ÔN TẬP KĨ THUẬT LẬP TRÌNH

Phần 1: CÁC THUẬT TOÁN THÔNG DỤNG

A. Mảng một chiều:

- Khai báo: <kiểu dữ liệu> <tên mảng>[<số phần tử>];
- Truy cập vào một phần tử: <tên mảng>[<chỉ số>]
- Nhập dữ liệu cho mảng một chiều các số nguyên:

```
void Nhapmang(int a[],int n)
{
	for (int i=0;i<n;i++)
	{
		printf("Nhap a[%d] = ",i);
		scanf("%d",&a[i]);
	}
}
```

- Xuất dữ liệu cho mảng một chiều các số nguyên:

```
void Xuatmang(int a[],int n)
{
	for (int i=0;i<n;i++) printf("%d\t",a[i]);
}
```

- Tính tổng các phần tử của mảng:

```
int Tong(int a[], int n)
{
	int t=0;
	for(int i=0; i<n; i++)  t=t+a[i];
	return t;
}
```

- Tính TBC các phần tử trong mảng:

```
float TBC(int a[], int n)
{
	int  tong=0;
	for (int i=0; i<n; i++)  tong+=a[i];
	return (float)tong/n; //ep kieu
}
```

- Tìm và in ra màn hình các phần tử chẵn trong mảng.

```
void PhanTuChan(int a[], int n)
{
	printf("\nDanh sach cac phan tu chan trong mang la:");
	for(int i=0; i<n; i++)
		if(a[i] % 2 == 0) printf("%3d",a[i]);
}
```

- Tìm và in ra màn hình các phần tử chẵn dương trong mảng.

```
void PhanTuChanDuong(int a[], int n)
```

```
{
        printf("\nDanh sach cac phan tu chan duong trong mang la:");
        for(int i=0; i<n; i++)
                if(a[i] % 2 == 0 && a[i] > 0)  printf("%3d",a[i]);
}
```

- Tìm và in ra màn hình các phần tử là số nguyên tố có trong mảng.

```
int ktNguyenTo(int x)
{
        int uoc=0;
        for(int i=1; i<=x; i++)
                if (x%i==0) uoc++;
        if (uoc==2) return 1;
  return 0;
}
void inNguyenTo(int a[], int n)
{
        printf("\nDanh sach cac phan tu chan duong trong mang la:");
        for(int i=0; i<n; i++)
              if(ktNguyenTo(a[i] == 1) printf("%3d",a[i]);
}
```

- Tìm phần tử lớn nhất có trong mảng.

```
void TimMax(int a[], int n)
{
        int max = a [0];
        for(int i = 0; i < n ; i++)
                if(a[i] > max) max = a[i];
        printf("\nPhan tu lon nhat trong mang la: %d",max);
}
```

- Tìm phần tử nhỏ nhất có trong mảng.

```
void TimMin(int a[], int n)
{
        int min = a [0];
        for(int i = 0; i < n ; i++)
                if(a[i] < min) min = a[i];
        printf("\nPhan tu lon nhat trong mang la: %d",max);
}
```

- Sắp xếp tăng dần:

```
void HoanVi(int &a, int &b)
{
        int t;
        t = a; a = b; b = t;
}
void SapXepTang(float a[], int n)
{
        for(int i = 0; i < n-1; i++)
              for(int j = i+1; j < n; j++)
```

```
                if(a[i] > a[j]) HoanVi(a[i],a[j]);
}
```

B. Ma trận:

- Khai báo: <kiểu dữ liệu> <tên ma trận>[<số dòng tối đa>][<số cột tối đa>];
- Truy cập một phần tử: <tên ma trận>[<chỉ số dòng>][<chỉ số cột>]
- Nhập số dòng, số cột:

```
void Enter(int &m,int &n)
{
	do
	{
		printf("Nhap so dong cua ma tran: ");
		scanf("%d",&m);
		if (m<1||m>100) printf("Nhap sai. Moi nhap lai.\n");
	}
	while (m<1||m>100);
	do
	{
		printf("Nhap so cot cua ma tran: ");
		scanf("%d",&n);
		if (n<1||n>100) printf("Nhap sai. Moi nhap lai.\n");
	}
	while (n<1||n>100);
}
```

- Nhập ma trận:

```
void Read(int a[][100],int m,int n)
{
	for (int i=0;i<m;i++)
		for (int j=0;j<n;j++)
		{
			printf("\tNhap a[%d][%d] = ",i,j);
			scanf("%d",&a[i][j]);
		}
}
```

- Xuất ma trận:

```
void Display(int a[][100],int m,int n)
{
	for (int i=0;i<m;i++)
	{
		for (int j=0;j<n;j++) printf("%d\t",a[i][j]);
		printf("\n");
	}
}
```

- Tính tổng các phần tử trong ma trận:

```
long Sum(int a[][100],int m,int n)
```

```
{
	long s=0;
	for (int i=0;i<m;i++)
		for (int j=0;j<n;j++)
			s+=a[i][j];
	return s;
}
```

- Tổng các phần tử trên đường chéo chính:

```
long MainDiagonal(int a[][100],int m)
{
	long s=0;
	for (int i=0;i<m;i++) s+=a[i][i];
	return s;
}
```

- Tổng các phần tử trên đường chéo phụ:

```
long DiagonalSide(int a[][100],int m,int n)
{
	long s=0;
	for (int i=0;i<m;i++) s+=a[i][n-1-i];
	return s;
}
```

- TBC các số dương:

```
float TBC(int a[][100],int m,int n)
{
	int count=0,s=0;
	for (int i=0;i<m;i++)
		for (int j=0;j<n;j++)
			if (a[i][j]>0)
			{
				count++;
				s+=a[i][j];
			}
	return (float) (s/count);
}
```

- Đếm số phần tử bằng x (cho trước):

```
int Count(int a[][100],int m,int n,int x)
{
	int c=0;
	for (int i=0;i<m;i++)
		for (int j=0;j<n;j++)
			if (a[i][j]==x) c++;
	return c;
}
```

- Hàm **memset**: dùng để khởi tạo giá trị ban đầu cho các phần tử của mảng 1 chiều/ma trận, trong thư viện <memory.h>. Cách dùng**: memset(<tên mảng/ma trận>,<giá trị cần đặt>,sizeof(<tên mảng>));** Ví dụ: khởi tạo mảng a gồm 10 phần tử có giá trị 0: int a[10]; memset(a,0,sizeof(a));
- Xoay ma trận:
  + Xoay 90^0 cùng chiều kim đồng hồ:

```
for (int k=0; k <n; k++)
{
        for (int j=0; j<m; j++) printf("%4d",a[m-1-j][k]);
        printf("\n");
}
```

  + Xoay 90^0 ngược chiều kim đồng hồ:

```
for (int k = 0; k <n; k++)
{
        for (int j=0; j<m; j++) printf("%4d",a[j][n-1- k]);
        printf("\n");
}
```

  + Xoay 180^0:

```
for (int k = 0; k < m; k++)
{
        for (int j=0; j<n; j++) printf("%4d",a[m-1-k][n-1-j]);
        printf("\n");
}
```

- Sắp xếp tăng dần:

```
void Sapxeptang(int a[][100],int m,int n)
{
        for (int i=0;i<m*n;i++)
                for (int j=0;j<m*n;j++)
                        if (a[i/n][i%c]<a[j/n][j%n]) Traodoi(a[i/n][i%n],a[j/n][j%n]);
}
```

C. Kí tự:

- Khai báo: char <tên biến>;
- Thư viện: <string.h>
- Nhập kí tự:
  + C1: scanf(“%c”,&<tên biến>);
  + C2: getch(<tên biến>);
  + C3: char ch=getche();
- Xuất kí tự:
  + C1: putch(<tên biến>); //có thể in ra màu khi dùng với textcolor(<mã màu từ 1 – 15>);
  + C2: printf(“%c”,<tên biến>);
  + C3: putc(<tên biến>);

D. Chuỗi (xâu) kí tự:

- Khai báo: char <tên biến>[<số kí tự tối đa>]; //nhớ +1 cho NULL
- Gán hằng: char <tên biến>[]= “<giá trị cần gán>”;

- Thư viện: <string.h>
- Nhập/xuất chuỗi dùng <stdio.h>
    + Nhập: scanf("%s",&<tên biến>); //không nhận dấu space, tab.
    + Xuất: printf("%s",<tên biến>);
- Nhập/xuất chuỗi dùng <string.h>
    + Nhập: gets(<tên biến>); //phải xoá bộ đệm bằng flushall(); hay fflush(stdin); trong <stdlib.h>.
    + Xuất: puts(<tên biến>); //xuất xong tự xuống dòng.
- Các hàm xử lí chuỗi (trong <string.h>):
    - Nối chuỗi: strcat(<tên chuỗi 1>,<tên chuỗi 2>); //kết quả chứa trong chuỗi 1.
    - Tìm lần xuất hiện đầu tiên của kí tự <x> trong chuỗi <a>: strchr(a,x);
    - So sánh: strcmp(<chuỗi 1>,<chuỗi 2>); // bằng nhau: ==0, lớn hơn: >0, bé hơn: <0.
    - Sao chép: strcpy(<chuỗi đích>,<chuỗi nguồn>);
    - Sao chép 1 phần: strncpy(<chuỗi đích>,<chuỗi nguồn>,<số kí tự cần chép>);
    - Tìm kiếm: strstr(<chuỗi 1>,<chuỗi 2>); //tìm sự xuất hiện đầu của chuỗi 2 trong chuỗi 1.
    - Lấy chiều dài: strlen(<tên chuỗi>);
    - Chuỗi thường -> hoa: strupr(<tên chuỗi>);
    - Chuỗi hoa -> thường: strlwr(<tên chuỗi>);
    - Chuỗi sang số: atoi(<tên chuỗi>); atol(<tên chuỗi>); atof(<tên chuỗi>);
    - Đảo chuỗi: strrev(<tên chuỗi>);

E. Kiểu dữ liệu có cấu trúc:

- Là một kiểu dữ liệu do người dùng tự định nghĩa. Bao gồm nhiều thành phần (trường – field), mỗi trường có một kiểu dữ liệu khác nhau.
- Cú pháp định nghĩa:

```
struct <tên cấu trúc>
{
        <kiểu> <trường 1>;
        <kiểu> <trường 2>;
        …
        <kiểu> <trường n>;
};
```

- Khai báo: typedef struct <tên struct> <tên kiểu mới>;
- Truy cập vào từng thành phần: <tên struct>.<tên trường>
- Với kiểu dữ liệu là số thực, khi nhập, ta không nhập trực tiếp mà phải thông qua biến tạm.
- Mảng cấu trúc: là một mảng có từng phần tử là 1 struct.
- Khi nhập dữ liệu cho trường là kiểu chuỗi, phải xoá bộ đệm bằng flushall(); hay fflush(stdin);

F. Kiểu con trỏ:

- Khai báo: <kiểu dữ liệu> *****<tên biến con trỏ>
- Gán địa chỉ : phép &<biến> ➔ <biến>.
- Lấy giá trị: <biến> ➔ *<biến>.
- Cấp phát/thu hồi: trong <alloc.h>
    + <biến con trỏ> = (<tên kiểu>*) malloc(<kích thước>); / free(<biến con trỏ>);

+ <biến con trỏ> = (<tên kiểu>*) calloc(<kích thước>,sizeof(<kiểu>)); / free(<biến con trỏ>);
+ <biến con trỏ> = new <kiểu dữ liệu>[<kích thước>]; / delete [] <biến con trỏ>;

- Cấp phát thêm: <biến con trỏ> = realloc(<biến con trỏ>,<kích thước cần cấp phát lại>);
- Con trỏ và mảng 1 chiều:
+ &<tên mảng>[0] ➔ <tên mảng>.
+ &<tên mảng>[<vị trí>] ➔ (<tên mảng> + <vị trí>).
+ <tên mảng>[<vị trí>] ➔ *(<tên mảng> + <vị trí>).

Hàm cấp phát bộ nhớ:

```
void Init(int *p,int n)
{
	p=(int*) calloc(n,sizeof(int));
	if (p==NULL)
	{
		printf("Khong du bo nho.\n");
		getch(); exit(1);
	}
}
```

- Con trỏ và ma trận:

Cách 1: con trỏ đơn cấp: *<tên biến con trỏ>.
+ <tên ma trận>[<chỉ số hàng>][<chỉ số cột>] ➔ *(<tên ma trận> + <chỉ số hàng>*<số cột> + <chỉ số cột>).
+ &<tên ma trận>[<chỉ số hàng>][<chỉ số cột>] ➔ (<tên ma trận> + <chỉ số hàng>*<số cột> + <chỉ số cột>).

Hàm cấp phát bộ nhớ:

```
void Init(int *p,int m,int n)
{
	p=(int*) calloc(m*n,sizeof(int));
	if (p==NULL)
	{
		printf("Khong du bo nho.\n");
		getch(); exit(1);
	}
}
```

Cách 2: con trỏ đa cấp: **<tên biến con trỏ>.
+ <tên ma trận>[<chỉ số hàng>][<chỉ số cột>] ➔ *(*(<tên ma trận> + <chỉ số hàng>) + <chỉ số cột>).
+ &<tên ma trận>[<chỉ số hàng>][<chỉ số cột>] ➔ (*(<tên ma trận> + <chỉ số hàng>) + <chỉ số cột>).

Hàm cấp phát bộ nhớ:

```
void Init(int **a,int m,int n)
{
	a=(int**) calloc(m,sizeof(int*));
	if (a==NULL)
	{
		printf("Khong du bo nho.\n");
```

```
                getch(); exit(1);
        }
        for (int i=0;i<m;i++)
        {
                a[i]=(int*) calloc(n,sizeof(int));
                if (a[i]==NULL)
                {
                        printf("Khong du bo nho.\n");
                        getch(); exit(1);
                }
        }
}
```

Hàm giải phóng bộ nhớ:

```
void MyFree(int **a,int m)
{
        for (int i=0;i<m;i++) free(a[i]);
        free(a);
}
```

- Con trỏ và kiểu dữ liệu có cấu trúc:

+ <tên struct>.<tên trường> ➔ <tên con trỏ struct> -> <tên trường> hoặc (*<tên con trỏ struct>).<tên trường>.

+ &<tên truct>.<tên trường> ➔ &<biến con trỏ struct> -> <tên trường>.

+ Truyền structure sang hàm: với hàm nhập, khi gọi tên hàm phải thêm &: <tên hàm>(&<tham số>,…);

G. Đệ quy:

- Đệ quy tuyến tính:

```
void <tên hàm>
{
        if <điều kiện dừng> return <giá trị hay kết thúc>; //phần neo
        else
        {
                //một số công việc
                //gọi đệ quy đến <tên hàm> //phần đệ quy
        }
}
```

- Đệ quy nhị phân:

```
void <tên hàm>
{
        if <điều kiện dừng> return <giá trị hay kết thúc>; //phần neo
        else
        {
                //làm một số công việc
                //gọi đệ quy đến <tên hàm> để giải quyết vấn đề nhỏ hơn       //phần đệ quy
                // gọi đệ quy đến <tên hàm> để giải quyết các vấn đề còn lại  //phần đệ quy
        }
```

```
}
```

- Đệ quy phi tuyến:

```
void <tên hàm>
{
	for (int i=1;i<=n;i++)
	{
		//làm một số công việc
		if <điều kiện dừng> return <giá trị hay kết thúc>; //phần neo
		else //gọi đệ quy đến <tên hàm> //phần đệ quy
	}
}
```

- Đệ quy hỗ tương ➔ bên trong hàm này có lời gọi hàm kia.
- Giải toán bằng đệ quy:
  + Thông số hoá bài toán.
  + Tìm phần neo.
  + Tìm giải thuật gọi đệ quy lui dần về phần neo.
- Với các giải thuật đệ quy trên mảng, ta **giảm dần** số phần tử của mảng (<mảng>[0->n-1 phần tử], phân tích thành [0;n-2 phần tử], tìm phần neo với [n-1] và khi n={0,1}).
- Sắp xếp mảng 1 chiều bằng đệ quy:

Thuật toán QuickSort (sắp xếp nhanh):

```
#define Swap(type,a,b) {type tmp=a; a=b; b=tmp;}
void QuickSort(int a[],int l,int r)
{
	int key = a[(l+r)/2];
	int i=l,j=r;
	while(i <= j)
	{
		while(a[i]<key) i++;
		while(a[j]>key) j--;
		if(i <= j)
		{
			if (i<j) Swap(int,a[i],a[j]);
			i++; j--;
		}
	}
	if (l<j) QuickSort(a,l,j);
	if (i<r) QuickSort(a,i,r);
}
```

➔ Lời gọi hàm: QuickSort(<tên mảng>,0,<số phần tử>);

H. Tệp tin (File): (xét với tệp tin văn bản)

- Khai báo: FILE *<tên biến tệp tin>;
- Mở tệp tin: <tên biến tệp tin> = fopen("<đường dẫn", "<chế độ>");

➔ Chế độ:

Chế độ	Ý nghĩa
r	Mở tập tin văn bản để đọc
w	Tạo ra tập tin văn bản mới để ghi
a	Nối vào tập tin văn bản
r+	Mở một tập tin văn bản để đọc/ghi
w+	Tạo ra tập tin văn bản để đọc ghi
a+	Nối vào hay tạo mới tập tin văn bản để đọc/ghi

- Đóng tệp tin: fclose(<tên biến tệp tin>); ➔ fcloseall(); //đóng tất cả các tệp tin.
- Kiểm tra đến cuối tệp: feof(<tên biến tệp tin>); //==EOF nếu cuối tệp, ngược lại ==0.
- Di chuyển con trỏ về đầu tệp: rewind(<tên biến tệp tin>);
- Đọc dữ liệu từ tệp tin:
  + getc(<tên biến tệp tin>); //trả về mã ASCII của kí tự trong tệp tin liên kết với biến con trỏ.
  + fgets(<biến đích>,<độ dài chuỗi>,<tên biến tệp tin>);
  + fscanf(<tên biến tệp tin>, "<định dạng>",<danh sách biến>);
- Ghi dữ liệu lên tệp tin:
  + putc(<tên biến kí tự>,<tên biến tệp tin>);
  + puts(<tên biến chuỗi>,<tên biến tệp tin>);
  + fprintf(<tên biến tệp tin>, "<định dạng>",<danh sách biến>);
- Quy trình làm việc với tệp tin:

  Khai báo ➔ Mở file ➔ Đọc/ghi ➔ Đóng file.

Phần 2: CÁC BÀI ÔN TẬP THEO CHỦ ĐỀ

➔ Cấu trúc sinh viên:

```
#include"stdio.h"
#include"conio.h"
#include"string.h"
#include"stdlib.h"
struct  sinhvien
{
        char masv[10];
        char hoten[30];
        int namsinh;
        float dtb;
};
void nhap1sv(sinhvien &a)
{
        printf("\nnhap ma sv: ");
        flushall();
        gets(a.masv);
        printf("\nnhap ho ten sv: ");
        flushall();
```

```
        gets(a.hoten);
        printf("\nnhap nam sinh: ");
        scanf("%d",&a.namsinh);
        printf("\nnhap diem trung binh: ");
        float d;
        scanf("%f",&d);
        a.dtb=d;

}

void nhapmangsv(sinhvien sv[], int &n)
{
        printf("\nnhap so luong sv: ");
        scanf("%d", &n);
        for(int i=0; i<n;i++)
        {
            printf("\nnhap sv thu %d", i);
            nhap1sv(sv[i]);
        }
}
void xuat1sv(sinhvien a)
{

        printf("\nma sv: ");
        puts(a.masv);
        printf("\nho ten: ");
        puts(a.hoten);
        printf("\nnam sinh %d",a.namsinh);
        printf("\ndiem trung binh %.1f",a.dtb);

}
void xuatmangsv(sinhvien sv[], int n)
{
        printf("\nthong tin sv da nhap");
        for(int i=0; i<n; i++)
        {
                printf("\nsv thu %d la: ", i);
                xuat1sv(sv[i]);
                printf("\n----------");
        }
}
void main()
{
    clrscr();
        sinhvien sv[20];
        int n;
        nhapmangsv(sv, n);
        xuatmangsv(sv,n);
        getch();
}
```

➔ Cấu trúc phân số:

```
#include <stdio.h>
#include <conio.h>
#include <memory.h>
```

```
struct PS
{
	int tu;
	int mau;
};


void Enter(PS &x)
{
	printf("\tNhap tu so: ");
	scanf("%d",&x.tu);
	printf("\tNhap mau so: ");
	scanf("%d",&x.mau);
}

void Read(PS a[],int n)
{
	for (int i=0;i<n;i++)
	{
		Enter(a[i]);
		printf("-----\n");
	}
}

void Output(PS x)
{
	printf("%d/%d",x.tu,x.mau);
}

void Display(PS a[],int n)
{
	for (int i=0;i<n;i++)
	{
		Output(a[i]);
		printf("\t");
	}
}

int UCLN(int a,int b)
{
	while (b>0)
	{
		a=a%b;
		int tmp=a; a=b; b=tmp;
	}
	return a;
}

PS Rutgon(PS &x)
{
	int tmp=UCLN(x.tu,x.mau);
	x.tu=x.tu/tmp;
	x.mau=x.mau/tmp;
}
```

```
PS Cong(PS a,PS b)
{
	PS c;
	c.tu=a.tu*b.mau+a.mau*b.tu;
	c.mau=a.mau*b.mau;
	Rutgon(c);
	return c;
}

void Find(PS a[],int n)
{
	float b[100];
	memset(b,0,sizeof(b));
	for (int i=0;i<n;i++) b[i]=(float)a[i].tu/a[i].mau;
	float u=b[0],v=b[0]; int uu=0,vv=0;
	for (int i=1;i<n;i++)
	{
		if (b[i]>u)
		{
			u=b[i];
			uu=i;
		}
		if (b[i]<v)
		{
			v=b[i];
			vv=i;
		}
	}
	printf("\nPhan so lon nhat: ");
	Output(a[uu]);
	printf("\nPhan so nho nhat: ");
	Output(a[vv]);
}

int CountMax(PS a[],int n)
{
	float b[100];
	memset(b,0,sizeof(b));
	for (int i=0;i<n;i++) b[i]=(float)a[i].tu/a[i].mau;
	float tmp=b[0]; int res=0;
	for (int i=1;i<n;i++)
		if (b[i]>tmp) tmp=b[i];
	for (int i=0;i<n;i++)
		if (b[i]==tmp) res++;
	return res;
}

void main()
{
	clrscr();
	fflush(stdin);
	PS a[100];int n;
	printf("Nhap so luong phan so: ");
	scanf("%d",&n);
```

```
	Read(a,n);
	printf("Day phan so vua nhap:\n");
	Display(a,n);
	PS res;
	res.tu=a[0].tu; res.mau=a[0].mau;
	for (int i=1;i<n;i++) res=Cong(res,a[i]);
	printf("\nTong cac phan so: ");
	Output(res);
	Find(a,n);
	printf("\nSo phan tu lon nhat la: %d",CountMax(a,n));
	getch();
}
```

➔ Đệ quy:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>

#define Nm 100
#define Swap(type,a,b) {type tmp=a; a=b; b=tmp;}

void Enter(int &n)
{
	do
	{
		printf("NHap n = ");
		scanf("%d",&n);
		if (n<=0) printf("Nhap sai. Moi nhap lai.\n");
	}
	while (n<=0);
}

void EnterArray(int a[],int n)
{
	if (n>0)
	{
		EnterArray(a,n-1);
		printf("\tNhap a[%d] = ",n-1);
		scanf("%d",&a[n-1]);
	}
}

void Display(int a[],int n)
{
	if (n>0)
	{
		Display(a,n-1);
		printf("%d\t",a[n-1]);
	}
}

long EvenSum(int a[],int n)
{
	if (n==0) return 0;
	if (a[n-1]%2==0) return a[n-1]+EvenSum(a,n-1);
```

```
	return EvenSum(a,n-1);
}

int IsPrime(int x)
{
	if (x<2) return 0;
	for (int i=2;i<=(int)sqrt(x);i++)
		if (x%i==0) return 0;
	return 1;
}

int PrimeCount(int a[],int n)
{
	if (n==0) return 0; int c=0;
	if (IsPrime(a[n-1])==1) c++;
	return c+PrimeCount(a,n-1);
}

int FindMax(int a[],int n)
{
	if (n==1) return a[0];
	if (a[n-1]>FindMax(a,n-1)) return a[n-1];
	return FindMax(a,n-1);
}

void QuickSort(int a[],int l,int r)
{
	int key = a[(l+r)/2];
	int i=l,j=r;
	while(i <= j)
	{
		while(a[i]<key) i++;
		while(a[j]>key) j--;
		if(i <= j)
		{
			if (i<j) Swap(int,a[i],a[j]);
			i++; j--;
		}
	}
	if (l<j) QuickSort(a,l,j);
	if (i<r) QuickSort(a,i,r);
}

int IsSquare(int x)
{
	return ((int)sqrt(x)==sqrt(x) ? 1:0);
}

void Square(int a[],int n)
{
	if (n==0) return;
	if (IsSquare(a[n-1])==1) printf("%d\t",a[n-1]);
	Square(a,n-1);
}
```

```
/* main program */

int main()
{
        clrscr();
        int a[Nm],n;
        Enter(n);
        EnterArray(a,n);
        printf("Mang vua nhap la:\n");
        Display(a,n);
        printf("\nTong cac so chan: %ld",EvenSum(a,n));
        printf("\nSo cac so nguyen to: %d",PrimeCount(a,n));
        printf("\nSo lon nhat la: %d",FindMax(a,n));
        QuickSort(a,0,n);
        printf("\nMang vua sap xep la:\n");
        Display(a,n);
        printf("\nCac so chinh phuong:\n");
        Square(a,n);
        getch();
        return 0;
}
```

➔ Con trỏ với ma trận:

```
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
#include <math.h>
#include <process.h>

void Enter(int &m,int &n)
{
   do
   {
        printf("Nhap so dong cua ma tran: ");
        scanf("%d",&m);
        if (m<1||m>100) printf("Nhap sai. Moi nhap lai.\n");
   }
   while (m<1||m>100);
   do
   {
        printf("Nhap so cot cua ma tran: ");
        scanf("%d",&n);
        if (n<1||n>100) printf("Nhap sai. Moi nhap lai.\n");
   }
   while (n<1||n>100);
}

void Init(int **a,int m,int n)
{
   a=(int**) calloc(m,sizeof(int*));
   if (a==NULL)
   {
        printf("Khong du bo nho.\n");
        getch(); exit(1);
   }
```

```
    for (int i=0;i<m;i++)
    {
        a[i]=(int*) calloc(n,sizeof(int));
        if (a[i]==NULL)
        {
            printf("Khong du bo nho.\n");
            getch(); exit(1);
        }
    }
}

void MyFree(int **a,int m)
{
    for (int i=0;i<m;i++) free(a[i]);
    free(a);
}

void EnterArray(int **a,int m,int n)
{
    for (int i=0;i<m;i++)
        for (int j=0;j<n;j++)
        {
            printf("\tNhap a[%d][%d] = ",i,j);
            scanf("%d",(*(a+i)+j));
        }
}

void Display(int **a,int m,int n)
{
    for (int i=0;i<m;i++)
    {
        for (int j=0;j<n;j++) printf("%d\t",*(*(a+i)+j));
        printf("\n");
    }
}

long ArraySum(int **a,int m,int n)
{
    long tmp=0;
    for (int i=0;i<m;i++)
        for (int j=0;j<n;j++) tmp+=*(*(a+i)+j);
    return tmp;
}

int IsPrime(int a)
{
    if (a<2) return 0;
    for (int i=2;i<=(int)sqrt(a);i++)
        if (a%i==0) return 0;
    return 1;
}

long PrimeSum(int **a,int m,int n)
{
    long tmp=0;
```

```
    for (int i=0;i<m;i++)
        for (int j=0;j<n;j++)
            if (IsPrime(*(*(a+i)+j))==1) tmp+=*(*(a+i)+j);
    return tmp;
}

int MainDianoside(int **a,int m)
{
    int tmp=0;
    for (int i=0;i<m;i++) tmp+=*(*(a+i)+i);
    return tmp;
}

int SideDianoside(int **a,int m)
{
    int tmp=0;
    for (int i=0;i<m;i++) tmp+=*(*(a+i)+m-1-i);
    return tmp;
}

float Average(int **a,int m,int n)
{
    int tmp=0, count=0;
    for (int i=0;i<m;i++)
        for (int j=0;j<n;j++)
            if (*(*(a+i)+j)>0)
            {
                tmp+=*(*(a+i)+j);
                count++;
            }
    return (float)tmp/count;
}

int Count(int **a,int m,int n,int x)
{
    int count=0;
    for (int i=0;i<m;i++)
        for (int j=0;j<n;j++)
            if (*(*(a+i)+j)==x) count++;
    return count;
}

int MaxArr(int **a,int m,int n)
{
    int tmp=**a;
    for (int i=0;i<m;i++)
        for (int j=0;j<n;j++)
            if (*(*(a+i)+j)>tmp) tmp=*(*(a+i)+j);
    return tmp;
}

int MinArr(int **a,int m,int n)
{
    int tmp=**a;
    for (int i=0;i<m;i++)
```

```
        for (int j=0;j<n;j++)
                if (*(*(a+i)+j)<tmp) tmp=*(*(a+i)+j);
    return tmp;
}

int MaxRow(int **a,int n,int i)
{
    int tmp=*(*(a+i));
    for (int j=0;j<n;j++)
        if (*(*(a+i)+j)>tmp) tmp=*(*(a+i)+j);
    return tmp;
}

int MinCollum(int **a,int m,int j)
{
    int tmp=*(*(a)+j);
    for (int i=0;i<m;i++)
        if (*(*(a+i)+j)<tmp) tmp=*(*(a+i)+j);
    return tmp;
}

int MaxNegative(int **a,int m,int n)
{
    int tmp=1;
    for (int i=0;i<m;i++)
        for (int j=0;j<n;j++)
            if (*(*(a+i)+j)<0)
            {
                tmp=*(*(a+i)+j);
                break;
            }
    for (i=0;i<m;i++)
        for (j=0;j<n;j++)
            if (*(*(a+i)+j)<0 && *(*(a+i)+j)>tmp)
                tmp=*(*(a+i)+j);
    return tmp;
}

int PrimeCount(int **a,int n,int i)
{
    int count=0;
    for (int j=0;j<n;j++)
        if (IsPrime(*(*(a+i)+j))==1) count++;
    return count;
}

int PrimeRow(int **a,int m,int n)
{
    int res=-1;
    int tmp=0;
    for (int i=0;i<m;i++)
        if (PrimeCount(a,n,i)>tmp)
        {
            res=i;
            tmp=PrimeCount(a,n,i);
```

```
        }
    return res;
}

/* main program */

int main()
{
    clrscr();
    int **a,m,n;
    Enter(m,n);
    Init(a,m,n);
    EnterArray(a,m,n);
    printf("Mang vua nhap la:\n");
    Display(a,m,n);
    printf("\nTong cac phan tu cua ma tran: %ld",ArraySum(a,m,n));
    printf("\nTong cac phan tu nguyen to cua ma tran: %ld",PrimeSum(a,m,n));
    if (m==n)
    {
        printf("\nTong cac phan tu tren duong cheo chinh: %d",MainDianoside(a,m));
        printf("\nTong cac phan tu tren duong cheo phu: %d",SideDianoside(a,m));
    }
    else printf("\nKhong co tong duong cheo chinh & phu.");
    printf("\nTBC cac so duong cua ma tran: %.1f",Average(a,m,n));
    int x;
    printf("\nNhap vao so can dem: ");
    scanf("%d",&x);
    printf("So lan xuat hien cua %d la: %d",x,Count(a,m,n,x));
    printf("\nPhan tu lon nhat trong mang: %d",MaxArr(a,m,n));
    printf("\nPhan tu nho nhat trong mang: %d",MinArr(a,m,n));
    printf("\nNhap dong can tim: ");
    scanf("%d",&x);
    printf("Phan tu lon nhat tren dong %d la: %d",x,MaxRow(a,n,x));
    printf("\nNhap cot can tim: ");
    scanf("%d",&x);
    printf("Phan tu nho nhat tren cot %d la: %d",x,MinCollum(a,m,x));
    if (MaxNegative(a,m,n)==1) printf("\nMang khong co so am.");
    else printf("\nPhan tu am co gia tri lon nhat: %d",MaxNegative(a,m,n));
    if (PrimeRow(a,m,n)==-1) printf("\nMang khong co SNT.");
    else printf("\nDong co nhieu SNT nhat: %d",PrimeRow(a,m,n));
    MyFree(a,m);
    getch();
    return 0;
}
```

➔ Thao tác trên con trỏ:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define ESC 27

void Capphat(int *a,int n)
{
        a=new int[n];
        for(int i=0;i<n;i++)
```

```
                a[i]=0;
        return;
}

void Mydelete(int *a)
{
        delete [] a;
        return;
}

void Nhap(int *a,int &n)
{
        do
        {
                printf("\nNhap so luong phan tu: ");
                scanf("%d", &n);
                if(n<1 || n>100 )
                {
                        clrscr();
                        printf("\nNhap sai! Nhap lai");
                }
        }while(n<1 || n>100);
        for(int i=0;i<n;i++)
        {
                printf("\nNhap phan tu thu a[%d]: ",i+1);
                scanf("%d", &a[i]);
        }
}

void Xuat(int *a,int n)
{
        for(int i=0;i<n;i++)
        {
                printf("%4d", a[i]);
        }
}

void Themdau(int *a,int &n)
{
        int x;
        printf("\n\tNhap phan tu muon them: ");
        scanf("%d", &x);
        for(int i=n;i>0;i--)
        {
                a[i]=a[i-1];
        }
        a[0]=x;
        n++;
        Capphat(a,n);
        printf("\nDa them phan tu %d vao dau mang.",x);
}

void Themvt(int *a,int &n)
{
```

```
	int x,vt;
	printf("\n\tNhap phan tu muon them: ");
	scanf("%d", &x);
	printf("\n\tNhap vi tri can them: ");
	scanf("%d", &vt);
	if(vt>=0 && vt<n)
	{
		for(int i=n;i>vt;i--)
		{
			a[i]=a[i-1];
		}
		a[vt]=x;
		n++;
		Capphat(a,n);
		printf("\nDa them phan tu %d vao vi ti %d trong mang.",x,vt);
	}
	else
		printf("\nVi tri vuot qua gioi han!");
}

void Themcuoi(int *a,int &n)
{
	int x;
	printf("\n\tNhap phan tu muon them: ");
	scanf("%d", &x);
	a[n]=x;
	n++;
	Capphat(a,n);
	printf("\nDa them phan tu %d vao cuoi mang.",x);
}

void Xoadau(int *a,int &n)
{
	for(int i=0;i<n-1;i++)
		a[i]=a[i+1];
	n--;
	Capphat(a,n);
	printf("\n\tDa Xoa phan tu dau tien ra khoi mang.\n");
}

void Xoavt(int *a,int &n)
{
	int vt;
	printf("\n\tNhap vi tri muon xoa: ");
	scanf("%d", &vt);
	if(vt>=0 && vt<n)
	{
		for(int i=vt;i<n-1;i++)
			a[i]=a[i+1];
		n--;
		Capphat(a,n);
		printf("\nDa Xoa phan tu o vi tri %d ra khoi mang.\n",vt);
	}
	else
		printf("\nVi tri vuot qua gioi han!\n");
```

```
}

void Xoacuoi(int *a,int &n)
{
	a[n]=a[n-1];
	n--;
	Capphat(a,n);
	printf("\nDa Xoa phan tu cuoi cung ra khoi mang.\n");
}

int max(int *a,int n)
{
	int max=a[0];
	for(int i=0;i<n;i++)
		if(a[i]>max)
			max=a[i];
	return max;
}

void xoa(int *a,int &n,int k)
{
	for(int i=k;i<n-1;i++)
		a[i]=a[i+i];
	n--;
	Capphat(a,n);
}

void Xoamax(int *a,int &n)
{
	for(int i=0;i<n;i++)
	{
		if(a[i]==max(a,n))
		{
			xoa(a,n,i);
			i--;
		}
	}
	printf("\nDa xoa het phan tu max ra khoi mang.\n");
}

char menu()
{
	clrscr();
	printf("\n\t!^^MOT SO THAO TAC TREN MANG 1 CHIEU SU DUNG CON TRO^^!\n\n");
	printf("\tChon mot trong cac chuc nang:\n");
	printf("\t\t1. Nhap mang.\n");
	printf("\t\t2. Xuat mang.\n");
	printf("\t\t3. Them phan tu.\n");
	printf("\t\t4. Xoa phan tu.\n");
	printf("\n\tNhan ESC de thoat chuong trinh\n");
	return getch();
}

void Chuongtrinh()
{
```

```
clrscr();
int *a,n;
char ch;
do
{
	ch=menu();
	if(ch=='1')
	{
		Nhap(a,n);
		Capphat(a,n);
		getch();
	}
	if(ch=='2')
	{
		printf("\nDanh sach cac phan tu trong mang:\n\n");
		Xuat(a,n);
		getch();
	}
	if(ch=='3')
	{
		printf("\n\t\tTHEM PHAN TU VAO MANG\n\n");
		printf("\nChon mot chuc nang:\n");
		printf("\t\t3.1 Them dau.\n");
		printf("\t\t3.2 Them vao vi tri bat ki.\n");
		printf("\t\t3.3 Them cuoi.\n");
		char t=getch();
		if(t=='1')
			Themdau(a,n);
		if(t=='2')
			Themvt(a,n);
		if(t=='3')
			Themcuoi(a,n);
		getch();
	}
	if(ch=='4')
	{
		printf("\n\t\tXOA PHAN TU RA KHOI MANG\n\n");
		printf("\t\tChon mot trong cac chuc nang:\n");
		printf("\t\t4.1 Xoa dau.\n");
		printf("\t\t4.2 Xoa vi tri can xoa.\n");
		printf("\t\t4.3 Xoa cuoi.\n");
		printf("\t\t4.4 Xoa max.\n");
		char t=getch();
		if(t=='1')
			Xoadau(a,n);
		if(t=='2')
			Xoavt(a,n);
		if(t=='3')
			Xoacuoi(a,n);
		if(t=='4')
			Xoamax(a,n);
		getch();
	}
}while(ch!=ESC);
Mydelete(a);
```

```
	getch();
}

void main()
{
	clrscr();
	Chuongtrinh();
	getch();
}
```

➔ Một số hàm đệ quy:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
//---------
void nhap(int &n)
{
	do
	{
		printf("\n Nhap n: ");
		scanf("%d",&n);
		if(n<=0 || n>100) printf("\n Nhap sai. Nhap lai.");
	}while(n<=0 || n>100);
}
//---------------
int nhapMang(int a[],int n)
{
	if(n == 0)
	return 0;
	nhapMang(a,n-1);
	printf("\n a[%d]= ",n-1);
	scanf("%d",&a[n-1]);
	return 0;
}
//---------------------
int xuatMang(int a[],int n)
{
	if(n == 0)
	return 0;
	xuatMang(a,n-1);
	printf("%4d",a[n-1]);
	return 0;
}
// xuat mang nguoc
int xuatMangN(int a[],int n)
{
	if(n == 0)
	return 0;
	printf("%4d",a[n-1]);
	xuatMang(a,n-1);
	//return 0;
}
//---------- Tong ----------
long tinhTong(int a[],int n)
{
```

```
	if(n == 0) return 0;
	long s = tinhTong(a,n-1);
	return s = s + a[n-1];
}
long tongChan(int a[],int n)
{
	if(n == 0)
	return 0;
	long s=tongChan(a,n-1);
	if(a[n-1]%2 == 0) s = s + a[n-1];
	return s;
}
long tongLe(int a[],int n)
{
	if(n == 0)
	return 0;
	long s = tongLe(a,n-1);
	if(a[n-1]%2 != 0) s = s + a[n-1];
	return s;
}
//---------- So nguyen to ----------
int ktnt(int n)
{
	int dem=0,i;
	for(i=1;i<=n;i++)
		if(n%i == 0) dem++;
	if(dem == 2) return 1;
	return 0;
}
long tongNT(int a[],int n)
{
	if(n == 0)
	return 0;
	long s= tongNT(a,n-1);
	if(ktnt(a[n-1]) == 1) s = s + a[n-1];
	return s;
}
//--------- tim le chan cuoi
int leCuoi(int a[],int n)
{
	if(n == 0)
	return 0;
	leCuoi(a,n-1);
	if(a[n-1]%2 != 0) return a[n-1];
}
int chanDau(int a[],int n)
{
	if(n==0)return 1;
	int cd=chanDau(a,n-1);
	printf("\nchandau %d \t %d",cd,a[n-1]);
	if(cd!=1)return cd; else
	if(a[n-1]%2==0) return a[n-1];
	return 1;
}
//-----so chinh phuong
```

```
int ktra(int x)
{
	if(sqrt(x)==(int)sqrt(x)) return 1;
	return 0;
}
int cpCuoi(int a[],int n)
{
	if(n==0) return 0;
	cpCuoi(a,n-1);
	int tmp=a[n-1];
	if(tmp == 1) tmp=a[n-1];
	return tmp;
}
//------- tim Max
int timMax(int a[],int n)
{
	if(n==0) return 0;
	int t=timMax(a,n-1);
	if(t<a[n-1]) t=a[n-1];
	return t;
}
//------ dem Chan
int demChan(int a[],int n)
{
	int dem=0;
	if(n==0) return 0;
	dem=demChan(a,n-1);
	if(a[n-1]%2==0) dem=dem+1;
	return dem;
}
int demMax1(int a[],int n,int max)
{
	int dem;
	if(n==0) return 0;
	dem=demMax1(a,n-1,max);
	if(a[n-1]==max) dem++;
	return dem;
}
int vtMaxDau1(int a[],int n)
{
	if(n==0) return -1;
	if(n==1) return 0;
	int vt=vtMaxDau1(a,n-1);
	if(a[n-1] > a[vt]) return n-1;
	else return vt;
}
//---------- sx tang mang
void hoanvi(int &a,int &b)
{
	int tmp;
	tmp=a; a=b; b=tmp;
}
void sxep(int a[],int n)
{
	if(n>1)
```

```
	{
		sxep(a,n-1);
		if(a[n-1]<a[n-2])
		{
			hoanvi(a[n-1],a[n-2]);
			sxep(a,n-1);
		}
	}
}

//--------------
void main()
{
	clrscr();
	int a[100];int n,chon;
	printf("\n 1. Nhap mang 1 chieu.");
	printf("\n 2. Xuat mang 1 chieu.");
	printf("\n 3. Tinh tong mang.");
	printf("\n 4. Tinh tong cac phan tu chan.");
	printf("\n 5. Tinh tong cac phan tu le.");
	printf("\n 6. Tinh tong cac phan tu nguyen to trong mang.");
	printf("\n 7. Tim phan tu chinh phuong cuoi cung trong mang.");
	printf("\n 8. Phan tu lon nhat trong mang.");
	printf("\n 9. Dem so phan tu chan trong mang.");
	printf("\n 10. Sap xep mang tang dan.");
	printf("\n 11. Dem so phan tu lon nhat.");
	do
	{
		printf("\n -------Nhap cong viec can lam: ");
		scanf("%d",&chon);
		switch(chon)
		{
			case 1:
			{
				nhap(n);
				nhapMang(a,n);
				break;
			}
			case 2:
			{
				printf("\n Mang sau khi nhap la: \n");
				xuatMang(a,n);
				break;
			}
			case 3:
			{
				printf("\n Tong mang: %ld",tinhTong(a,n));
				break;
			}
			case 4:
			{
				printf("\n Tong chan: %ld",tongChan(a,n));
				break;
			}
			case 5:
```

```
                {
                        printf("\n Tong le: %ld",tongLe(a,n));
                        break;
                }
                case 6:
                {
                        long nt = tongNT(a,n);
                        if(nt == 0) printf("\n Khong co so nguyen to trong mang");
                        else printf("\n Tong so nguyen to trong mang: %ld",nt);
                        break;
                }
                case 7:
                {
                        int t1 = cpCuoi(a,n);
                        if(t1 == 0) printf("\n Khong co so chinh phuong trong mang");
                        else printf("\n Chinh phuong cuoi la: %ld",t1);
                        break;
                }
                case 8:
                {
                        printf("\n Phan tu lon nhat trong mang la: %d",timMax(a,n));
                        break;
                }
                case 9:
                {
                        int tmp=demChan(a,n);
                        if(tmp==0) printf("\n Khong co phan tu chan trong mang.");
                         else printf("\n So phan tu chan trong mang la: %d ",tmp);
                        break;
                }
                case 10:
                {
                        sxep(a,n);
                        xuatMang(a,n);
                        break;
                }
                case 11:
                {
                        int max=timMax(a,n);
                        printf("\nDem so ptln la: %d",demMax1(a,n,max));
                        break;
                }
                default:
                {
                        chon=0;
                        break;
                }
        }
    } while(chon!=0);
    getch();
}
```

---HẾT---

Phước Nguyễn ©, April 18, 2016