Mảng 1 chiều

```
void khoiTao(int a[], int n)
{
 srand(time(NULL));
 for(int i = 0; i < n; i++)
       a[i] = rand()%100; //+ rand()%10;
}

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

int nhoNhat(int a[], int n) // tim phan tu nho nhat
{
 int min = a[0];
 for(int i = 0; i < n; i++)
       if(a[i] < min)
             min = a[i];
 return min;
}

int lonNhat(int a[], int n) // tim phan tu lon nhat
{
 int max = a[0];
 for(int i = 0; i < n; i++)
       if(a[i] > max)
             max = a[i];
 return max;
}

int tong(int a[], int n) // tong cac phan tu trong
mang
{
 int S = 0;
 for(int i = 0; i < n; i++)
       S += a[i];
 return S;
}

int tongChan(int a[], int n) // tong cac phan tu
chan trong mang
{
 int S = 0;
 for(int i = 0; i < n; i++)
       if(a[i]%2 == 0)
             S += a[i];
 return S;
}

int tongLe(int a[], int n) // tong cac phan tu le
trong mang
{
 int S = 0;
 for(int i = 0; i < n; i++)
       if(a[i]%2 != 0)
             S += a[i];
 return S;
}
/* kiem tra so nguyen to
 return dem
 dem = 1 : la so nguyen to
 dem != 1 : khong la so nguyen to
*/
bool kiemTraSNT(int x)
{
 if(x<=2) return false;
 int dem = 1;
 for(int j = 2; j <= sqrt(x); j++)
       if(x%j == 0)
             return false;
 if(dem==1)
       return true;
}

int tongSNT(int a[], int n) // tong cac phan tu la
so nguyen to trong mang
{
 int S = 0;
 for(int i = 0; i < n; i++)
       if(kiemTraSNT(a[i]))
       {
             printf("%d ", a[i]);
             S += a[i];
       }
 return S;
}

void xoaTaiViTri(int a[], int &n, int index) // xoa
phan tu tai vi tri
{
 for(int i = index;i <n ; i++)
 {
       a[i] = a[i+1];
 }
 n--;
}



void xoaVTLe(int a[], int &n) // xoa cac phan tu tai
vi tri le
{
 for(int i = n-1;i >=0 ; i--)
       if((i+1)%2!=0)
             xoaTaiViTri(a,n,i);
}

int viTri(int a[], int n, int x) // ham xac dinh vi
tri cua phan tu x
{
 for(int i = 0;i < n ; i++)
             if(a[i] == x)
                   return i;
}
```

```
int demMax(int a[], int n, int max) // ham dem gia tri max co trong mang
{
 int tam = 0;
 for(int i = 0;i < n ; i++)
              if(a[i] == max)
                     tam++;
 return tam;
}

void xoaGiaTriLonNhat(int a[], int &n)
{
 int max = lonNhat(a, n);
 printf("\n\tGia tri lon nhat: %d",max);
 int dem = demMax(a, n, max);
 printf("\n\tSo luong: %d\n",dem);
 while(dem != 0)
 {
       xoaTaiViTri(a,n,viTri(a, n, max));
       dem--;
 }
}

void themTaiViTri(int a[], int &n, int index, int x) // them gia tri x tai vi tri
{
 n++;
 for(int i = n-1;i >=index ; i--)
 {
       a[i] = a[i-1];
 }
 a[index]=x;
}

void themSauPTLonNhat(int a[], int &n, int x){
 themTaiViTri(a,n,viTri(a,n,lonNhat(a,n))+1,x);
}

void themSauPTLaSNT(int a[], int &n, int x){
 int indexSNT=-1;
 for(int i=0; i<n; i++)
       if(kiemTraSNT(a[i])){
              indexSNT=i;
       }
 if(indexSNT!=-1)
       themTaiViTri(a,n,indexSNT+1,x);
 else
       printf("\n\tKhong tim thay SNT trong mang.");
}

void sapXepGiam(int a[], int n){
 for(int i=0; i<n; i++)
       for(int j=0; j<n; j++)
              if(a[i]>a[j]){
                     int temp=a[i];
                     a[i]=a[j];
                     a[j]=temp;
              }
}
```

```
void noiMang(int a[], int na, int b[], int nb
                        ,int c[], int &nc){
	nc=0;
	for(int i=0; i<na;i++){
		c[nc]=a[i];
		nc++;
	}
	for(int i=0; i<nb;i++){
		c[nc]=b[i];
		nc++;
	}
}

void noiMangXenKe(int a[], int na, int b[], int nb
                        ,int c[], int &nc){
	nc=0;
	if(na>=nb)
		for(int i=0; i<na; i++){
			c[nc]=a[i];
			nc++;
			if(i<nb){
				c[nc]=b[i];
				nc++;
			}
		}
	else
		for(int i=0; i<nb; i++){
			c[nc]=b[i];
			nc++;
			if(i<na){
				c[nc]=a[i];
				nc++;
			}
		}
}
```

```
int nhoNhat(int a[], int n) // tim phan tu nho nhat
{
	int min = a[0];
	for(int i = 0; i < n; i++)
		if(a[i] < min)
			min = a[i];
	return min;
}

int lonNhat(int a[], int n) // tim phan tu lon nhat
{
	int max = a[0];
	for(int i = 0; i < n; i++)
		if(a[i] > max)
			max = a[i];
	return max;
}

int tong(int a[], int n) // tong cac phan tu trong
mang
{
	int S = 0;
	for(int i = 0; i < n; i++)
		S += a[i];
	return S;
}

int tongChan(int a[], int n) // tong cac phan tu
chan trong mang
{
	int S = 0;
	for(int i = 0; i < n; i++)
		if(a[i]%2 == 0)
			S += a[i];
	return S;
}

int tongLe(int a[], int n) // tong cac phan tu le
trong mang
{
	int S = 0;
	for(int i = 0; i < n; i++)
		if(a[i]%2 != 0)
			S += a[i];
	return S;
}
/* kiem tra so nguyen to
	return dem
	dem = 1 : la so nguyen to
	dem != 1 : khong la so nguyen to
*/
bool kiemTraSNT(int x)
{
	int dem = 1;
	if(x<=2) return false;
	for(int j = 2; j <= sqrt(x); j++)
		if(x%j == 0)
			return false;
	if(dem==1)
		return true;
}

int tongSNT(int a[], int n) // tong cac phan tu la
so nguyen to trong mang
{
	int S = 0;
	for(int i = 0; i < n; i++)
		if(kiemTraSNT(a[i]))
		{
			printf("%d ", a[i]);
			S += a[i];
		}
	return S;
}

void xoaTaiViTri(int a[], int &n, int index) // xoa
phan tu tai vi tri
{
	for(int i = index;i <n ; i++)
	{
		a[i] = a[i+1];
	}
	n--;
}



void xoaVTLe(int a[], int &n) // xoa cac phan tu
tai vi tri le
{
```

```
	for(int i = n-1;i >=0 ; i--)
		if((i+1)%2!=0)
			xoaTaiViTri(a,n,i);
}

int viTri(int a[], int n, int x) // ham xac dinh vi tri cua phan tu x
{
	for(int i = 0;i < n ; i++)
		if(a[i] == x)
			return i;
}

int demMax(int a[], int n, int max) // ham dem gia tri max co trong mang
{
	int tam = 0;
	for(int i = 0;i < n ; i++)
		if(a[i] == max)
			tam++;
	return tam;
}

void xoaGiaTriLonNhat(int a[], int &n)
{
	int max = lonNhat(a, n);
	printf("\n\tGia tri lon nhat: %d",max);
	int dem = demMax(a, n, max);
	printf("\n\tSo luong: %d\n",dem);
	while(dem != 0)
	{
		xoaTaiViTri(a,n,viTri(a, n, max));
		dem--;
	}
}

void themTaiViTri(int a[], int &n, int index, int x) // them gia tri x tai vi tri
{
	n++;
	for(int i = n-1;i >=index ; i--)
	{
		a[i] = a[i-1];
	}
	a[index]=x;
}

void themSauPTLonNhat(int a[], int &n, int x){
	themTaiViTri(a,n,viTri(a,n,lonNhat(a,n))+1,x);
}

void themSauPTLaSNT(int a[], int &n, int x){
	int indexSNT=-1;
	for(int i=0; i<n; i++)
		if(kiemTraSNT(a[i])){
			indexSNT=i;
		}
	if(indexSNT!=-1)
		themTaiViTri(a,n,indexSNT+1,x);
	else
		printf("\n\n\tKhong tim thay SNT trong mang.\n");
}
```

MANG 2 CHIEU

```
void taoMang_2c(int a[][MAX],int b[][MAX],
                          int &rowA, int &colA_rowB, int &colB){
	x:
	while(true){
		cout<<"Nhap so hang cua mtr A: ";
		cin>>rowA;
		cout<<"Nhap so cot cua mtr A (= so hang cua mtr B): ";
		cin>>colA_rowB;
		cout<<"Nhap so cot cua mtr B: ";
		cin>>colB;
		if(rowA<0 || rowA>MAX || colA_rowB<0 || colA_rowB>MAX || colB<0 || colB>MAX)
			goto x;
		else break;
	}
	srand(time(NULL));
	for(int i=0; i<rowA;i++)
		for(int j=0; j<colA_rowB; j++)
			a[i][j]=rand()%10;

	for(int i=0; i<colB;i++)
		for(int j=0; j<colA_rowB; j++)
			b[i][j]=rand()%10;
}

void xuatMang_2c(int array[][MAX], int row, int col){
	for(int i=0; i<row;i++){
		for(int j=0; j<col; j++)
			printf("%5d",array[i][j]);
		cout<<endl;
	}
}
// tinh tong C = A + B
void tinhTongMangC(int a[][MAX],int b[][MAX],int c[][MAX],
                                int colA_rowB,int rowC, int colC){
	int tam;
	for(int i=0; i<rowC;i++)
		for(int j=0; j<colC; j++)
		{
			tam=0;
			for(int k=0; k<colA_rowB; k++){
				tam+=a[i][k]*b[k][j];
			}
			c[i][j]=tam;
		}
}
```

STRUCT

```
#include <iostream.h>
#include <string.h>
using namespace std;

typedef struct date{
	unsigned int ngay, thang, nam;
};

typedef struct SV{
	char ten[20];
	date ngaysinh;
};

SV nhapSV(){
	SV sv;
	fflush(stdin);
	cout<<"\nTen: "; gets(sv.ten);
	cout<<"Ngay sinh: "; cin>>sv.ngaysinh.ngay;
	cout<<"Thang: "; cin>>sv.ngaysinh.thang;
	cout<<"Nam: "; cin>>sv.ngaysinh.nam;
	return sv;
}

void xuatSV(SV ds[], int n){
	printf("\n-----------------------------------
-\n");
	printf("\t%-20s	%-10s", "Ho ten", "Ngay
sinh");
	for(int i=0; i<n; i++){
		printf("\n");
		printf("\t%-20s	%d/%d/%d",
ds[i].ten, ds[i].ngaysinh.ngay,
ds[i].ngaysinh.thang, ds[i].ngaysinh.nam);
	}
	printf("\n");
}

int soSanh(SV sv1, SV sv2){
	if(sv1.ngaysinh.nam > sv2.ngaysinh.nam)
		return 1;
	if(sv1.ngaysinh.nam < sv2.ngaysinh.nam)
		return -1;
	return 0;
}

void sapXep(SV ds[], int n){
	for(int i=0; i<n-1; i++)
		for(int j=i+1; j<n; j++){
			if(soSanh(ds[i], ds[j])== -1){
				SV temp = ds[i];
				ds[i] = ds[j];
				ds[j] = temp;

			}
		}

}

main(){
	int n;
	cout<<"Nhap so luong Sinh vien can nhap: ";
	cin>>n;
	SV dssv[n];
	for(int i=0; i<n;i++)
		dssv[i]=nhapSV();
	sapXep(dssv,n);
	xuatSV(dssv, n);
}
```

PHAN SO

```
#include <iostream.h>

struct PS{
   int tu, mau;
};

PS nhap(){
   PS ps;
   cout<<"\tTu so:  "; cin>>ps.tu;
   x:
   cout<<"\tMau so: "; cin>>ps.mau;
   if(ps.mau == 0)
          goto x;
   return ps;
}

void xuat(PS ps){
   if(ps.mau == 1)
          cout<<ps.tu;
   else if(ps.tu == 0)
          cout<<"0";
   else{
          if(ps.mau < 0){
                 ps.tu = -ps.tu;
                 ps.mau = - ps.mau;
          }
          cout<<ps.tu<<"/"<<ps.mau;
   }
}

int UCLN(int a, int b){
   if(a<0) a=-a;
   if(b<0) b=-b;
   if(a==0 || b==0)
          return 1;
   while (a != b){
          if(a>b)
                 a=a-b;
          if(b>a)
                 b=b-a;
   }
   return a;
}

PS rutGon(PS ps){
   PS pskq;
   int ucln = UCLN(ps.tu, ps.mau);
   pskq.tu = ps.tu/ucln;
   pskq.mau = ps.mau/ucln;
   return pskq;
}

PS tong(PS ps1, PS ps2){
   PS ps;
   ps.tu = ps1.tu*ps2.mau + ps2.tu*ps1.mau;
   ps.mau = ps1.mau*ps2.mau;
   ps = rutGon(ps);
   return ps;
}

PS hieu(PS ps1, PS ps2){
      PS ps;
      ps.tu = (ps1.tu*ps2.mau) - (ps2.tu*ps1.mau);
      ps.mau = ps1.mau*ps2.mau;
      ps = rutGon(ps);
      return ps;
   }

   PS tich(PS ps1, PS ps2){
      PS ps;
      ps.tu = ps1.tu*ps2.tu;
      ps.mau = ps1.mau*ps2.mau;
      ps = rutGon(ps);
      return ps;
   }

   PS thuong(PS ps1, PS ps2){
      PS ps;
      ps.tu = ps1.tu*ps2.mau;
      ps.mau = ps1.mau*ps2.tu;
      ps = rutGon(ps);
      return ps;
   }

   main(){
      PS ps1, ps2, psx;

      cout<<"\tNhap phan so thu nhat:"<<endl;
      ps1=nhap();
      cout<<"\n\tNhap phan so thu hai:"<<endl;
      ps2=nhap();

      cout<<"\n\n\tTong: \n";
      psx=tong(ps1, ps2);
      cout<<"\t"; xuat(ps1); cout<<" + ";
xuat(ps2); cout<<" = "; xuat(psx);

      cout<<"\n\n\tHieu: \n";
      psx=hieu(ps1, ps2);
      cout<<"\t"; xuat(ps1); cout<<" - ";
xuat(ps2); cout<<" = "; xuat(psx);

      cout<<"\n\n\tTich: \n";
      psx=tich(ps1, ps2);
      cout<<"\t"; xuat(ps1); cout<<" * ";
xuat(ps2); cout<<" = "; xuat(psx);

      cout<<"\n\n\tThuong: \n";
      psx=thuong(ps1, ps2);
      cout<<"\t("; xuat(ps1); cout<<") / (";
xuat(ps2); cout<<") = "; xuat(psx);
      cout<<"\n\n\t";
   }
```