Bài 1: Dùng hàm thực hiện các yêu cầu sau:

Viết chương trình thực hiện các yêu cầu sau:

Nhập vào 3 số thực a, b, c từ bàn phím.

a. Kiểm tra a, b, c có lập thành ba cạnh của tam giác không?

b. Nếu có hãy kiểm tra đây là tam giác cân, đều, vuông hay tam giác thường.

Tính chu vi và diện tích của tam giác.

In kết quả ra màn hình.

Bài 2: Dùng hàm thực hiện các chương trình sau:

1. Hàm kiểm tra 1 số có phải là nguyên tố không?
2. In ra các số nguyên tố <n
3. Hàm kiểm tra 1 số có là số hoàn hảo không? (số hoàn hảo là số có tổng các ước = chính nó)
4. In ra các số hoàn hảo <n
5. Hàm tìm USCLN của hai số nguyên
6. Nhập vào 2 số nguyên tương ứng là tử số và mẫu số của phần số, in ra màn hình phân số tối giản

#include <conio.h>

#include <stdio.h>

// Hàm ki?m tra s? hoàn ch?nh

int ktSoHoanChinh(int n){

int sum = 0;

//Ch?y vòng l?p t? 1 d?n s? n dã cho

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

// Ð?i v?i m?i s? di qua

// Ki?m tra xem n có chia h?t cho s? dó hay không.

if (n % i == 0)

sum += i;

}

if(sum == n)

return 1;

return -1;

}

main(){

int n;

printf("Nhap vao so nguyen duong n = ");

scanf("%d", &n);

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

{

if(ktSoHoanChinh(i) ==1)

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

}

}

Bài 3: dùng hàm thực hiện các chương trình sau:

1. Giải và biện luận phương trình : Ax2 +bx+c=0
2. Giải hệ phương trình : ax+by=c

Dx+ey=f

Bài 4: Viết chương trình thực hiện các yêu cầu sau:

Nhập vào một số nguyên n.

a. Đếm xem số đó có bao nhiêu chữ số và tính tổng của chúng. In kết quả ra màn hình.

b. Phân tích số đó thành tích của các số nguyên tố.

VD: N=12. Phân tích N=2\*2\*3.

Bài tập về Mảng

Bài 6: dùng hàm xử lý các công việc sau:

1. chương trình nhập vào một dãy số, sau đó in ra màn hình các số âm ở

trên một dòng và các số dƣơng ở trên một dòng.

1. Sắp xếp các số âm ở đầu theo thứ tự tăng dần, số dương ở cuối giảm dần
2. Viết chương trình tìm phần tử âm cuối cùng của dãy
3. Sắp xếp dãy tăng dần
4. Đưa các số dương của dãy số cho vào một mảng b và các số âm của dãy vào một mảng c

Bài 8: Cho mảng một chiều A, n phần tử (n<100). Xây dựng chương trình thực hiện các công việc sau:

* 1. Nhập mảng
  2. Xuất mảng
  3. Đếm số phần tử âm
  4. Tìm phần tử Max và Min của mảng
  5. Tìm phần tử âm lớn nhất và dương bé nhất của mảng
  6. Liệt kê các số nguyên tố trong mảng
  7. Liệt kê các số chính phương trong mảng
  8. Sắp xếp mảng theo thứ tự tăng dần
  9. Sắp xếp mảng theo thứ tự giảm dần
  10. Thêm/Xóa/Sửa một phần tử vào mảng
  11. Tách mảng thành hai mảng: mảng na lưu các số nguyên tố, mảng nb lưu các số không là nguyên tố. in ra màn hình

Câu 9. Viết chƣơng trình nhập vào một số n sau đó in ra màn hình ma trận xoáy

cấp n có dạng nhƣ sau:

Giả sử với n = 4

1 2 3 4

12 13 14 5

11 16 15 6

10 9 8 7

Bài tập ma trận hai chiều

Bài 7: Cho hai dãy số a1, a2, …, an. và b1, b2,…, bn

1. Gộp xen kẽ 2 mảng trên

#include<stdio.h>

void nhapmang(int a[], int n, char ch)

{

printf("\nNhap vao gia tri mang: %c", ch);

int i;

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

{

printf("\n Phan tu thu: %d la : ",i);

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

}

}

void xuatmang(int a[], int n, char ch)

{

printf("\n Mang %c: vua nhap: ",ch);

int i;

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

{

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

}

}

void gopmangxenke(int a[],int b[],int c[],int n)

{

int j=0;

int i;

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

{

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

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

}

}

main()

{

int a[100],b[100],c[200], n;

printf("Nhap so phan tu cua mang: ");

scanf("%d",&n);

nhapmang(a,n,'A');

xuatmang(a,n,'A');

nhapmang(b,n,'B');

xuatmang(b,n,'B');

gopmangxenke(a,b,c,n);

xuatmang(c,2\*n,'C');

}

1. Sắp xếp 2 dãy trên đều được sắp xếp theo thứ tự tăng dần. Viết chƣơng trình để từ hai dãy số trên xây dựng một dãy mới cũng theo thứ tự tăng dần.

* Bài 8: Viết chương trình nhập vào một ma trận A với kích thước NxM và xuất ma trận ra màn hình
* Yêu cầu: Xuất ra màn hình các phần tử A[i][j] sao cho A[i][j] là phần tử lớn nhất trên dòng i và nhỏ nhất trên cột

#include <stdio.h>

#include <conio.h>

#define maxd 50

#define maxc 50

void nhapmt(int a[][maxc],int \*d, int \*c)

{

int i,j;

for ( i=0;i<\*d;i++)

{ for( j=0; j<\*c; j++)

{

printf("\n Phan tu thu i,j la: ",i,j) ;

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

}

}

}

void xuatmt(int a[][maxc], int n,int m)

{

int i,j;

printf("Ma tran:\n");

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

{

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

printf(" %3d",a[i][j]);

printf("\n");

}

}

//ham tim phan tu max tren 1 dong

int Maxdong(int a[][maxc],int m,int \*i )

{ int j,max=a[\*i][0];//dong co dinh

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

if(a[\*i][j]>max) max=a[\*i][j] ;

return max ;

}

//ham tim phan tu min tren 1 cot

int Mincot(int a[][maxc],int n,int \*j )

{ int i,min=a[0][\*j];

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

if(a[i][\*j]<min) min=a[i][\*j];

return min ;

}

void main()

{

int a[maxd][maxc], n, m, i, j;

printf("\nNhap so dong: ");scanf("%d",&n);

printf("\nNhap so cot: ");scanf("%d",&m);

printf("Moi nhap kich thuoc ma tran \n");

nhapmt(a,&n,&m);

xuatmt(a,n,m);

printf("\nCac phan tu co Max dong va Min cot la: ");

for(i=0;i<n;i++) //xet toan bo ma tran neu co phan tu a[i][j]nao

for(j=0;j<m;j++)//bang maxdong va bang mincot thi in ra

if((a[i][j]==Maxdong(a,m,&i))&&(a[i][j]==Mincot(a,n,&j)))

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

}

* Bài 9: Cho ma trận A với kích thước NxM. Thực hiện các yêu cầu sau:
  1. Yêu cầu:
     1. Tính tổng các phần tử trong ma trận
     2. Liệt kê các phần tử chia hết cho 7
     3. Tìm tổng các phần tử nằm trên đường viền của ma trận
     4. Tìm tổng các phần tử nằm trên phần tam giác trên của ma trận
     5. Tìm cột có tích max
     6. Tìm dòng có nhiều số chính phương nhất
     7. Tìm dòng nào được sắp xếp tăng hoặc giảm
     8. Hoán vị ma trận thành MxN
  2. Tạo menu cho phép người dùng chọn các chức năng cần thực hiện
  3. #include<stdio.h>
  4. #include<conio.h>
  5. #include<stdlib.h>
  6. const int maxc 50;
  7. const int maxd 50;
  8. //Dùng ham nhap mang, xuatmang
  9. void Nhapmang(int a[][n],int d, int c)
  10. {
  11. for(int i=0;i<d;i++)
  12. for(int j=0;j<c;j++)
  13. {
  14. printf("Doc phan tu % d, %d: ",i,j);
  15. //a[i][j]
  16. scanf("%d",&a[i][j]);
  17. }
  18. }
  19. void Xuatmang(int a[][n],int d, int c)
  20. {
  21. for(int i=0;i<d;i++)
  22. {
  23. for(int j=0;j<c;j++)
  24. printf("%d ",a[i][j]);
  25. printf("\n");
  26. }
  27. }
  28. int TinhTongMT(int a[maxd][maxd],int n, int m)
  29. {
  30. int kq=0;
  31. for (int i=0; i<n; i++)
  32. for (int j=0; j<m; j++)
  33. kq+=a[i][j];
  34. return kq;
  35. }
  36. void LietKeChiaHet7(int a[maxd][maxc], int n, int m)
  37. {
  38. int soPT=0;
  39. for (int i=0; i<n; i++)
  40. for (int j=0; j<m; j++)
  41. if (a[i][j]%7==0)
  42. {
  43. cout <<" "<<a[i][j];
  44. soPT++;
  45. }
  46. if (soPT==0)
  47. cout <<"\n Khong co phan tu chia het cho 7 trong ma tran";
  48. }
  49. int TongDuongVien(int a[maxd][maxc], int n, int m)
  50. {
  51. int tong=0, i;
  52. for (i=0; i<m; i++) // tinh tong hang dau tien va cuoi cung
  53. {
  54. tong+=a[0][i];
  55. tong+=a[n][i];
  56. }
  57. for (i=1; i<m; i++)
  58. {
  59. tong+=a[i][0];
  60. tong+=a[i][m];
  61. }
  62. return tong;
  63. }
  64. int TongTamGiacTren(int a[maxd][maxc], int n, int m)
  65. {
  66. int tong=0;
  67. if (n!=m)
  68. {
  69. cout <<"\n Chuc nang nay chi tinh cho ma tran vuong";
  70. return -1;
  71. }
  72. else
  73. for (int i=0; i<n; i++)
  74. tong+=a[i][i];
  75. return tong;
  76. }
  77. void TimCotCoTichMax(int a[MaxC][MaxD], int n, int m)
  78. {
  79. int tichCot=1, tichMax=1, cotTichMax=0;
  80. for (int i=0; i<m; i++)
  81. {
  82. for (int j=0; i<n; j++)
  83. tichCot\*=a[j][i];
  84. if (tichCot>tichMax)
  85. {
  86. tichMax= tichCot;
  87. cotTichMax = i;
  88. }
  89. }
  90. cout <<cotTichMax<<" "<<tichMax;
  91. }
  92. void TimDongNhieuChinhPhuong(int a[MaxC][MaxD], int n, int m)
  93. {
  94. // mang chua so luong so chinh phuong cua moi dong
  95. int dong[Max];
  96. int dongcantim=0;
  97. int i,j;
  98. for (i=0; i<n; i++) //khoi tao mang 0
  99. dong[i]=0;
  100. for (i=0; i<n; i++)
  101. for (j=0; j<m; j++)
  102. if (sqrt(a[i][j]) == (int)sqrt(a[i][j]))
  103. dong[i]++;
  104. // Tim dong co nhieu so chinh phuong nhat
  105. for (i=0; i<n-1; i++)
  106. if (dong[i]<dong[i+1])
  107. dongcantim = i+1;
  108. cout <<"\n Dong nhieu so chinh phuong nhat, bao gom so: "<<dongcantim<<" "<<dong[dongcantim];
  109. }
  110. void HoanViMT(int a[MaxC][MaxD], int n, int m)
  111. {
  112. int MT2[MaxC][MaxD];
  113. int i,j;
  114. for (i=0; i<n; i++)
  115. for (int j=0; j<m; j++)
  116. MT2[i][j] = a[j][i];
  117. cout <<"\n Ma tran hoan vi la \n";
  118. for (i=0; i<n; i++)
  119. {
  120. for (j=0; j<m; j++)
  121. cout<<" "<<MT2[i][j];
  122. cout<<endl;
  123. }
  124. }
  125. void ChuyenMaTranThanhMang(int a[MaxC][MaxD], int n, int m)
  126. {
  127. int Max1Chieu, i, j, k=-1, temp;
  128. int mang[200];
  129. for (i=0; i<n; i++)
  130. for (j=0; j<m; j++)
  131. mang[++k]= a[i][j];
  132. for (i=0; i<k-1; i++)
  133. for (j=i+1;j<k; j++)
  134. if (mang[i]<mang[j])
  135. {
  136. temp=mang[i];
  137. mang[i]=mang[j];
  138. mang[j]=temp;
  139. }
  140. for (i=0; i<k; i++)
  141. cout<<" "<< mang[i];
  142. }
  143. void TimPhanTuMax(int a[MaxC][MaxD], int n, int m)
  144. {
  145. int max, dong, cot;
  146. max=a[0][0];
  147. for (int i=0; i<n; i++)
  148. for (int j=0; j<m; j++)
  149. if (a[i][j]>max)
  150. {
  151. max=a[i][j];
  152. dong=i;
  153. cot=j;
  154. }
  155. cout<<"\n Phan tu max la phan tu a["<<dong<<"]["<<cot<<"] co gia tri la "<<max;
  156. }
  157. void TimMinMoiCot(int a[MaxC][MaxD], int n, int m)
  158. {
  159. int min[Max], temp=0, chiso=-1;
  160. for (int i=0; i<m; i++)
  161. {
  162. temp=a[m][0]; // Gan min cua moi cot la o dau tien cua cot/
  163. for (int j=0; j<n; j++)
  164. if (a[i][j]<temp)
  165. temp=a[i][j];
  166. min[++chiso] = temp; // Gan gia tri min cua moi cot vao mang 1 chieu
  167. }
  168. cout <<"\n Phan tu min cua moi cot la: ";
  169. for (int k=0; k<chiso; k++)
  170. cout<<" "<<min[k];
  171. }

BÀI 10: Viết chƣơng trình nhập vào hai ma trận An\*m và Bm\*p. Tính ma trận tích

Cn\*p = An\*m \* Bm\*p sau đó in cả ba ma trận ra màn hình.

//Nhap mang 2 chieu doc tu ban phim

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

const int n=50;

//Dùng ham nhap mang, xuatmang

void Nhapmang(int a[][n],int d, int c,char ch)

{

printf("\n Moi ban nhap ma tran: %c",ch);

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

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

{

printf("\nDoc phan tu % d, %d: ",i,j);

//a[i][j]

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

}

}

void Xuatmang(int a[][n],int d, int c,char ch)

{

printf("\n xuat mang %c\n",ch);

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

{

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

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

printf("\n");

}

}

void tonghaimatran(int a[][n], int b[][n], int tong[][n], int d, int c )

{

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

{

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

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

}

}

void tichmatran(int a[][n], int b[][n], int tong[][n], int d, int c ,int p)

{

int i,j,k;

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

for(k=0;k<p;k++)

{

tong[i][k]=0;

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

tong[i][k]=a[i][j]+b[j][k];

}

}

void phantuduongcheochinh(int a[][50], int d,int c)

{

int s=0;

printf("\ncac phan tu tren duong cheo chinh: ");

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

{

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

if (i==j ) {

printf("%4d",a[i][i]);

s=s+a[i][j];}

}

printf("\n tong cac phan tu tren cheo chinh: %d",s);

}

void phantuduongcheophu(int a[][50], int d, int c)

{

int s=0;

printf("\ncac phan tu tren duong cheo chinh: ");

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

{

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

if (i+j==d-1 ) {

printf("%4d",a[i][j]);

s=s+a[i][j];}

}

printf("\n tong cac phan tu tren cheo phu: %d",s);

}

void tongvien(int a[][50],int d, int c)

{

int s=0;

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

{

s=s+a[0][i];

}

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

{

s=s+a[d-1][i];

}

for(int i=1;i<d-1;i++)

{

s=s+a[d][1];

}

for(int i=1;i<d-1;i++)

{

s=s+a[i][c-1];

}

printf("\nTong la: %d",s);

}

main()

{

int a[n][n],b[n][n],tong[n][n];

int d,c,p;

//Vao so phan tu mang

printf("Nhap so d= ");scanf("%d",&d);

printf("Nhap so c= ");scanf("%d",&c);

printf("Nhap so p= ");scanf("%d",&p);

Nhapmang(a,d,c,'A');

printf("\n MANG VUA NHAP\n");

Xuatmang(a,d,c,'A');

Nhapmang(a,c,p,'B');

printf("\n MANG VUA NHAP\n");

Xuatmang(a,c,p,'B');

tonghaimatran(a,b,tong,d,p);

Xuatmang(tong,d,p,'tong');

tichmatran(a,b,tong,d,c,p);

Xuatmang(tong,d,p,'tich');

/\* phantuduongcheochinh(a,d,c);

phantuduongcheophu(a,d,c);

tongvien(a,d,c);\*/

getch();

}

Bài 11: Câu 18. Viết chƣơng trình nhập vào ma trận An\*n sau đó thực hiện các công việc sau:

a. Kiểm tra xem ma trận vừa nhập có phải là ma trận đơn vị hay không.

b. Tính tổng của các phần tử nằm trên hàng thứ i, với i nhập từ bàn phím

(1<= i <= n).

c. Tính tích của các phần tử nằm trên cột thứ j, với j nhập vào từ bàn phím

(1<= j <= n).

d. Kiểm tra xem trên hàng thứ i có chứa số nguyên tố hay không? (i = 1..n)

và hàng nào có chứa nhiều số nguyên tố nhất.

#include<conio.h>

#include<stdio.h>

void nhapmang(int n, int A[100][100])

{

int i, j;

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

{

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

{

printf("nhap A[%d][%d] = ", i, j);

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

}

}

}

void xuatmang(int n, int A[100][100])

{

int i, j;

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

{

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

{

printf("%3d", A[i][j]);

}

printf("\n");

}

}

/\*kiem tra ma tran vua nhap co phai la ma tran don vi khong\*/

int donvi(int n, int A[100][100])

{

int i, j;

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

{

if(A[i][i] !=1)

{

return 0;

}

}

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

{

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

{

if((A[i][j] !=0) || (A[j][i] !=0))

{

return 0;

}

}

}

return 1;

}

void tonghang(int a[][100], int n, int \*i)

{

int j;

int s=0;

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

{

s=s+a[\*i][j];

}

printf("\n Tong hang s= %d",s);

}

void tichcot(int a[][100],int n, int \*c)

{

int i, s=1;

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

{

s=s\*a[i][\*c];

}

printf("\Tich la s: %d",s);

}

int ktnguyento(int n)

{

int i;

if (n==1) return 0;

if (n==2||n==3) return 1;

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

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

return 1;

}

int kthangnguyento(int a[][100], int n, int \*h)

{

int j;

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

{

if (ktnguyento(a[\*h][j]==1)) return 1;

}

return 0;

}

void timdonghieungto(int a[][100], int n)

{

// mang chua so luong so chinh phuong cua moi dong

int dem[100]={0};

int i,j;

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

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

if (ktnguyento(a[i][j])==1)

dem[i]++;

int Max = dem[0];

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

{Max = (Max > dem[i]) ? Max : dem[i]; }

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

if(Max == dem[i])

{printf("\nDong %d co so nguyen to nhieu nhat: %d so nguyen to", i, Max);

}

}

int main()

{

int n, A[100][100] ;

printf("nhap mang\n");

scanf("%d", &n);

nhapmang(n, A);

xuatmang(n, A);

if(donvi(n, A)==0)

{

printf("mang vua nhap khong phai la ma tran don vi");

}

if(donvi(n, A) ==1)

{

printf(" mang vua nhap la ma tran don vi");

}

int h,c;

do{

printf("\n Nhap hang: "); scanf("%d",&h);}while (h<-1||h>n);

tonghang(A,n,&h);

do{

printf("\n Nhap cot: "); scanf("%d",&c);}while (c<-1||c>n);

tichcot(A,n,&c);

if (kthangnguyento(A,n,&h)==1) printf("\nhang %d co so nguyen to",h);

else printf("\n %d khong co so nguyen to",h);

timdonghieungto(A,n);

getch();

return 0;

}