1. **Hàm**
2. **KT Số NT**

Bool soNT(int n)

{

Int I = 2;

Bool laNT = true;

If(n<2)

laNT = false;

else

{

While(i<= sqrt(n\*1.0) && laNT == true)

{

If(n%i == 0)

{

laNT = false;

}

Else

I++;

}

}

If(laNT == true)

Return true;

Else

Return false;

}

**2.Đổi hệ đếm**

**a.Dec to bin**

int dectoBin(int Dec)

{

int i=0;

int Bin =0;

int temp = Dec;

do

{

Bin = Bin + ((temp%2) \* pow(10.0,i));

temp = temp /2;

i++;

}while(temp >0);

return Bin;

}

**b.Bin to dec**

int bintoDec(int Bin)

{

int i = 0;

int Dec = 0;

int temp = Bin;

do

{

Dec = Dec + ((temp % 10) \* pow(2.0, i));

temp = temp / 10;

i++;

} while (temp > 0);

return Dec;

}

**3.Tính tổng dãy số ( 1->n)**

int tong(int n)

{

int sum =0;

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

{

sum += i;

}

return sum;

}

**4.Tính giai thừa (n!)**

int giaithua(int n)

{

int tich = 1;

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

{

tich = tich \* i;

}

return tich;

}

**2. Mảng**

**1. Tính số fibonaci thứ n**

#include <iostream>

using namespace std;

int fibonaci(int n)

{

int u=0, v=1,t=0,i=3;

while(i <=n)

{

t=u+v;

u=v; v=t; i++;

}

return t;

}

void main()

{

int n;

cout << "Nhap n: ";

cin >> n;

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

{

cout << " " << fibonaci(i);

}

system("pause");

}

**2.Xóa giá trị trùng**

void xoatrung(int a[], int &n)

{

for(int i=0; i<n-1; i++)

{

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

{

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

{

for(int k=j; k<n-1; k++)

{

a[k] = a[k+1];

}

n--; j--;

}

}

}

}

**3.Đếm số giá trị trong mảng**

int demx(int a[], int n , int x)

{

int dem =0;

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

{

if(a[i] == x)

dem ++;

}

return dem;

}

**4.Tổng/ tích các giá trị trong mảng ( chẵn lẻ )**

int tong(int a[],int n)

{

int sum = 0;

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

{

sum += a[i];

}

return sum;

}

int tich(int a[],int n)

{

int tich = 1;

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

{

tich \*= a[i];

}

return tich;

}

**5.Hàm trả về vị trí của pt lớn/bé nhất trong mảng**

int Vitribenhat(int a[], int n)

{

int vitri;

int min = 0;

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

{

if (a[min] > a[i])

min = i;

}

return min;

}

int Vitrilonnhat(int a[], int n)

{

int vitri;

int max = 0;

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

{

if (a[max] < a[i])

max = i;

}

return max;

}