**BÀI TẬP VỀ NHÀ**

# Câu 1:

**int a[10]={2, 55, 4, 23, 12, 7, 6};**

**int i=0, n=10; long long s=0;**

**while(i<n && a[i]%2)**

**{ s +=a[i]; i++; }**

**cout<<“\ns= ”<<s;**

## S=85

## Vòng lập lập lại 6 lần

## Viết lại vòng lập với for:

int a[10]={2, 55, 4, 23, 12, 7, 6};

int n=10; long long s=0;

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

{

If(a[i]%2)

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

cout<<“\ns= ”<<s;

### Viết lại với do…while:

int a[10]={2, 55, 4, 23, 12, 7, 6};

int n=10; long long s=0;

do{

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

cout<<“\ns= ”<<s;}

while(i<n && a[i]%2);

# câu 2:

int a[10]={2, 55, 4, 23, 12, 7, 6};

int i=0, n=10; long long s=0;

while(i<n)

{ if(a[i]%2)

s +=a[i];

i++;

if(a[i]==0) break; }

cout<<“\ns= ”<<s;

## s=85

## vòng lập lập lại 6 lần

## viết lại với for:

int a[10]={2, 55, 4, 23, 12, 7, 6};

n=10; long long s=0;

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

{ if(a[i]%2)

s +=a[i];

i++;

if(a[i]==0) break; }

cout<<“\ns= ”<<s;

### viết lại với do…while:

int a[10]={2, 55, 4, 23, 12, 7, 6};

int i=0,n=10; long long s=0;

do{

s +=a[i];

i++;

} while((a[i]%2)&&(a[i]==0))

cout<<“\ns= ”<<s;

**trong đoạn mã trên có thể bỏ được if(a[i]==0).**

# câu 3:xác định kuôn mẫu hàm cho các bài toán:

## kiểm tra xem k có phải là số hoàn thiện hay không?

void kthoanthien(int k)

{

int s = 0;

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

{

if (n%i == 0)

s += i;

}

if (s == n)

return 1;

else

return 0;

}

## Đếm số hoàn thiện trong mảng (a, n)

Int DemHThien(int a[], int n)

{

int dem = 0;

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

if (hoanthien(a[i]) == 1)

dem++;

return dem;

}

## Tính tổng các chữ số của n

Int Tongn(int n)

{

if (n <= 0)

return 0;

return tongchuso(n / 10) + (n % 10);

}

## In tất cả các chữ số lẻ của n

Void InSoLe(int a[],int n)

{

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

if(a[i]%2!=0)

cout<<a[i];

}

## Tính tổng từ 1 đến n

Int TinhTong(int n, int & tong)

{

Long long s=0;

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

s+=i;

return s;

}

## Hoán vị 2 số

Void HoanVi(int &a, int &b)

{

Int tam;

a=tam;

tam=b;

a=b;

}

## Tìm ước số chung lớn nhất của 2 số

int TimUocSo(int a, int b)

{

while(a!=b)

if(a>b)

a=a-b;

else

b=b-a;

return a;

}

## Tìm min trên mảng 1 chiều (a, n)

Int TimMin(int a[MAX], int n)

{

Int min = a[0];

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

{

if (a[i] < a[0])

min=a[i];

}

return min;

}

## Sắp xếp tăng mảng 1 chiều (a, n)

Void SapTang(int a[MAX], int n)

{

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

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

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

{

int tam = a[i];

a[i] = a[j];

a[j] = tam;

}

}

## Giải phương trình bậc 1

Void GPT1(int a,int b, int &x)

{

If(a==0&&b==0)

Cout<<”Phuong trinh vo so nghiem”;

Else

Cout<<”phuong trinh vo nghiem”

If(a!=0)

Cout<<”nghiem la :”<<x=-b/a;

}

## Giải phương trình bậc 2

Void GPT2(int a, int b, int c, int &x1, int &x2)

{

Float dentan=b\*b-4\*a\*c;

If(a==0)

{

If(b==0)

Cout<<”pt vo so nghiem”;

Else

Cout<<”pt vo nghiem”;

}

else(a!=0)

{

If(denta>0)

X1=(-b + [sqrt](http://cppreference.com/stdmath/sqrt.html" \t "_blank)(denta))/(2\*a);

X2=(-b - [sqrt](http://cppreference.com/stdmath/sqrt.html" \t "_blank)(denta))/(2\*a);

If(denta==0)

X1=x2=-b/(2\*a);

If(denta<0)

Cout<<”phuong trinh vo nghiem”;

}

}

## Đọc dữ liệu từ tập tin vào mảng (a, n)

Void DocDL(int a[MAX],int n)

## Ghi dữ liệu từ mảng (a, n) ra tập tin

Void DocDL(int a[MAX],int n)

# Đoạn code:

## Đoạn 1:

int kq=0; int l=1, r=20;

while(l<r)

{ if((l\*r)%2==0) kq += l+r; l++; r--; }

printf("\nkq= %d",kq);

**kết quả bằng: 21.**

## Đoạn 2:

int n=2, d=0;

do

{ if(n%2)

{ printf("%d\t",n\*2); d++; }

n++; if(d==5) break; }

while(1);

**kết quả:**

**n=4; n=6;n=8;n=10;n=12;n=14**

## đoạn 3:

void Solve(int &k, int &h)

{ int tam=k+h; k=tam; h=tam-h; }

Cho biết kết quả của đoạn code: int k=12, h=8;

Solve(k,h);

Solve(h,k);

Solve(k,h);

printf("\nk= %d, h=%d",k,h);

**kết quả:**

k=20

h=8

## đoạn 4:

int a[9]={0},i=0,j=0;

printf("\n%d: ",i);

while(i<3)

{ printf("%d\t",j);

if(j<3){ a[i\*3+j] += i\*j+i+j; j++; }

else{ i++; j=i;

printf("\n%d: ",i); } }

**kết quả:**

**i=0**

## đoạn 5:

void main()

{ int n=1;

for(; n<5; n++)

{ int n=1;

cout<<n<<“\n”; n++; }

}

**Kết quả:**

n=1;