ID：106501001

Name：顏郁芩

Q1(OJ6)

#include<iostream>

using namespace std;

int main()

{

int a,b,c;

int arr[10]={2,3,5,7,11,13,15,17,19,23};

int count;

while(cin>>a)

{

cin>>b>>c;

int a1=a,b1=b,c1=c;

int x=1,y=1;

int i=0;

while(i<10)

{

while(a1%arr[i]==0&&b1%arr[i]==0&&c1%arr[i]==0)

{

a1=a1/arr[i];

b1=b1/arr[i];

c1=c1/arr[i];

x=x\*arr[i];

}

i=i+1;

}

y=x;

int j=0;

while(j<10)

{

count=0;

if(a1%arr[j]==0)

count=count+1;

if(b1%arr[j]==0)

count=count+1;

if(c1%arr[j]==0)

count=count+1;

if(count==2)

{

if(a1%arr[j]==0)

a1=a1/arr[j];

if(b1%arr[j]==0)

b1=b1/arr[j];

if(c1%arr[j]==0)

c1=c1/arr[j];

y=y\*arr[j];//y是最小公倍數

}

j++;

}

y=y\*a1\*b1\*c1;

cout<<x<<" "<<y<<endl;

}

return 0;

}

Q2

#include<iostream>

using namespace std;

int ever(int,int);

int stair(int);

void array(void);

int arr[100];

void array()

{

for(int j=0;j<100;j++)

arr[j]=j+1;

}

int main()

{

int n;

array();

while(cin>>n)

{

int i=0;

while(i<=n)

{

int a=1,b=1,c=1,d,e;

if(i==0)

{

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

c=c\*arr[j];

a=1;

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

b=b\*arr[j];

}

else

{

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

c=c\*arr[j];

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

a=a\*arr[j];

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

b=b\*arr[j];

}

d=a\*b;

e=c/d;

cout<<e<<" ";

i++;

}

cout<<endl;

}

}

Q3.

pass by value =用新的記憶體位置與名稱紀錄相同數值

pass by reference=雖然變數名稱不同不過所使用的儲存位置相同

Q4.

假設有三個數a.b.c

int x[3]:x[0]=a,x[1]=b,x[2]=c-----變數存成x陣列

int x1, x2, x3:x1=a,x2=b,x3=c----分散的三個變數

Q5.

第一個輸入的值是接下來會有幾個數字，第二個是正確答案，接下來的數字要依照→猜中間數的規則去猜，也就是第一個猜的數(輸入的第三數)要是0~10的中間數5。

接下來比較正確答案，假設為K，比較K跟5的關係。

如果K界在0~5之間下一數就要猜0~5的中間數，但因為0~5有六個數，中間數會是2.5所以要猜比較大的那個數也就是3；反之如果界在5~10之間就要猜8。

以此判斷下去。

會出現的狀況：(先把次數跟答案令為X,Y)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 所有可能出現的輸入 | Y值 | X值應為 | 次數應為 | 輸出FALSE條件 |
| X Y 5 8 9 10 | 10 | 5 | 4 | 1. X不是應該輸入的值(猜到也不會停下，或是次數不足以猜到值) 2. 輸入方式不在列表內(沒有照著取中間值的規定去猜) |
| X Y 5 8 9 | 9 | 4 | 3 |
| X Y 5 8 | 8 | 3 | 2 |
| X Y 5 8 7 | 7 | 4 | 3 |
| X Y 5 8 7 6 | 6 | 5 | 4 |
| X Y 5 | 5 | 2 | 1 |
| X Y 5 3 4 | 4 | 4 | 3 |
| X Y 5 3 | 3 | 3 | 2 |
| X Y 5 3 2 | 2 | 4 | 3 |
| X Y 5 3 2 1 | 1 | 5 | 4 |
| X Y 5 3 2 1 0 | 0 | 6 | 5 |