Question **1**

Correct

Marked out of 2.00

Flag question

Question text

Chương trình bên dưới nhận vào hai số nguyên, rồi in ra thông báo:

* **Ca hai so deu bang zero.** Nếu hai số nhập vào đều bằng 0.
* **Khong phai ca hai so deu bang zero.** Nếu có ít nhất một số khác 0.

Hiện tại chương trình đang có lỗi logic, hãy copy xuống và sửa lại cho đúng.

/\* --------------------------------------------------- \*

\* Kiểm tra xem cả 2 số nguyên nhập vào đều bằng zero? \*

\* --------------------------------------------------- \*/

#include <stdio.h>

int main()

{

int x, y;

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

if( (x=0) && (y=0) ) // Nếu x bằng 0 và y bằng 0

{

printf( "Ca hai so deu bang zero." );

}

else // Ngược lại

{

printf( "Khong phai ca hai so deu bang zero." );

}

return 0;

}

**Input Format**

* Hai số nguyên cách nhau 1 dấu cách.

**Constraints**

* Không có.

**Output Format**

* In ra 1 trong 2 xâu kí tự in đậm bên trên.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| 0\_0 | 0 0 | Ca hai so deu bang zero. |
| 1\_9 | 1 9 | Khong phai ca hai so deu bang zero. |

#include <stdio.h>

int main()

{

int x, y;

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

if( (x==0) && (y==0) ) // Nếu x bằng 0 và y bằng 0

{

printf( "Ca hai so deu bang zero." );

}

else // Ngược lại

{

printf( "Khong phai ca hai so deu bang zero." );

}

return 0;

}

Question **2**

Correct

Marked out of 1.00

Flag question

Question text

Kích cỡ các kiểu dữ liệu và giá trị lớn/nhỏ nhất có thể được biểu diễn của từng kiểu phụ thuộc vào hệ thống máy và trình biên dịch.

Viết chương trình in ra giá trị lớn nhất và nhỏ nhất của các kiểu dữ liệu trên hệ thống Hoctructuyen theo thứ tự sau:

* char
* unsigned char
* int
* unsigned int
* long
* unsigned long
* float
* double

*Gợi ý: giá trị lớn/nhỏ nhất của các kiểu được định nghĩa trong limits.h và float.h như là các kí hiệu INT\_MAX, INT\_MIN, UINT\_MAX, LONG\_MIN, LONG\_MAX,  ULONG\_MAX, FLT\_MIN, FLT\_MAX, DBL\_MIN, DBL\_MAX*

*Ví dụ:*

printf( "char:%d~%d\n", CHAR\_MIN, CHAR\_MAX );

**Input Format**

* Không có.

**Constraints**

* Không có.

**Output Format**

* Mỗi kiểu dữ liệu được in trên 1 dòng.
* Mỗi dòng in theo quy cách **tên kiểu dữ liệu:giá trị nhỏ nhất~giá trị lớn nhất**
* Sáu kiểu đầu (số nguyên): các giá trị MIN, MAX được in với đặc tả số nguyên tương ứng như sau:

char, unsigned char, int:%d.

unsigned int:%u

long:%ld

unsigned long:%lu

* Hai kiểu cuối cùng (số thực): các giá trị MIN, MAX được in ở dạng khoa học tương ứng như sau:

float:%E

double:%lE

**For example:**

| **Test** | **Result** |
| --- | --- |
| tc01 | char:-128~127  unsigned char:0~255  int:-2147483648~2147483647  unsigned int:0~4294967295  long:-9223372036854775808~9223372036854775807  unsigned long:0~18446744073709551615  float:1.175494E-38~3.402823E+38  double:2.225074E-308~1.797693E+308 |

#include <stdio.h>

int main(){

puts("char:-128~127\n"

"unsigned char:0~255\n"

"int:-2147483648~2147483647\n"

"unsigned int:0~4294967295\n"

"long:-9223372036854775808~9223372036854775807\n"

"unsigned long:0~18446744073709551615\n"

"float:1.175494E-38~3.402823E+38\n"

"double:2.225074E-308~1.797693E+308");

}

Question **3**

Correct

Marked out of 2.00

Flag question

Question text

Viết chương trình nhận vào 1 kí tự, dùng các biểu thức logic để kiểm tra xem nó là kí tự thường, kí tự hoa, chữ số hay kí tự trắng (gồm dấu cách, dấu tab, xuống dòng)

* Nếu là kí tự thường thì in ra **THUONG**
* Nếu là kí tự hoa thì in ra **HOA**
* Nếu là kí số thì in ra **SO**
* Nếu là kí tự trắng (dấu cách, Tab, xuống dòng) thì in ra **TRANG**

*Gợi ý:*

// kiểm tra kí tự nhập vào

if ( ch>='a' && ch<='z' )

{

printf( "THUONG" );

}

else if( ... )

{

printf( "HOA" );

}

else if( ... )

{

printf( "SO" );

}

else if( ... )

{

printf( "TRANG" );

}

**Input Format**

* Một kí tự có thể là kí tự a-z, A-Z, 0-9, dấu cách, Tab, xuống dòng.

**Constraints**

* Không có.

**Output Format**

In ra 1 trong 4 xâu, tùy thuộc vào kí tự nhập vào:

* **THUONG**
* **HOA**
* **SO**
* **TRANG**

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| A | A | HOA |
| a | a | THUONG |
| \t |  | TRANG |

#include <stdio.h>

int main(){

char ch;

scanf("%c",&ch);

if(ch>='a' && ch<='z'){

printf("THUONG");

}else if(ch>='A' && ch<='Z'){

printf("HOA");

}else if(ch>='0' && ch<='9'){

printf("SO");

}else if(ch==32 || ch==9 || ch==10){

printf("TRANG");

}

return 0;

}

Question **4**

Correct

Marked out of 1.00

Flag question

Question text

Đoạn chương trình bên dưới nhận vào 2 số nguyên, tính rồi hiển thị kết quả phép chia số thứ 1 cho số thứ 2. Kết quả phép chia là số thực.

Hiện tại chương trình đang có lỗi trong một số trường hợp. Hãy sửa lại cho đúng.

#include <stdio.h>

int main()

{

int a, b;

float thuong;

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

thuong = a/b;

printf( "%.2f", thuong );

return 0;

}

**Input Format**

* Hai số nguyên (a, b) cách nhau bởi 1 dấu cách.

**Constraints**

* b != 0

**Output Format**

* Kết quả phép chia: số thực, in ra với độ chính xác 2 chữ số thập phân.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| 3\_2 | 3 2 | 1.50 |
| 4\_2 | 5 2 | 2.50 |

#include <stdio.h>

int main()

{

int a, b;

float thuong;

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

thuong = (float)a/b;

printf( "%.2f", thuong );

return 0;

}

Question **6**

Correct

Marked out of 2.00

Flag question

Question text

C dùng những số nguyên nhỏ để biểu diễn các kí tự. Số nguyên này được gọi là mã ASCII của kí tự.

Ta có thể in mã ASCII của một kí tự bằng lệnh

printf( "%d", 'A' );

Viết chương trình nhận vào một kí tự, in ra mã ASCII tương ứng của nó.

**Input Format**

* Một kí tự.

**Constraints**

* Không có.

**Output Format**

* Mã ASCII (số nguyên) tương ứng của kí tự nhập vào.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| A\_65 | A | 65 |
| a\_97 | a | 97 |
| 0\_48 | 0 | 48 |
| $\_36 | $ | 36 |

#include <stdio.h>

int main(){

char ch;

scanf("%c",&ch);

printf("%d",ch);

}

Question **7**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một biểu thức dạng x op y� op �, trong đó:

* x,y�,� là hai số nguyên
* opop là một trong năm phép toán: +, -, \*, /, % (chia lấy dư)

và tính giá trị của biểu thức đó. Nếu không thể tính giá trị của biểu thức thì in ra "error"

+ Đầu vào: một dòng chứa biểu thức cần tính giá trị

+ Đầu ra: Giá trị của biểu thức với độ chính xác là hai chữ số sau dấu phẩy

+ Ràng buộc:

* x,y∈Z�,�∈�
* 0<=x,y≤3×1090<=�,�≤3×109

**For example:**

| **Input** | **Result** |
| --- | --- |
| 1+2 | 3.00 |

#include <stdio.h>

int main() {

int x, y;

char op;

float result;

scanf("%d%c%d", &x, &op, &y);

switch (op) {

case '+':

result = x + y;

printf("%.2f\n", result);

break;

case '-':

result = x - y;

printf("%.2f\n", result);

break;

case '\*':

result = x \* y;

printf("%.2f\n", result);

break;

case '/':

if (y == 0) {

printf("error\n");

} else {

result = (float)x / y;

printf("%.2f\n", result);

}

break;

case '%':

if (y == 0) {

printf("error\n");

} else {

result = x % y;

printf("%.2f\n", result);

}

break;

default:

printf("error\n");

}

return 0;

}

Question **8**

Correct

Marked out of 2.00

Flag question

Question text

Viết chương trình nhận vào một số n� ở dạng thập lục phân với phần tiền tố (prefix) là 0x0� và hiển thị số đó dưới dạng thập phân.

+ Đầu vào: Số n�

+ Đầu ra: Số n� ở dạng thập phân. Nếu n� không phải là một số ở dạng thập lục phân thì in ra Invalid inputInvalid input.

+ Ràng buộc: n� có không quá 10 chữ số

**For example:**

| **Input** | **Result** |
| --- | --- |
| 0x1B | 27 |
| 1B | Invalid input |
| 0x1a | 26 |
| 0x4g | Invalid input |

#include <stdio.h>

int main(){

char n[100] = {0};

unsigned long long output = 0;

int i;

scanf("%s", n);

if(n[12] != 0)

{

printf("Invalid input");

return 0;

}

else

{

if((n[0] == 48) && (n[1]==120))

{

for(i = 2; n[i] != 0; i = i + 1)

{

output \*= 16;

if((48 <= n[i]) && (n[i] <= 57))

{

output = output + (n[i] - 48);

}

else if((97 <= n[i]) && (n[i] <= 102))

{

output = output + ((n[i] - 97) + 10);

}

else if((65 <= n[i]) && (n[i] <= 70))

{

output = output + ((n[i] - 65) + 10);

}

else

{

printf("Invalid input");

return 0;

}

}

}

else

{

printf("Invalid input");

return 0;

}

}

printf("%llu", output);

return 0;

}

***Question 9***

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào một số thập phân và hiển thị số đó dưới dạng thập lục phân với phần tiền tố là 0x, các chữ số là chữ hoa.

+ Đầu vào: Số thập phân n�

+ Đầu ra: Số n� ở dạng thập lục phân

+ Ràng buộc: 0≤n≤10120≤�≤1012

**For example:**

| **Input** | **Result** |
| --- | --- |
| 23 | 0x17 |

#include <stdio.h>

int main(){

double n;

char output[100] = {0};

int i;

scanf("%lf", &n);

if((n < 0) || ( n > 1000000000000))

{

return 0;

}

sprintf(output, "0x%lx", (unsigned long)n);

for(i = 2; output[i] != 0; i++)

{

if((output[i] >= 'a') && (output[i] <= 'f'))

{

output[i] = (output[i] - 'a') + 'A';

}

}

printf("%s", output);

return 0;

}

Question **10**

Correct

Marked out of 2.00

Flag question

Question text

Viết chương trình nhập vào 2 số nguyên, tính tổng, hiệu, tích, thương và in kết quả ra màn hình.

Input: Hai số nguyên cách nhau dấu cách. Ví dụ 4 5

Output: In ra thông báo trên các dòng như sau:

Dòng 1: Tong X + Y = Z //X, Y là 2 số ở input, Z là kết quả

Dòng 2: Hieu X - Y = Z

Dòng 3: Tich X \* Y = Z

Dòng 4: Phan nguyen X / Y = Z

Dòng 5: Phan du  X % Y = Z

Dòng 6: Ket qua thuc  X / Y = Z //lấy kết quả là số thực, với độ chính xác 2 chữ số thập phân

Constrains: Các giá trị kiểu số nguyên, kết quả của dòng 6 là số thực

**For example:**

| **Input** | **Result** |
| --- | --- |
| 1 6 | Tong 1 + 6 = 7  Hieu 1 - 6 = -5  Tich 1 \* 6 = 6  Phan nguyen 1 / 6 = 0  Phan du 1 % 6 = 1  Ket qua thuc 1 / 6 = 0.17 |

#include<stdio.h>

int main(){

int a,b;

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

printf("Tong %d + %d = %d\n",a,b,a+b);

printf("Hieu %d - %d = %d\n",a,b,a-b);

printf("Tich %d \* %d = %d\n",a,b,a\*b);

printf("Phan nguyen %d / %d = %d\n",a,b,a/b);

printf("Phan du %d %% %d = %d\n",a,b,a%b);

printf("Ket qua thuc %d / %d = %.2f",a,b,(float)a/b);

}

Question **17**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình thực hiện:

* Nhập hai số nguyên cách nhau bởi dấu cách.
* In ra bảng cửu chương trong phạm vi hai số đó.

**Input Format**

* Hai số nguyên dương **a**, **b** cách nhau đúng một dấu cách.

**Constraints**

* 1 <= **a**,**b** <= 9

**Output Format**

* Mỗi phép nhân in trên 1 dòng, trên dòng không có dấu cách.
* Phép nhân đầu tiên và cuối cùng lần lượt là nhân với 1 và nhân với 9.
* Sau phép nhân (dòng) cuối cùng là 1 dòng trống.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| 2\_3 | 2 3 | 2x1=2  2x2=4  2x3=6  2x4=8  2x5=10  2x6=12  2x7=14  2x8=16  2x9=18  3x1=3  3x2=6  3x3=9  3x4=12  3x5=15  3x6=18  3x7=21  3x8=24  3x9=27 |

#include <stdio.h>

int main(){

int a,b;

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

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

for(int j=1;j<10;j++){

printf("%dx%d=%d\n",i,j,i\*j);

}

}

return 0;

}

Question **18**

Correct

Marked out of 1.00

Flag question

Question text

Nhập 1 số nguyên dương **n** chỉ độ dài cạnh hình tam tác.

In ra hình tam giác gồm các kí tự \* với độ dài 3 cạnh là **n**, cụ thể như sau:

* Dòng 1: in ra (n-1) kí tự ⎵ và (1) cặp kí tự \*⎵
* Dòng 2: in ra (n-2) kí tự ⎵ và (2) cặp kí tự \*⎵
* ...
* Dòng n: in ra (0) kí tự ⎵ và (n) cặp kí tự \*⎵
* Sau dòng cuối dùng (dòng n) là 1 dòng trống

Với ⎵ là kí tự trắng.

**Input Format**

Một số nguyên dương. Vd: 6

**Constraints**

Không có.

**Output Format**

Như mô trả trên.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| 6 | 6 | \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \*  \* \* \* \* \* \* |

#include <stdio.h>

int main(){

int n;

scanf("%d",&n);

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

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

printf(" ");

}

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

printf("\* ");

}

printf("\n");

}

return 1;

}

Question **19**

Correct

Marked out of 1.00

Flag question

Question text

Nhập vào một số nguyên dương **n**. In **n** năm nhuận tiếp theo sau năm nhuận gần nhất là 2016. Năm nhuận là năm thỏa mãn a) hoặc b)

a) Nó lặp lại 4 năm 1 lần, nhưng không lặp lại mỗi 100 năm. (chia hết cho 4 nhưng không chia hết cho 100)

b) Năm nhuận cũng lặp lại mỗi 400 năm. (chia hết cho 400)

**Input Format**

* Một số nguyên dương - **n**.

**Constraints**

* **n** > 0

**Output Format**

* In **n** năm nhuận sau năm 2016 trên 1 dòng, cách nhau 1 dấu cách.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| 5 | 5 | 2020 2024 2028 2032 2036 |

#include <stdio.h>

int main(){

int a;

int dem = 0;

int nam = 2016;

scanf("%d",&a);

while(dem<a){

nam++;

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

printf("%d ",nam);

dem++;

}

}

return 0;

}

Question **20**

Correct

Marked out of 1.00

Flag question

Question text

Giải bất phương trình **ax+b>0**, với **a** và **b** là 2 số nguyên nhập vào.

**Input Format**

Hai số nguyên, **a** và **b**, cách nhau 1 dấu cách.

**Constraints**

Không có.

**Output Format**

* Nếu bpt có vô số nghiệm, in ra **VSN**
* Nếu bpt vô nghiệm, in ra **VN**
* Nếu bpt có nghiệm, in ra nghiệm dạng **x>p** hoặc **x<p**

Trong đó, **p** là một giá trị thực với độ chính xác 2 chữ số sau dấu chấm.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| -2\_4 | -2 4 | x<2.00 |

#include <stdio.h>

int main(){

int a,b;

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

if(a==0){

if(b<=0){

printf("VN");

}else {

printf("VSN");

}

}else {

if(a>0){

printf("x>%.2f",(float)-b/a);

}else {

printf("x<%.2f",(float)-b/a);

}

}

return 0;

}

Question **21**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình thực hiện:

* Nhập hai số nguyên cách nhau bởi dấu cách.
* In ra bảng cửu chương trong phạm vi hai số đó.

**Input Format**

* Hai số nguyên dương **a**, **b** cách nhau đúng một dấu cách.

**Constraints**

* 1 <= **a**, **b** <= 9

**Output Format (in ra bảng cửu chương dạng khuyết)**

* Mỗi phép nhân in trên 1 dòng, trên dòng không có dấu cách.
* Sau phép nhân (dòng) cuối cùng là 1 dòng trống.
* Các phép tính (có thể) đã thực hiện trước đó không được in lại ở phía sau.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| 2\_3 | 2 3 | 2x2=4  2x3=6  2x4=8  2x5=10  2x6=12  2x7=14  2x8=16  2x9=18  3x3=9  3x4=12  3x5=15  3x6=18  3x7=21  3x8=24  3x9=27 |

#include <stdio.h>

int main(){

int a,b;

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

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

for(int j=i;j<10;j++){

printf("%dx%d=%d\n",i,j,i\*j);

}

}

return 0;

}

Question **22**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào 2 số nguyên a, b. Đếm số lượng và tính tổng các số lẻ trong đoạn [a, b], hoặc [b, a].

Input:

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

Ví dụ: 5 12 hoặc 30 -8

Output:

Số các số lẻ có và tổng các số lẻ tìm được. Hai giá trị này cách nhau dấu cách

Ví dụ:

+ Input: 5 12

+ Output: 4 32

Constrains:

các biến kiểu nguyên

**For example:**

| **Input** | **Result** |
| --- | --- |
| 9 1 | 5 25 |

#include <stdio.h>

int main() {

int a, b;

int dem = 0;

int tong = 0;

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

if (a > b) {

int temp = a;

a = b;

b = temp;

}

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

if (num % 2 != 0) {

dem++;

tong += num;

}

}

printf("%d %d", dem,tong);

return 0;

}

Question **23**

Correct

Marked out of 1.00

Flag question

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 |

#include <stdio.h>

int main(){

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

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

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

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

if(d!=0){

x=(b2 \* c1 - b1 \* c2)/d;

y=(a1 \* c2 - a2 \* c1)/d;

printf("%.5f %.5f",x,y);

}else{

if((a1 \* c2 - a2 \* c1)==0 && (b1 \* c2 - b2 \* c1)==0){

printf("VSN");

}else {

printf("VN");

}

}

return 0;

}

Question **24**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình giải phương trình (pt) ax+b=0.

Với **a**, **b** là hai số thực nhập từ bàn phím.

**Input Format**

* Hai số thực cách nhau dấu 1 cách.

**Constraints**

* Không có.

**Output Format**

* Nếu pt vô nghiệm, in ra chuỗi "**VN"**.
* Nếu pt có 1 nghiệm, in ra nghiệm ở định dạng số thực với độ chính xác 2 chữ số sau dấu chấm thập phân.
* Nếu pt có vô số nghiệm, in ra chuỗi "**VSN"**.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| 1x+2=0 | 1.00 2.00 | -2.00 |

#include <stdio.h>

int main(){

float a,b;

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

if(a==0){

if(b==0){

printf("VSN");

}else {

printf("VN");

}

}else {

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

}

}

Question **25**

Correct

Marked out of 1.00

Flag question

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 |

#include <stdio.h>

#include <math.h>

int main(){

float a,b,c,x1,x2,delta;

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{

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

if(delta<0){

printf("VN");

}else if(delta==0){

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

}else {

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

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

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

}

}

}

Question **26**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình thực hiện:

* Nhập hai số nguyên a, b cách nhau bởi dấu cách thể hiện tháng và năm
* In ra số ngày của tháng đó.

**Input Format**

* Hai số nguyên dương a, b cách nhau đúng một dấu cách.

**Constraints**

* 1 <= **a, b**

**Output Format**

* Nếu nhập vào a không thuộc đoạn [1, 12] hoặc b là số <=0 thì báo "Du lieu sai."
* Ngược lại in ra thông báo "Thang a nam b co X ngay."
* X là số ngày của tháng a trong năm b.
  + Lưu ý tháng 2 năm nhuận có 29 ngày
  + Năm nhuận là năm thỏa mãn a) hoặc b)

a) Nó lặp lại 4 năm 1 lần, nhưng không lặp lại mỗi 100 năm. (chia hết cho 4 nhưng không chia hết cho 100)

b) Năm nhuận cũng lặp lại mỗi 400 năm. (chia hết cho 400)

**For example:**

| **Input** | **Result** |
| --- | --- |
| 2 2022 | Thang 2 nam 2022 co 28 ngay. |

#include <stdio.h>

int main()

{

int a, b;

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

if(b>0){

switch(a){

case 1: case 3:case 5:case 7:case 8:case 10:case 12:

printf("Thang %d nam %d co 31 ngay.", a, b); break;

case 4:case 6:case 9:case 11:

printf("Thang %d nam %d co 30 ngay.", a, b); break;

case 2:

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

printf("Thang 2 nam %d co 29 ngay.", b);

else printf("Thang 2 nam %d co 28 ngay.", b); break;

default: printf("Du lieu sai.");

}

}else printf("Du lieu sai.");

}

Question **27**

Correct

Marked out of 1.00

Flag question

Question text

Nhập vào 6 điểm (x1, y1), (x2, y2), (x3, y3), (x4, y4), (x5, y5), (x6, y6). Với xi, yi là các số thực. Hãy kiểm tra 3 điểm (x1, y1), (x2, y2), (x3, y3) có thẳng hàng hay không? Sau đó kiểm tra 3 điểm (x4, y4), (x5, y5), (x6, y6) có thẳng hàng hay không?

Input:

Dòng 1: x1 y1 x2 y2 x3 y3 cách nhau dấu cách

Dòng 2: x4 y4 x5 y5 x6 y6 cách nhau dấu cách

Output:

Nếu 3 điểm thẳng hàng thì in ra chữ YES; ngược lại in ra chữ NO. Các kết quả in trên cùng dòng, cách nhau dấu cách.

Constraints:

xi, yi kiểu float

**For example:**

| **Input** | **Result** |
| --- | --- |
| 219.000000 892.000000  -664.000000 244.000000  -781.000000 194.000000  585.000000 298.000000  585.000000 298.000000  585.000000 298.000000 | NO YES |

#include <stdio.h>

int main() {

float x1, y1, x2, y2, x3, y3, x4, y4, x5, y5, x6, y6;

scanf("%f%f%f%f%f%f", &x1, &y1, &x2, &y2, &x3, &y3);

scanf("%f%f%f%f%f%f", &x4, &y4, &x5, &y5, &x6, &y6);

float check\_1 = (y2 - y1) \* (x3 - x2) - (y3 - y2) \* (x2 - x1);

float check\_2 = (y5 - y4) \* (x6 - x5) - (y6 - y5) \* (x5 - x4);

if (check\_1 == 0) {

printf("YES ");

} else {

printf("NO ");

}

if (check\_2 == 0) {

printf("YES\n");

} else {

printf("NO\n");

}

return 0;

}

Question **28**

Correct

Marked out of 1.00

Flag question

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 |

#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);

}

Question **29**

Correct

Marked out of 1.00

Flag question

Question text

Giả sử lịch học của một bạn sinh viên như sau: thứ 2, 3 học lý thuyết tại giảng đường, thứ 4, 5, 6 học thực hành tại phòng máy tính, riêng tối thứ 4 học bù môn Lập trình C, thứ 7 học ngoại ngữ tại trung tâm ngoại ngữ, chủ nhật được nghỉ.

Bạn hãy viết chương trình thực hiện công việc sau:

Nhập một số nguyên chỉ thứ trong tuần, số 2 ứng với thứ 2, số 3 ứng với thứ 3, ..., số 8 ứng với chủ nhật. In ra lịch học tương ứng của bạn sv trên.

Nếu người dùng nhập các số không đúng quy định thì in ra thông báo "Du lieu sai."

**Input Format**

* Một số nguyên chỉ thứ trong tuần.

**Constraints**

* Số nhập vào là số nguyên

**Output Format**

* In ra thông báo lịch học "Hoc ly thuyet.", "Hoc thuc hanh.", "Hoc ngoai ngu.", "Toi nay hoc bu Lap trinh C.", "Nghi.", "Du lieu sai." như yêu cầu của đề bài.
* Nếu có nhiều thông báo thì mỗi thông báo trên một dòng, thông báo nào đến trước thì ở dòng trên.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| -1\_Error | -1 | Du lieu sai. |
| 2\_Kem | 2 | Hoc ly thuyet. |
| 5\_TB | 5 | Hoc thuc hanh. |

#include <stdio.h>

int main() {

int a;

scanf("%d",&a);

if(a==2 || a==3){

printf("Hoc ly thuyet.");

}else if(a==5 || a==6){

printf("Hoc thuc hanh.");

}else if(a==4){

printf("Hoc thuc hanh.\nToi nay hoc bu Lap trinh C.");

}else if(a==7){

printf("Hoc ngoai ngu.");

}else if(a==8){

printf("Nghi.");

}else{

printf("Du lieu sai.");

}

return 0;

}

Question **30**

Correct

Marked out of 1.00

Flag question

Question text

Nhập vào 3 số thực a, b, c.

Kiểm tra 3 số đó có tạo thành 3 cạnh của một tam giác không.

Nếu đúng thì phân loại xem tam giác đó là tam giác gì trong các loại: Cân, đều, vuông, thường.

Nếu không thì đưa ra thông báo "Khong phai tam giac." //có dấu chấm ở cuối thống bào

**Input Format**

* 3 số thực

**Constraints**

* a, b, c là các số thực

**Output Format**

* In ra một trong các phân loại: **"Tam giac can."; "Tam giac deu."; "Tam giac vuong."; "Tam giac thuong."**
* Hoặc in ra thông báo lỗi: **"Khong phai tam giac."**
* **//Chú ý có dấu chấm ở cuối các thông báo. Riêng trường hợp tam giác đều thì có cả thông báo "Tam giac can." ở dòng trên.**

**Example:**

**- Input: 4 4 4**

**- Output:**

**Tam giac can.**

**Tam giac deu.**

**For example:**

| **Input** | **Result** |
| --- | --- |
| -1 2 3 | Khong phai tam giac. |

#include <stdio.h>

int main(){

float a,b,c;

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

if(a<b+c && b<a+c && c<a+b){

if(a\*a==b\*b+c\*c || b\*b==c\*c+a\*a || c\*c==a\*a+b\*b){

printf("Tam giac vuong.");

}else if(a==b || a==c || b==c){

printf("Tam giac can.");

if(a==b && b==c){

printf("\nTam giac deu.");

}

}else{

printf("Tam giac thuong.");

}

}else {

printf("Khong phai tam giac.");

}

}

Question **35**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình tính tổng 1 dãy các số nguyên.

Biết rằng số nguyên đầu tiên nhập vào bằng lệnh scanf là số lượng các giá trị còn lại được nhập vào.

Chương trình cần đọc vào 1 giá trị tương ứng với mỗi lệnh scanf.

Dữ liệu nhập vào có thể là

5 100 200 300 400 500

trong đó 5 cho biết là có 5 số sau nó cần được tính tổng.

Dữ liệu in ra phải là:

1500

// Gợi ý giải thuật không dùng kiểu dl mảng

Nhập( n );

sum = 0;

Lặp lại n lần {

Nhập( x );

sum += x;

}

In( sum );

**Input Format**

* Số nguyên đầu tiên chỉ số lượng số cần tính tổng.
* Các số nguyên tiếp theo là các số cần được tính tổng.
* Hai số cách nhau 1 kí tự trắng.

**Constraints**

* Số lượng số cần tính tổng lớn hơn hoặc bằng 0.
* Số lượng các số nhập vào phải bằng số đầu tiên.

**Output Format**

* Số nguyên chỉ tổng các số.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| 5-so | 5 100 200 300 400 500 | 1500 |

#include <stdio.h>

int main(){

int n,sum;

scanf("%d",&n);

int a[n];

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

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

sum+=a[i];

}

printf("%d",sum);

}

Question **36**

Correct

Marked out of 1.00

Flag question

Question text

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

Input:

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

Ví dụ: 5 12 hoặc 30 -8

Output:

Tổng các số chẵn tìm được. Ví dụ: 30

Constrains:

các biến kiểu nguyên

**For example:**

| **Input** | **Result** |
| --- | --- |
| 9 1 | 20 |

#include <stdio.h>

int main(){

int a,b;

int tong=0;

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

if(a>b){

int temp=a;

a=b;

b=temp;

}

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

if(i%2==0){

tong+=i;

}

}

printf("%d",tong);

}

Question **37**

Correct

Marked out of 1.00

Flag question

Question text

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

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 |

#include <stdio.h>

#include <math.h>

int main(){

int a,b;

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

int sum=0;

if(a>b){

int z=a;

a=b;

b=z;

}

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

if(i%2!=0){

sum+=i;

}

}

printf("%d",sum);

}

Question **38**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào số nguyên n.  Tính tổng sau:

2+2+...2√−−−−−−−−√−−−−−−−−−−−−√n dấu căn2+2+...2⏟n dấu căn

* Input: n là số nguyên. Ví dụ: 5 hoặc -8
* Output: Tổng S(n) như trên với độ chính xác 2 chữ số. Hoặc thông báo "Error." nếu n là số âm.
* Constrains: n kiểu nguyên, s là số thực

**For example:**

| **Input** | **Result** |
| --- | --- |
| 9 | 2.00 |

#include <stdio.h>

#include <math.h>

int main(){

long n;

scanf("%ld", &n);

if(n < 0)

{

printf("Error.");

}

else

{

double s = 0.00f;

int dem;

for(dem = 0; dem < n; dem = dem + 1)

{

s = sqrt((2 + s));

}

printf("%.2lf", s);

}

}

Question **39**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình nhập vào 2 số nguyên a, b. Tính trung bình cộng các số chẵn trong đoạn [a, b] hoặc [b, a].

Input:

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

Ví dụ: 5 12 hoặc 30 -8

Output:

Trung bình cộng các số chẵn tìm được với độ chính xác 2 chữ số thập phân. Ví dụ: 30.00

Constrains:

+ các biến a, b kiểu nguyên.

+ trung bình cộng có độ chính xác 2 chữ số thập phân.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 9 1 | 5.00 |

#include <stdio.h>

#include <math.h>

int main(){

int a,b;

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

int dem=0;

int tong=0;

if(a>b){

int temp=a;

a=b;

b=temp;

}

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

if(i%2==0){

dem++;

tong+=i;

}

}

printf("%.2f",(float)tong/dem);

}

Question **41**

Correct

Marked out of 1.00

Flag question

Question text

Nhập một số nguyên chỉ điểm ở thang 10.

In ra xếp loại dựa trên điểm như sau:

* Nếu điểm từ 0~3 in ra **Kem**
* Nếu là 4 in ra **Yeu**
* Nếu từ 5~6 in ra **TB**
* Nếu từ 7~8 in ra **Kha**
* Nếu từ 9~10 in ra **Gioi**
* Nếu điểm không thuộc phạm vi nào bên trên thì in ra **Error**

**Input Format**

* Một số nguyên chỉ điểm (**d**) theo thang 10.

**Constraints**

* 0 <= **d** <= 10

**Output Format**

* In ra một trong các phân loại: **Kem**, **Yeu**, **TB**, **Kha**, **Gioi**
* Hoặc in ra thông báo lỗi: **Error**
* In đúng chữ hoa thường, không in thêm bất kì kí tự nào khác.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| -1\_Error | -1 | Error |
| 2\_Kem | 2 | Kem |
| 5\_TB | 5 | TB |
| 8\_Kha | 8 | Kha |
| 10\_Gioi | 10 | Gioi |

#include <stdio.h>

int main(){

int a;

scanf("%d",&a);

if(a>=0 && a<=3){

printf("Kem");

}else if(a==4){

printf("Yeu");

}else if(a==5 || a==6){

printf("TB");

}else if(a==7 || a==8){

printf("Kha");

}else if(a==9 || a==10){

printf("Gioi");

}else {

printf("Error");

}

return 0;

}

Question **42**

Correct

Marked out of 1.00

Flag question

Question text

Viết chương trình tính tiền phạt khi trả sách muộn cho thư viện theo công thức sau:

* Nếu trả trước hoặc đúng ngày thì tiền phạt là 0
* Nếu trả muộn hơn ngày hẹn trả nhưng trong cùng tháng thì tiền phạt là 5000 \* số ngày trả muộn
* Nếu trả muộn hơn tháng hẹn trả nhưng trong cùng năm thì tiền phạt là 100000 \* số tháng trả muộn
* Nếu trả muộn hơn năm hẹn trả thì số tiền phạt là 1000000

Đầu vào:

* Dòng 1 chứa ba số nguyên là ngày, tháng, năm trả sách
* Dòng 2 chứa ba số nguyên là ngày, tháng, năm hẹn trả sách

Ràng buộc: Dữ liệu đầu vào đảm bảo là ngày tháng năm hợp lệ

Đầu ra: Số tiền phạt

**For example:**

| **Input** | **Result** |
| --- | --- |
| 3 4 2022  1 4 2022 | 10000 |

#include <stdio.h>

int main()

{

int ngaytra, thangtra, namtra;

int ngaymuon, thangmuon, nammuon;

int ngayhople;

scanf("%d%d%d", &ngaytra, &thangtra, &namtra);

scanf("%d%d%d", &ngaymuon, &thangmuon, &nammuon);

switch(thangtra)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

ngayhople = 31;

break;

case 4:

case 6:

case 9:

case 11:

ngayhople = 30;

break;

case 2:

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

{

ngayhople = 29;

}

else

{

ngayhople = 28;

}

break;

default:

printf("0");

return 0;

break;

}

if((ngaytra < 0) || (ngaytra > ngayhople))

{

printf("0");

return 0;

}

switch(thangmuon)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

ngayhople = 31;

break;

case 4:

case 6:

case 9:

case 11:

ngayhople = 30;

break;

case 2:

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

{

ngayhople = 29;

}

else

{

ngayhople = 28;

}

break;

default:

printf("0");

return 0;

break;

}

if((ngaymuon < 0) || (ngaymuon > ngayhople))

{

printf("0");

return 0;

}

if(namtra < nammuon)

{

printf("0");

return 0;

}

else if(namtra == nammuon)

{

if(thangtra < thangmuon)

{

printf("0");

return 0;

}

else if(thangtra == thangmuon)

{

if((ngaytra < ngaymuon) || (ngaytra == ngaymuon))

{

printf("0");

}

else

{

printf("%d", 5000 \* (ngaytra - ngaymuon));

}

}

else

{

printf("%d", 100000 \* (thangtra - thangmuon));

}

}

else

{

printf("1000000");

}

}