* ***So nguyen to***

int prime(int n)

{

if (n<2) return 0;

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

{

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

}

return 1;

}

* ***So Hoan Hao***

int SoHoanHao(int n)

{

int s=0;

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

{

if (n%i==0) s=s+i;

}

if (s==n) return 1;

else return 0;

}

* ***So Chinh Phuong***

int SoChinhPhuong(int n)

{

int x=sqrt(n);

if (x\*x==n) return 1;

else return 0;

}

* ***So Amstrong***

int Amstrong(int n)

{

int m=0,remainder=0,result=0,power;

int original=n;

while (original !=0)

{

original=original/10;

m++;

}

original=n;

while (original !=0)

{

remainder=original%10;

power=pow(remainder,m);

result+=power;

original=original/10;

}

if (result ==n) return 1;

else return 0;

}

* ***So Doi Xung***

int daonguoc(int n)

{

int k=n;

int m=0;

while(n>0)

{

m=m\*10+n%10;

n=n/10;

}

if (k==m) return 1;

else return 0;

}

* ***Ham Kiem Tra Mang Doi Xung***

int isSymatric(int a[],int size)

{

for (int i=0;i<size/2;i++)

{

if (a[i]!=a[size-1-i])

return 0;

}

return 1;

}