<=100). In danh sách các thí sinh thi trượt ra màn hình. Biết thông tin của một thí sinh gồm: tên, số báo danh, điểm môn 1, điểm môn 2, điểm môn 3. Thí sinh thi trượt khi tổng điểm < 15.00 hoặc có một môn mà  điểm <1.00.

Input:

+ Dòng thứ nhất nhập vào n là số thí sinh

+ Dòng tiếp theo nhập vào thông tin của các thí sinh theo thứ tự tên, số báo danh, điểm môn 1, điểm môn 2, điểm môn 3.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach thi sinh thi truot"

+ Các dòng tiếp theo, mỗi dòng in ra thông tin của từng thí sinh. Các thông tin cách nhau dấu cách (thứ tự: tên, số báo danh, điểm môn 1, điểm môn 2, điểm môn 3), tổng điểm in ở cuối mỗi dòng.

+ Dòng cuối in ra thông báo "So thi sinh thi truot: X". X là số lượng thí sinh thi trượt.

Constrains: 1<=n<=100, điểm các môn và tổng điểm có độ chính xác 2 chữ số thập phân, số báo danh là số nguyên dương, tên thí sinh không chứa dấu cách.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  Huy 87 5.00 3.00 3.00  Lan 2 3.00 2.00 3.00  Anh 25 7.00 9.00 10.00  Van 35 0.00 10.00 9.00  Trang 76 9.00 9.00 9.00 | Danh sach thi sinh thi truot  Huy 87 5.00 3.00 3.00 11.00  Lan 2 3.00 2.00 3.00 8.00  Van 35 0.00 10.00 9.00 19.00  So thi sinh thi truot: 3 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

typedef struct

{

char ten[30];

int sbd;

float diem1,diem2,diem3;

}ths;

void nhap(ths a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].sbd);

scanf("%f",&a[i].diem1);

scanf("%f",&a[i].diem2);

scanf("%f",&a[i].diem3);

}

}

float tong(ths x)

{

return (x.diem1+x.diem2+x.diem3);

}

void tke(ths a[], int n)

{

int i,d=0;

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

{

if(tong(a[i])<15 || a[i].diem1<1

|| a[i].diem2<1 || a[i].diem3<1)

{

printf("%s ",a[i].ten);

printf("%d ",a[i].sbd);

printf("%.2f ",a[i].diem1);

printf("%.2f ",a[i].diem2);

printf("%.2f ",a[i].diem3);

printf("%.2f \n",tong(a[i]));

d++;

}

}

printf("So thi sinh thi truot: %d",d);

}

int main()

{

ths a[100];

int n;

scanf("%d",&n);

if(n>=1 && n<=100)

{

nhap(a,n);

printf("Danh sach thi sinh thi truot\n");

tke(a,n);

}

}

Question **7**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n sinh viên (n<=100). Sắp xếp danh sách đó theo tên (thứ tự từ a đến z) và in ra màn hình. Biết thông tin của một sinh viên gồm: tên, mã sv, điểm Toán, điểm Triết, điểm LT C

Input:

+ Dòng thứ nhất nhập vào n là số sinh viên

+ Dòng tiếp theo nhập vào thông tin của các sinh viên theo thứ tự tên, mã, điểm Toán, điểm Triết, điểm LT C.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach sinh vien"

+ Dòng thứ hai in ra số sinh viên trong danh sách

+ Các dòng tiếp theo, mỗi dòng in ra thông tin của từng sinh viên. Các thông tin cách nhau dấu cách, điểm trung bình in ở cuối mỗi dòng.

Constrains: 1<=n<=100, điểm các môn và điểm trung bình có độ chính xác 2 chữ số thập phân, mã sinh viên là các số nguyên dương, tên sinh viên không chứa dấu cách.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  Huy 87 5.00 5.00 4.00  Lan 2 7.00 2.00 7.00  Anh 25 7.00 0.00 10.00  Van 35 0.00 10.00 9.00  Trang 76 2.00 9.00 1.00 | Danh sach sinh vien  5  Anh 25 7.00 0.00 10.00 5.67  Huy 87 5.00 5.00 4.00 4.67  Lan 2 7.00 2.00 7.00 5.33  Trang 76 2.00 9.00 1.00 4.00  Van 35 0.00 10.00 9.00 6.33 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<string.h>

typedef struct

{

char ten[30];

int maSV;

float diemT,diemTR,diemLTC;

}sv;

void nhap(sv a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].maSV);

scanf("%f",&a[i].diemT);

scanf("%f",&a[i].diemTR);

scanf("%f",&a[i].diemLTC);

}

}

float dtb(sv x)

{

return ((x.diemT+x.diemTR+x.diemLTC)/3);

}

void sapXep(sv a[], int n)

{

int i,j;

sv tg;

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

{

for(j=i+1;j<n;j++)

if(strcmp(a[i].ten,a[j].ten)>0)

{

tg=a[i];

a[i]=a[j];

a[j]=tg;

}

printf("%s ",a[i].ten);

printf("%d ",a[i].maSV);

printf("%.2f ",a[i].diemT);

printf("%.2f ",a[i].diemTR);

printf("%.2f ",a[i].diemLTC);

printf("%.2f\n",dtb(a[i]));

}

}

int main()

{

sv a[100];

int n;

scanf("%d",&n);

nhap(a,n);

printf("Danh sach sinh vien\n");

printf("%d\n",n);

sapXep(a,n);

}

Question **8**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n thí sinh thi đại học (n<=100). In thông tin thí sinh thủ khoa (thi đỗ và có tổng điểm cao nhất) ra màn hình.

   +Nếu có nhiều thủ khoa thì in ra thủ khoa có số thứ tự nhỏ nhất trong danh sách sách.

   + Nếu không có thủ khoa (mọi thí sinh đều trượt) thì thông báo "Khong co ai thi do"

   + Biết thông tin của một thí sinh gồm: tên, số báo danh, điểm môn 1, điểm môn 2, điểm môn 3.

   + Thí sinh thi đỗ khi tổng điểm >= 15.00 và điểm các môn >=1.00.

Input:

+ Dòng thứ nhất nhập vào n là số thí sinh

+ Dòng tiếp theo nhập vào thông tin của các thí sinh theo thứ tự tên, số báo danh, điểm môn 1, điểm môn 2, điểm môn 3.

Output:

Nếu tìm được thủ khoa:

+ Dòng thứ nhất in ra thông báo "So thu tu cua thu khoa: X", X là số thự tự của thí sinh đỗ thủ khoa tìm được theo yêu cầu.

+ Dòng thứ hai in ra thông tin của thí sinh thủ khoa tìm được. Các thông tin cách nhau dấu cách (thứ tự: tên, số báo danh, điểm môn 1, điểm môn 2, điểm môn 3), tổng điểm in ở cuối mỗi dòng.

Nếu KHÔNG tìm được thủ khoa:

+ In ra thông báo "Khong co ai thi do"

Constrains: 1<=n<=100, điểm các môn và tổng điểm có độ chính xác 2 chữ số thập phân, số báo danh là số nguyên dương, tên thí sinh không chứa dấu cách.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  Huy 87 5.00 3.00 3.00  Trang 76 9.00 9.00 9.00  Lan 2 3.00 2.00 3.00  Anh 25 8.00 9.00 10.00  Van 35 0.00 10.00 9.00 | So thu tu cua thu khoa: 1  Trang 76 9.00 9.00 9.00 27.00 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

typedef struct

{

char ten[30];

int sbd;

float diem1,diem2,diem3;

}ths;

void nhap(ths a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].sbd);

scanf("%f",&a[i].diem1);

scanf("%f",&a[i].diem2);

scanf("%f",&a[i].diem3);

}

}

float tong(ths x)

{

return (x.diem1+x.diem2+x.diem3);

}

void thuKhoa(ths a[], int n)

{

int i;

int dem=0;

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

if(tong(a[i])<15 || a[i].diem1<1 || a[i].diem2<1 || a[i].diem3<1)

{

dem++;

}

if(dem==n)

{

printf("Khong co ai thi do");

}

else

{

int vt=0;

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

if(tong(a[vt])<tong(a[i]) && a[i].diem1>=1 && a[i].diem2>=1 && a[i].diem3>=1)

vt=i;

printf("So thu tu cua thu khoa: %d\n",vt);

printf("%s ",a[vt].ten);

printf("%d ",a[vt].sbd);

printf("%.2f ",a[vt].diem1);

printf("%.2f ",a[vt].diem2);

printf("%.2f ",a[vt].diem3);

printf("%.2f \n",tong(a[vt]));

}

}

int main()

{

ths a[100];

int n;

scanf("%d",&n);

nhap(a,n);

thuKhoa(a,n);

}

Question **9**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n nhân viên (n<=100). Tìm nhân viên có hệ số lương thấp nhất. Nếu có nhiều nhân viên có hệ số lương thấp nhất bằng nhau thì kết quả là nhân viên đầu tiên trong danh sách. Biết thông tin của một nhân viên gồm: tên, mã nv, hệ số lương, phụ cấp

Input:

+ Dòng thứ nhất nhập vào n là số nhân viên

+ Dòng tiếp theo nhập vào thông tin của các nhân viên theo thứ tự tên, mã, hệ số lương, phụ cấp.

Output:

+ Dòng thứ nhất in ra thông báo "Nhan vien co he so luong thap nhat"

+ Dòng thứ hai in ra thông tin của nhân viên tìm được. Các thông tin cách nhau dấu cách, lương tháng in ở cuối mỗi dòng. Lương tháng được tính bằng hệ số lương \* 2000000 + phụ cấp.

Constrains: 1<=n<=100, mã nhân viên, phụ cấp là các số nguyên dương, tên nhân viên không chứa dấu cách, các thông tin còn lại là số thực có độ chính xác 2 chữ số thập phân.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  Anh 78 2.34 1000000  Vinh 5 3.30 500000  Nhung 61 2.67 700000  Trang 27 2.34 250000 | Nhan vien co he so luong thap nhat  Anh 78 2.34 1000000 5680000.00 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

typedef struct

{

char ten[30];

int maNV;

float hsLuong;

int phuCap;

} nv;

float luong(nv x)

{

return (x.hsLuong\*2000000)+x.phuCap;

}

void nhap(nv a[],int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].maNV);

scanf("%f",&a[i].hsLuong);

scanf("%d",&a[i].phuCap);

}

}

void lMin(nv a[], int n)

{

int i;

nv min=a[0];

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

if(min.hsLuong>a[i].hsLuong)

{

min=a[i];

}

}

printf("%s ",min.ten);

printf("%d ",min.maNV);

printf("%.2f ",min.hsLuong);

printf("%d ",min.phuCap);

printf("%.2f",luong(min));

}

int main()

{

nv a[100];

int n;

scanf("%d",&n);

if(n>=1 && n<=100)

{

nhap(a,n);

printf("Nhan vien co he so luong thap nhat\n");

lMin(a,n);

}

}

Question **10**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n sinh viên (n<=100). In ra màn hình danh sách các sv phải học lại ít nhất 2 môn (sinh viên phải học lại khi điểm < 4.0). Biết thông tin của một sinh viên gồm: tên, mã sv, điểm Toán, điểm Triết, điểm LT C.

Input:

+ Dòng thứ nhất nhập vào n là số sinh viên

+ Dòng tiếp theo nhập vào thông tin của các sinh viên theo thứ tự tên, mã, điểm Toán, điểm Triết, điểm LT C.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach sinh vien hoc lai"

+ Các dòng tiếp theo, mỗi dòng in ra thông tin của từng sinh viên. Các thông tin cách nhau dấu cách (thứ tự tên, mã, điểm Toán, điểm Triết, điểm LT C), điểm trung bình in ở cuối mỗi dòng.

+ Dòng cuối cùng in ra thông báo "So sv phai hoc lai: X"//X là số lượng sv phải học lại ít nhất 2 môn

Constrains: 1<=n<=100, điểm các môn và điểm trung bình có độ chính xác 2 chữ số thập phân, mã sinh viên là các số nguyên dương, tên sinh viên không chứa dấu cách.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  Huy 87 5.00 3.00 3.00  Lan 2 3.00 2.00 3.00  Anh 25 7.00 9.00 10.00  Van 35 0.00 10.00 9.00  Trang 76 9.00 9.00 9.00 | Danh sach sinh vien hoc lai  Huy 87 5.00 3.00 3.00 3.67  Lan 2 3.00 2.00 3.00 2.67  So sv phai hoc lai: 2 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

typedef struct

{

char ten[30];

int maSV;

float diemT, diemTR, diemLTC;

}sv;

void nhap(sv a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].maSV);

scanf("%f",&a[i].diemT);

scanf("%f",&a[i].diemTR);

scanf("%f",&a[i].diemLTC);

}

}

float dtb(sv x)

{

return ((x.diemT+x.diemTR+x.diemLTC)/3);

}

void hocLai(sv a[], int n)

{

int i,d=0;

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

if((a[i].diemT<4.0 && a[i].diemTR<4.0)

|| (a[i].diemT<4.0 && a[i].diemLTC<4.0)

|| (a[i].diemTR<4.0 && a[i].diemLTC<4.0)

|| (a[i].diemT<4.0 && a[i].diemTR<4.0

&& a[i].diemLTC<4.0 ))

{

printf("%s ",a[i].ten);

printf("%d ",a[i].maSV);

printf("%.2f ",a[i].diemT);

printf("%.2f ",a[i].diemTR);

printf("%.2f ",a[i].diemLTC);

printf("%.2f \n",dtb(a[i]));

d++;

}

printf("So sv phai hoc lai: %d",d);

}

int main()

{

sv a[100];

int n;

scanf("%d",&n);

if(n>=1 && n<=100)

{

nhap(a,n);

printf("Danh sach sinh vien hoc lai\n");

hocLai(a,n);

}

}

Question **11**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n sinh viên (n<=100). Sắp xếp danh sách đó theo mã sinh viên (thứ tự tăng dần) và in ra màn hình. Biết thông tin của một sinh viên gồm: tên, mã sv, điểm Toán, điểm Triết, điểm LT C

Input:

+ Dòng thứ nhất nhập vào n là số sinh viên

+ Dòng tiếp theo nhập vào thông tin của các sinh viên theo thứ tự tên, mã, điểm Toán, điểm Triết, điểm LT C.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach sinh vien"

+ Dòng thứ hai in ra số sinh viên trong danh sách

+ Các dòng tiếp theo, mỗi dòng in ra thông tin của từng sinh viên. Các thông tin cách nhau dấu cách, điểm trung bình in ở cuối mỗi dòng.

Constrains: 1<=n<=100, điểm các môn và điểm trung bình có độ chính xác 2 chữ số thập phân, mã sinh viên là các số nguyên dương, tên sinh viên không chứa dấu cách.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  Huy 87 5.00 5.00 4.00  Lan 2 7.00 2.00 7.00  Anh 25 7.00 0.00 10.00  Van 35 0.00 10.00 9.00  Trang 76 2.00 9.00 1.00 | Danh sach sinh vien  5  Lan 2 7.00 2.00 7.00 5.33  Anh 25 7.00 0.00 10.00 5.67  Van 35 0.00 10.00 9.00 6.33  Trang 76 2.00 9.00 1.00 4.00  Huy 87 5.00 5.00 4.00 4.67 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

typedef struct

{

char ten[30];

int maSV;

float diemT,diemTR,diemLTC;

}sv;

void nhap(sv a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].maSV);

scanf("%f",&a[i].diemT);

scanf("%f",&a[i].diemTR);

scanf("%f",&a[i].diemLTC);

}

}

float dtb(sv x)

{

return ((x.diemT+x.diemTR+x.diemLTC)/3);

}

void sapXep(sv a[], int n)

{

int i,j;

sv tg;

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

{

for(j=i+1;j<n;j++)

if(a[i].maSV>a[j].maSV)

{

tg=a[i];

a[i]=a[j];

a[j]=tg;

}

printf("%s ",a[i].ten);

printf("%d ",a[i].maSV);

printf("%.2f ",a[i].diemT);

printf("%.2f ",a[i].diemTR);

printf("%.2f ",a[i].diemLTC);

printf("%.2f\n",dtb(a[i]));

}

}

int main()

{

sv a[100];

int n;

scanf("%d",&n);

if(n>=1 && n<=100)

{

nhap(a,n);

printf("Danh sach sinh vien\n");

printf("%d\n",n);

sapXep(a,n);

}

}

Question **12**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n sinh viên (n<=100). Tìm sinh viên có điểm trung bình cao nhất. Nếu có nhiều sv bằng điểm nhau thì kết quả là sinh viên có vị trí nhỏ nhất trong danh sách.

Biết thông tin của một sinh viên gồm: tên, mã sv, điểm Toán, điểm Triết, điểm LT C

Input:

+ Dòng thứ nhất nhập vào n là số sinh viên

+ Dòng tiếp theo nhập vào thông tin của các sinh viên theo thứ tự tên, mã, điểm Toán, điểm Triết, điểm LT C.

Output:

+ Dòng thứ nhất in ra thông báo "So thu tu cua sinh vien: X" //X là số thứ tự của sv trong danh sách

+ Dòng thứ hai in ra thông tin của sinh viên tìm được. Các thông tin cách nhau dấu cách, điểm trung bình in ở cuối mỗi dòng.

Constrains: 1<=n<=100, điểm các môn và điểm trung bình có độ chính xác 2 chữ số thập phân, mã sinh viên là các số nguyên dương, tên sinh viên không chứa dấu cách.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  Anh 78 10.00 1.00 0.00  Vinh 5 7.00 5.00 8.00  Nhung 61 7.00 3.00 7.00  Trang 27 6.00 7.00 7.00 | So thu tu cua sinh vien: 1  Vinh 5 7.00 5.00 8.00 6.67 |

Answer:(penalty regime: 0 %)

Viết chương trình nhập vào một danh sách gồm n sinh viên (n<=100). Tìm sinh viên có điểm trung bình thấp nhất. Nếu có nhiều người bằng điểm thì kết quả là người ở cuối danh sách. Biết thông tin của một sinh viên gồm: tên, mã sv, điểm Toán, điểm Triết, điểm LT CQuestion **13**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào danh sách gồm n hình tròn trong mặt phẳng. Sắp xếp giảm dần danh sách các hình tròn theo diện tích. In danh sách đã sắp xếp ra màn hình. Biết thông tin của hình tròn gồm mã hình tròn, bán kính, tọa độ tâm (x, y).

Input:

+ Dòng thứ nhất nhập vào số nguyên dương n

+ Dòng tiếp theo nhập vào thông tin của từng hình tròn. Thứ tự nhập vào là mã, bán kính, tâm x, tâm y.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach hinh tron"

+ Dòng thứ 2 in ra số lượng các hình tròn (n)

+ Các dòng tiếp theo in ra thông tin các hình tròn. Thứ tự in ra là mã, tọa độ x, tọa độ y, bán kính. Các thông tin cách nhau dấu cách. Giá trị số thực có độ chính xác 3 chữ số thập phân.

Constrains:

+ 1<=n<=1000

+ bán kính là số thực, tọa độ tâm là các số nguyên

+ Số pi =3.14159

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  78 19.667 962 464  5 15.333 281 827  61 30.667 995 942  27 12.333 391 604 | Danh sach hinh tron  4  61 995 942 30.667  78 962 464 19.667  5 281 827 15.333  27 391 604 12.333 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<math.h>

#define pi 3.14159

typedef struct

{

char maHT[2];

float r;

int x,y;

} ht;

void nhap(ht a[], int n)

{

int i;

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

{

scanf("%s",a[i].maHT);

scanf("%f",&a[i].r);

scanf("%d",&a[i].x);

scanf("%d",&a[i].y);

}

}

float dt(ht a)

{

return (a.r\*a.r\*pi);

}

void ktHT(ht a[], int n)

{

int i,j;

ht tg;

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

for(j=i+1;j<n;j++)

if(dt(a[i]) < dt(a[j]))

{

tg=a[i];

a[i]=a[j];

a[j]=tg;

}

printf("%s ",a[i].maHT);

printf("%d ",a[i].x);

printf("%d ",a[i].y);

printf("%.3f \n",a[i].r);

}}

int main()

{ ht a[1000];

int n;

scanf("%d",&n);

if(n>=1 && n<=1000)

{

nhap(a,n);

printf("Danh sach hinh tron\n");

printf("%d\n",n);

ktHT(a,n);

}

}

Question **14**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n sinh viên (n<=100). In ra màn hình danh sách các sv phải học lại (điểm<4.0). Biết thông tin của một sinh viên gồm: tên, mã sv, điểm Toán, điểm Triết, điểm LT C

Input:

+ Dòng thứ nhất nhập vào n là số sinh viên

+ Dòng tiếp theo nhập vào thông tin của các sinh viên theo thứ tự tên, mã, điểm Toán, điểm Triết, điểm LT C.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach sinh vien hoc lai"

+ Các dòng tiếp theo, mỗi dòng in ra thông tin của từng sinh viên. Các thông tin cách nhau dấu cách (thứ tự tên, mã, điểm Toán, điểm Triết, điểm LT C), điểm trung bình in ở cuối mỗi dòng.

+ Dòng cuối cùng in ra thông báo "So sv phai hoc lai: X"//X là số lượng sv phải học lại

Constrains: 1<=n<=100, điểm các môn và điểm trung bình có độ chính xác 2 chữ số thập phân, mã sinh viên là các số nguyên dương, tên sinh viên không chứa dấu cách.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  Huy 87 5.00 5.00 5.00  Lan 2 7.00 2.00 7.00  Anh 25 7.00 9.00 10.00  Van 35 0.00 10.00 9.00  Trang 76 9.00 9.00 9.00 | Danh sach sinh vien hoc lai  Lan 2 7.00 2.00 7.00 5.33  Van 35 0.00 10.00 9.00 6.33  So sinh vien phai hoc lai: 2 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

typedef struct

{

char ten[30];

int maSV;

float diemT, diemTR, diemLTC;

}sv;

void nhap(sv a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].maSV);

scanf("%f",&a[i].diemT);

scanf("%f",&a[i].diemTR);

scanf("%f",&a[i].diemLTC);

}

}

float dtb(sv x)

{

return ((x.diemT+x.diemTR+x.diemLTC)/3);

}

void hocLai(sv a[], int n)

{

int i,d=0;

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

if(a[i].diemT<4.0 || a[i].diemTR<4.0 || a[i].diemLTC<4.0)

{

printf("%s ",a[i].ten);

printf("%d ",a[i].maSV);

printf("%.2f ",a[i].diemT);

printf("%.2f ",a[i].diemTR);

printf("%.2f ",a[i].diemLTC);

printf("%.2f \n",dtb(a[i]));

d++;

}

printf("So sinh vien phai hoc lai: %d",d);

}

int main()

{

sv a[100];

int n;

scanf("%d",&n);

if(n>=1 && n<=100)

{

nhap(a,n);

printf("Danh sach sinh vien hoc lai\n");

hocLai(a,n);

}

}

Question **15**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n nhân viên (n<=100). Sắp xếp danh sách nhân viên theo tên. Thứ tự từ a đến z.  Biết thông tin của một nhân viên gồm: tên, mã nv, hệ số lương, phụ cấp

Input:

+ Dòng thứ nhất nhập vào n là số nhân viên

+ Dòng tiếp theo nhập vào thông tin của các nhân viên theo thứ tự tên, mã, hệ số lương, phụ cấp.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach nhan vien"

+ Dòng thứ hai in ra số nhân viên trong danh sách

+ Các dòng tiếp theo, mỗi dòng in ra thông tin của từng nhân viên. Các thông tin cách nhau dấu cách, lương tháng in ở cuối mỗi dòng. Lương tháng được tính bằng hệ số lương \* 2000000 + phụ cấp.

Constrains: 1<=n<=100, mã nhân viên, phụ cấp là các số nguyên dương, tên nhân viên không chứa dấu cách, các thông tin còn lại là số thực có độ chính xác 2 chữ số thập phân.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  Anh 78 2.34 1000000  Vinh 5 3.3 500000  Nhung 61 2.67 700000  Trang 27 4.32 250000 | Danh sach nhan vien  4  Anh 78 2.34 1000000 5680000.00  Nhung 61 2.67 700000 6040000.00  Trang 27 4.32 250000 8890000.00  Vinh 5 3.30 500000 7100000.00 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<string.h>

typedef struct

{

char ten[30];

int maNV;

float hsLuong;

int phuCap;

} nv;

void nhap(nv a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].maNV);

scanf("%f",&a[i].hsLuong);

scanf("%d",&a[i].phuCap);

}

}

float luong(nv x)

{

return ((x.hsLuong\*2000000)+x.phuCap);

}

void sapXep(nv a[], int n)

{

int i,j;

nv tg;

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

{

for(j=i+1;j<n;j++)

if(strcmp(a[i].ten,a[j].ten)>0)

{

tg=a[i];

a[i]=a[j];

a[j]=tg;

}

printf("%s ",a[i].ten);

printf("%d ",a[i].maNV);

printf("%.2f ",a[i].hsLuong);

printf("%d ",a[i].phuCap);

printf("%.2f\n",luong(a[i]));

}

}

int main()

{

nv a[100];

int n;

scanf("%d",&n);

if(n>=1 && n<=100)

{

nhap(a,n);

printf("Danh sach nhan vien\n");

printf("%d\n",n);

sapXep(a,n);

}

}

Question **16**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n thí sinh thi đại học (n<=100). In danh sách các thí sinh thi đỗ ra màn hình. Biết thông tin của một thí sinh gồm: tên, số báo danh, điểm môn 1, điểm môn 2, điểm môn 3. Thí sinh thi đỗ khi tổng điểm >= 15.00 và điểm các môn >=1.00.

Input:

+ Dòng thứ nhất nhập vào n là số thí sinh

+ Dòng tiếp theo nhập vào thông tin của các thí sinh theo thứ tự tên, số báo danh, điểm môn 1, điểm môn 2, điểm môn 3.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach thi sinh thi do"

+ Các dòng tiếp theo, mỗi dòng in ra thông tin của từng thí sinh. Các thông tin cách nhau dấu cách (thứ tự: tên, số báo danh, điểm môn 1, điểm môn 2, điểm môn 3), tổng điểm in ở cuối mỗi dòng.

+ Dòng cuối in ra thông báo "So thi sinh thi do: X". X là số lượng thí sinh thi đỗ.

Constrains: 1<=n<=100, điểm các môn và tổng điểm có độ chính xác 2 chữ số thập phân, số báo danh là số nguyên dương, tên thí sinh không chứa dấu cách.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  Huy 87 5.00 3.00 3.00  Lan 2 3.00 2.00 3.00  Anh 25 7.00 9.00 10.00  Van 35 0.00 10.00 9.00  Trang 76 9.00 9.00 9.00 | Danh sach thi sinh thi do  Anh 25 7.00 9.00 10.00 26.00  Trang 76 9.00 9.00 9.00 27.00  So thi sinh thi do: 2 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

typedef struct

{

char ten[30];

int sbd;

float diem1, diem2, diem3;

}ths;

void nhap(ths a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].sbd);

scanf("%f",&a[i].diem1);

scanf("%f",&a[i].diem2);

scanf("%f",&a[i].diem3);

}

}

float tong(ths x)

{

return (x.diem1+x.diem2+x.diem3);

}

void tke(ths a[], int n)

{

int i,d=0;

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

{

if(tong(a[i])>=15 && a[i].diem1>=1

&& a[i].diem2>=1 && a[i].diem3>=1)

{

printf("%s ",a[i].ten);

printf("%d ",a[i].sbd);

printf("%.2f ",a[i].diem1);

printf("%.2f ",a[i].diem2);

printf("%.2f ",a[i].diem3);

printf("%.2f \n",tong(a[i]));

d++;

}

}

printf("So thi sinh thi do: %d", d);

}

int main()

{

ths a[100];

int n;

scanf("%d",&n);

if(n>=1 && n<=100)

{

nhap(a,n);

printf("Danh sach thi sinh thi do\n");

tke(a,n);

}

}

Question **17**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào danh sách gồm n hình tròn trong mặt phẳng. In thông tin của hình tròn có tâm gần gốc tọa độ nhất ra màn hình. Nếu có nhiều hình tròn thỏa mãn thì kết quả tìm được là hình tròn có số thứ tự nhỏ nhất. Biết thông tin của hình tròn gồm mã hình tròn, bán kính, tọa độ tâm (x, y).

Input:

+ Dòng thứ nhất nhập vào số nguyên dương n

+ Dòng tiếp theo nhập vào thông tin của từng hình tròn. Thứ tự nhập vào là mã, bán kính, tâm x, tâm y.

Output:

+ In ra thông tin của hình tròn tìm được. Thứ tự in ra là mã, tọa độ x, tọa độ y, bán kính. Các thông tin cách nhau dấu cách. Giá trị số thực có độ chính xác 3 chữ số thập phân.

Constrains:

+ 1<=n<=1000

+ bán kính là số thực, tọa độ tâm là các số nguyên

+ Số pi =3.14159

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  78 19.667 962 464  5 15.333 281 827  61 30.667 995 942  27 12.333 391 604 | 27 391 604 12.333 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include <math.h>

#include <string.h>

typedef struct

{

char MaHT[2] ;

float BK;

int x;

int y;

} HT;

void nhap (HT a[], int n)

{

int i;

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

{

scanf("%s",a[i].MaHT);

scanf("%f",&a[i].BK);

scanf("%d",&a[i].x);

scanf("%d",&a[i].y);

}

}

float ktrTam(HT a)

{

return sqrt(pow((a.x), 2) + pow(( a.y), 2));

}

void ktrTmin (HT a[],int n)

{

int i;

HT min=a[0];

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

{

if (ktrTam(min)>ktrTam(a[i]))

{

min=a[i];

}

}

printf("%s ",min.MaHT);

printf("%d ",min.x);

printf("%d ",min.y);

printf("%.3f",min.BK);

}

int main ()

{

HT a[1000];

int n;

scanf("%d\n",&n);

if(n>=1 && n<=1000)

{

nhap(a,n);

ktrTmin(a,n);

}

}

Question **18**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một danh sách gồm n sinh viên (n<=100). In ra màn hình danh sách các sv đạt học bổng (điểm các môn >=5.5 và điểm trung bình>=7.0). Biết thông tin của một sinh viên gồm: tên, mã sv, điểm Toán, điểm Triết, điểm LT C

Input:

+ Dòng thứ nhất nhập vào n là số sinh viên

+ Dòng tiếp theo nhập vào thông tin của các sinh viên theo thứ tự tên, mã, điểm Toán, điểm Triết, điểm LT C.

Output:

+ Dòng thứ nhất in ra thông báo "Danh sach sinh vien dat hoc bong"

+ Các dòng tiếp theo, mỗi dòng in ra thông tin của từng sinh viên. Các thông tin cách nhau dấu cách, điểm trung bình in ở cuối mỗi dòng.

+ Dòng cuối cùng in ra thông báo "So sinh vien dat hoc bong: X"//X là số lượng sv đạt học bổng

Constrains: 1<=n<=100, điểm các môn và điểm trung bình có độ chính xác 2 chữ số thập phân, mã sinh viên là các số nguyên dương, tên sinh viên không chứa dấu cách.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  Huy 87 5.00 3.00 5.00  Lan 2 7.00 7.00 7.00  Anh 25 7.00 5.00 10.00  Van 35 9.00 10.00 9.00  Trang 76 4.00 9.00 9.00 | Danh sach sinh vien dat hoc bong  Lan 2 7.00 7.00 7.00 7.00  Van 35 9.00 10.00 9.00 9.33  So sinh vien dat hoc bong: 2 |

Answer:(penalty regime: 0 %

#include<stdio.h>

typedef struct

{

char ten[30];

int maSV;

float diemT, diemTR, diemLTC;

}sv;

void nhap(sv a[], int n)

{

int i;

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

{

scanf("%s",a[i].ten);

scanf("%d",&a[i].maSV);

scanf("%f",&a[i].diemT);

scanf("%f",&a[i].diemTR);

scanf("%f",&a[i].diemLTC);

}

}

float dtb(sv x)

{

return ((x.diemT+x.diemTR+x.diemLTC)/3);

}

void hocBong(sv a[], int n)

{

int i,d=0;

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

if(a[i].diemT>=5.5 && a[i].diemTR>=5.5

&& a[i].diemLTC>=5.5 && dtb(a[i])>=7.0)

{

printf("%s ",a[i].ten);

printf("%d ",a[i].maSV);

printf("%.2f ",a[i].diemT);

printf("%.2f ",a[i].diemTR);

printf("%.2f ",a[i].diemLTC);

printf("%.2f \n",dtb(a[i]));

d++;

}

printf("So sinh vien dat hoc bong: %d\n",d);

}

int main()

{

sv a[100];

int n;

scanf("%d",&n);

if(n>=1 && n<=100)

{

nhap(a,n);

printf("Danh sach sinh vien dat hoc bong\n");

hocBong(a,n);

}

}

Question **1**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một chuỗi ký tự chỉ gồm chữ cái, chữ số. Nếu các chữ cái, chữ số trong chuỗi xuất hiện nhiều hơn một lần thì xóa ở lần xuất hiện thứ 2, 3, ... In chuỗi kết quả ra màn hình.

Input: chuỗi ký tự có độ dài <=100.

Output: in ra chuỗi ký tự sao cho mỗi ký tự chỉ xuất hiện không quá một lần.

Constrains: chuỗi ký tự chỉ chứa chữ cái, chữ số, độ dài không quá 100.

**For example:**

| **Input** | **Result** |
| --- | --- |
| Chaocacbansinhvien | Chaocbnsive |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<string.h>

#include<ctype.h>

int main()

{

int X[256];

char s[100];

fflush(stdin);

scanf("%s", s);

for(int i=0; i<256; i++)

{

X[i]=0;

}

for(unsigned int i=0; i<strlen(s); i++)

{

X[(int)s[i]]++;

if(X[(int)s[i]]<2) printf("%c",s[i]);

}

}

Question **2**

Correct

Marked out of 1.00

Flag question

Question text

Giả sử mật khẩu để đăng nhập vào tài khoản của bạn là "123456a@B". Bạn hãy viết chương trình cho yêu cầu nhập vào mật khẩu. Nếu nhập sai (không quá 5 lần) thì cho nhập lại. Nếu nhập sai quá 5 lần thì thông báo "Tai khoan bi khoa, X". Nếu nhập đúng thì thông báo "Dang nhap thanh cong, X".

Trong đó: X là số lần bạn nhập mật khẩu.

Input:

+ một chuỗi ký tự không có dấu cách

Output:

+ Thông báo như đề bài

Constrains:

+ Mật khẩu là chuỗi ký tự không chứa dấu cách, dấu tab, độ dài tối đa là 30.

**For example:**

| **Input** | **Result** |
| --- | --- |
| p1KaLmIj2k  B7fZp5Tm1q  bi6dss92C3  1aPGJK9Lvk  7DMPDONXrw | Tai khoan bi khoa, 5 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<string.h>

int main()

{

char mk[30]="123456a@B";

char s[30];

int temp=0, d=0;

while(temp<5){

temp++;

scanf("%s",s);

if(strcmp(s,mk)==0)

{

printf("Dang nhap thanh cong, %d ",temp);

d++;

break;

}

}

if(temp==5&&d==0)

printf("Tai khoan bi khoa, 5");

}

Question **3**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một chuỗi ký tự. In ra số lần xuất hiện từng ký tự trong chuỗi.

Input: Chuỗi ký tự không bao gồm các ký tự điều khiển.

Output: Gồm nhiều dòng, mỗi dòng in ra X: Y

Trong đó: X là ký tự, Y là số lần xuất hiện của X trong chuỗi ban đầu. Thứ tự in ra tuân theo mã ASCII của ký tự.

Constrains: Chuỗi ký tự không bao gồm các ký tự điều khiển, có độ dài không quá 100.

Ví dụ:

Input: Chao Cac Ban

Output:

 : 2 //dấu cách xuất hiện 2 lần

B: 1

C: 2

a: 3

c: 1

h: 1

n: 1

o: 1

**For example:**

| **Input** | **Result** |
| --- | --- |
| Chao Cac Ban | : 2  B: 1  C: 2  a: 3  c: 1  h: 1  n: 1  o: 1 |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<string.h>

#include<ctype.h>

int main()

{

int X[256];

char s[100];

fflush(stdin);

scanf("%[^\n]", s);

for(int i=0; i<256; i++)

{

X[i]=0;

}

for(unsigned int i=0; i<strlen(s); i++)

{

X[(int)s[i]]++;

}

for(unsigned int i=0; i<256; i++)

{

if(X[i]>0)

{

printf("%c: %d\n", i, X[i]);

}

}

}

Question **4**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một chuỗi gồm các chữ cái và dấu cách. Chuẩn hóa chuỗi vừa nhập thành dạng tên riêng.

Input: chuỗi ký tự bao gồm chữ cái, dấu cách.

Output: chuỗi ký tự ở dạng tên riêng.

Constrains: chuỗi ký tự chỉ chứa các chữ cái, dấu cách.

**For example:**

| **Input** | **Result** |
| --- | --- |
| NGuyen VAn AnH | Nguyen Van Anh |

Answer:(penalty regime: 0 %)

#include<stdio.h>

#include<string.h>

#include<ctype.h>

void Xoa(char s[], unsigned int k, unsigned int l)

{

for(unsigned int i=k; i<l-1; i++)

{

s[i]=s[i+1];

}

s[l-1]='\0';

}

void XoaCach(char s[], unsigned int l)

{

for(unsigned int i=0; i<l; i++)

{

if(s[i]==' ' && s[i+1]==' ')

{

Xoa(s,i+1,l);

i--;

l--;

}

}

if(s[0]==' ') Xoa(s,0,l);

l=strlen(s);

if(s[l-1]==' ') s[l-1]='\0';

}

void Hoa(char s[], unsigned int l)

{

s[0]=toupper(s[0]);

for(unsigned int i=1; i<l; i++)

{

if(isalpha(s[i])==1 && s[i-1]!=' ') s[i]=tolower(s[i]);

if(s[i]==' ') s[i+1]=toupper(s[i+1]);

}

}

int main()

{

char s[100];

fflush(stdin);

scanf("%[^\n]", s);

unsigned int l=strlen(s);

XoaCach(s,l);

l=strlen(s);

Hoa(s,l);

printf("%s", s);

}