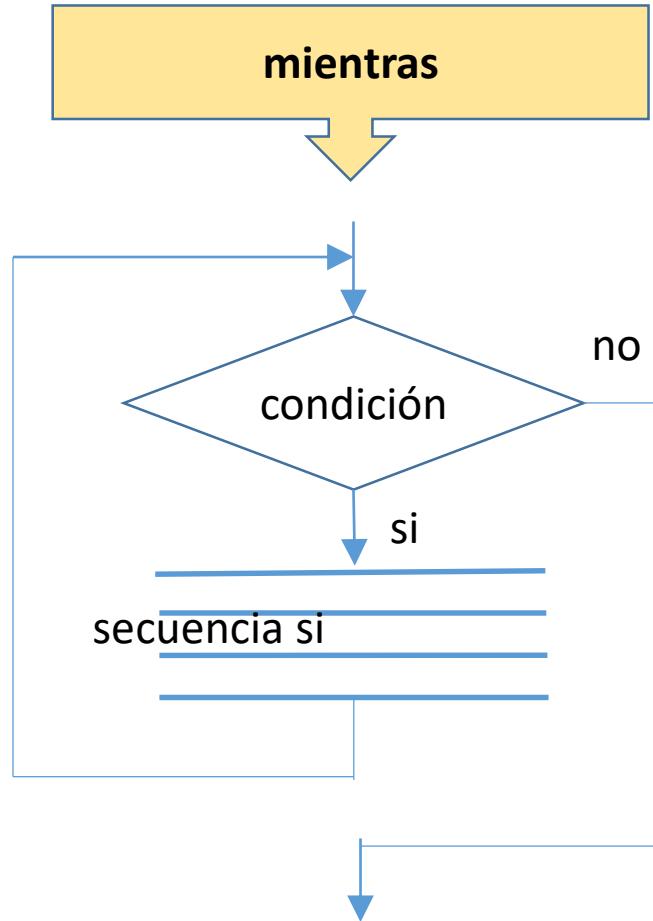




Facultad de Ingeniería Ingeniería y de Sistemas

Algoritmia y Estructura de Datos: Estructuras de control - Iteración

Estructuras de control: Iteración



```
while (condicion) {  
    secuencia si;  
}
```

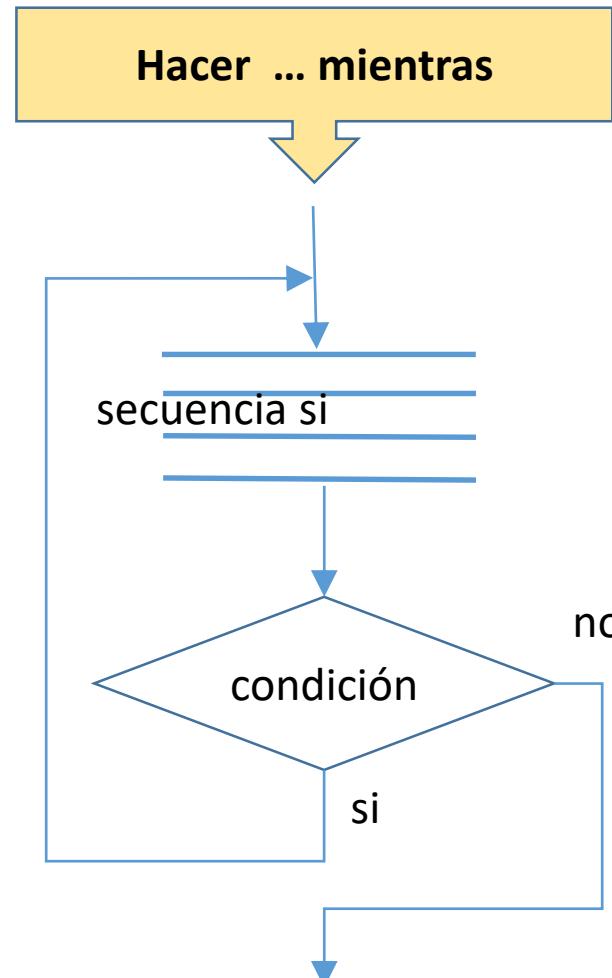
```
while (a < n) {  
    a = a + 1;  
    cout<<a<<endl;  
}
```

Importante

La secuencia puede no ejecutarse. Todo depende del valor de la condición

Si se ejecuta la secuencia si, esta debe en alguna ocasión cambiar el resultado de la condición

Estructuras de control: Iteración



```
do {  
    secuencia si;  
} while (condicion);
```

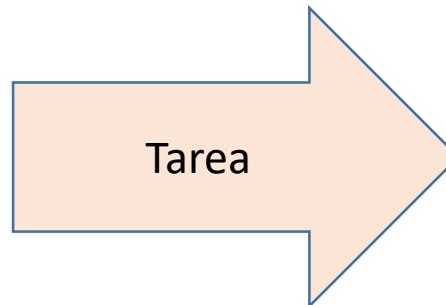
```
do {  
    a = a + 1;  
    cout<<a<<endl;  
} while (a < n);
```

Importante

La secuencia si, se ejecuta como mínimo una vez.

La ejecución de la secuencia si, en alguna ocasión debe cambiar el resultado de la condición

Estructuras de control: Iteración



Desarrollar un algoritmo que reciba n números enteros y determine que cuál es el máximo valor par y cual es mínimo valor impar. Considere que cero es par.



Suponer un valor inicial máximo para los pares y otro valor inicial mínimo para los impares. Luego leer un número, determinar si es par o impar. Si es par compararlo con el maximo par y si es impar con el mínimo impar. Reemplazar el maximo par con el número si resulta mayor. Para el caso de los impares reemplazar el minimo impar si resulta menor.

Estructuras de control: Iteración

Tarea

Desarrollar un algoritmo que reciba n números enteros y determine cuál es el máximo valor par y cual es mínimo valor impar. Considere que cero es par.

Solución

```
1 #include <iostream>
2 using namespace std;
3 main(){
4     int n,num,i=0,contpar=0,contimp=0, maxpar, minimpar;
5     cout<<"Cuantos numeros ingresara? "; cin>>n;
6     do { cout<<"Ingrese un numero entero: "; cin>>num;
7         if(num%2!=0) {                                     // cuando es impar
8             contimp++;
9             if (contimp==1) minimpar = num;    // supone que el primer impar es el minimo
10            else if (num<minimpar)           // se compara con el minimo impar
11                minimpar = num;
12        }
13        else {                                         // cuando es par
14            contpar++;
15            if (contpar==1) maxpar = num;    // supone que el primer par es el maximo
16            if(num>maxpar)               // se compara con el maximo par
17                maxpar = num;
18        }
19        i++;
20    } while(i<n);
21    if(contpar>0) cout<<"El maximo par es "<<maxpar<<endl;
22    else
23        cout<<"No se ingresaron numeros pares"<<endl;
24    if(contimp>0) cout<<"El minimo impar es "<<minimpar<<endl;
25    else
26        cout<<"No se ingresaron numeros impares"<<endl;
27 }
```

Algoritmia: máximo y mínimo de un conjunto de valores

Importante

Se requiere suponer un valor inicial para el máximo o para el mínimo, que tenga alguna posibilidad de ser refutado por la data.

Si **se conocen** los límites del conjunto de datos es posible asignar un valor inicial conveniente.

Ejemplo: mínimo de las notas de un examen de algoritmia.
El valor inicial que conviene será 20

Se **no se conocen** los límites del conjunto de datos, entonces el valor inicial mínimo o maximo debe ser el primer valor de la data.

Estructuras de control: iteración

Tarea

Desarrollar un algoritmo que lea un número entero positivo y determine cuantas cifras tiene, cuántas son pares y cuántas son impares. Asuma que cero es par.
Resolverlo utilizando while y/o do ... while

Ejemplo: 4387

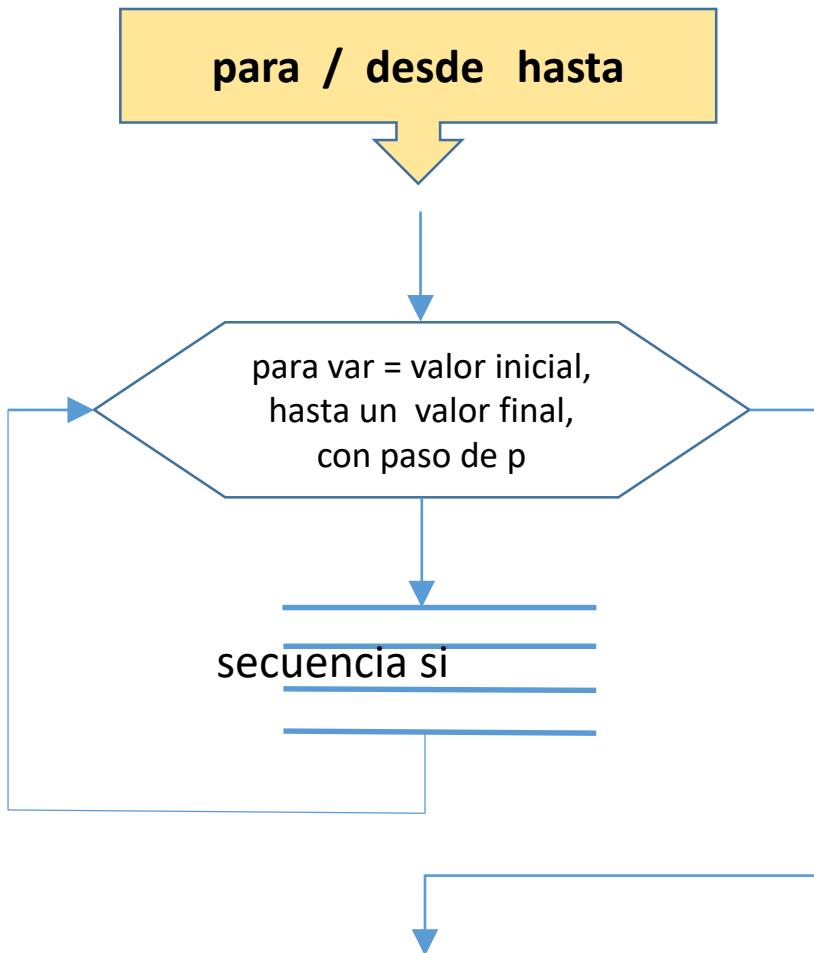
4 cifras

2 cifras pares

2 cifras impares

```
1 #include <iostream>
2 using namespace std;
3 main() {
4     int n,contcifras=0,aux,contpar=0, contimpar=0;
5     do { cout<<"ingrese un numero entero positivo" << endl; cin>>n;
6         if(n<=0)
7             cout<<"numero no es positivo, ingrese de nuevo" << endl;
8     } while(n<=0);
9     aux=n;
10    while(aux>0){
11        if((aux%10)%2==0)
12            contpar++;
13        else
14            contimpar++;
15        aux=aux/10;
16    }
17    cout<<"el numero " << n << " tiene " << contpar+contimpar << " cifras" << endl;
18    if(contpar>0)
19        cout<<"tiene " << contpar << " cifras pares" << endl;
20    else
21        cout<<"no tiene cifras pares" << endl;
22    if(contimpar>0)
23        cout<<"tiene " << contimpar << " cifras impares" << endl;
24    else
25        cout<<"no tiene cifras impares" << endl;
26 }
```

Estructuras de control: iteración



```
for ( int var = valor inicial; var < valor final; paso) {  
    secuencia si;  
}
```

```
for ( int i = 1; i < 100; i = i + 2) {  
    suma = suma + i;  
    cout<<suma<<endl;  
}
```

Algoritmia: contadores

Importante

Todo variable que actuará de **contador** requiere un valor inicial igual a cero.

Ejemplo: contador de impares en un conjunto de n numeros enteros

```
conimp = 0;  
for (int i = 0; i < n; i = i + 1) {  
    cout<<"Ingrese un numero entero ";  
    cin>>num;  
    if (num % 2 != 0)  
        conimp = conimp + 1;  
}
```

Algoritmia: sumadores

Importante

Todo variable que actuará de **sumador** requiere un valor inicial igual a cero.

Ejemplo: sumador de pares en un conjunto de n números enteros

```
sumpar = 0;  
for (int i = 0; i < n; i = i + 1) {  
    cout<<"Ingrese un numero entero ";  
    cin>>num;  
    if (num % 2 == 0)  
        sumpar = sumpar + num;  
}
```

Algoritmia: descomposición y composición de números

Tarea

Escribir un algoritmo que lea un número entero positivo y construya un número con las cifras en orden invertido.

Ejemplo: si el número es 14327 debe dar como resultado 72341

Estrategia

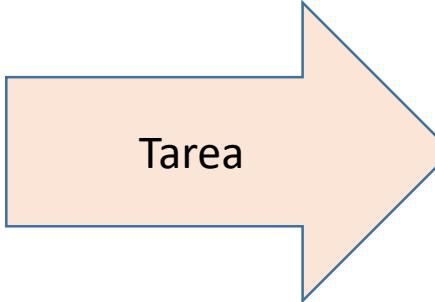
Descomponer el número en dígitos utilizando el residuo de la división por 10 y con cada dígito obtenido, construir sucesivamente el nuevo número multiplicándolo por 10 añadiéndole el dígito en curso.

Dígitos para construir un número: 4, 5, 0 y 2

Sea nn nuevo número entonces:
 $nn = 0;$

```
nn = nn*10+4; // nn será 4
nn = nn*10+5; // nn sera 45
nn = nn*10+0; // nn será 450
nn = nn*10+2; // nn sera 4502
```

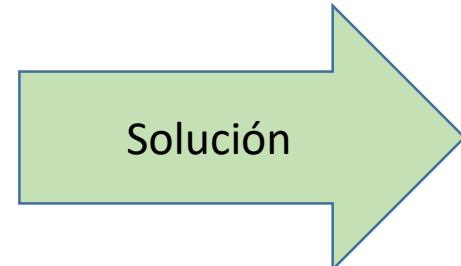
Algoritmia: descomposición y composición de números



Tarea

Escribir un algoritmo que lea un número entero positivo y construya un número con las cifras en orden invertido.

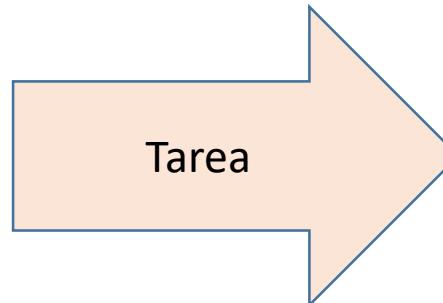
Ejemplo: si el número es 14327 debe dar como resultado 72341



Solución

```
1 #include<iostream>
2 using namespace std;
3 main(){
4     int num,numinv=0,digito;
5     do {
6         cout<<"Digite un numero entero positivo: ";
7         cin>>num;
8     } while(num<=0);
9
10    while(num>0){
11        digito = num % 10;
12        numinv = numinv * 10 + digito;
13        num = num/10;
14    }
15    cout<<"El numero invertido es "<<numinv<<endl;
16 }
```

Algoritmia: descomposición y composición de números



Escribir un algoritmo que lea un número entero positivo y determine cuantas cifras diferentes tiene.

Ejemplo: si el número es 190391 entonces tiene 4 cifras diferentes



Hacer un bucle del dígito 0 al dígito 9 y en cada vuelta determinar si el número contiene el dígito en curso. Determinar si el número contiene cierto dígito, se hace descomponiendo el número en dígitos, y contando las veces que ocurre el dígito que se busca..

Algoritmia: descomposición y composición de números

Tarea

Escribir un algoritmo que lea un número entero positivo y determine cuantas cifras diferentes tiene.

Ejemplo: si el número es 190391 entonces tiene 4 cifras diferentes

Solución

```
1 #include <iostream>
2 using namespace std;
3 main(){
4     int num,digito,concifdif,coni=0,aux;
5     concifdif=0;
6     do {
7         cout<<"ingrese un numero entero positivo:"<<endl;
8         cin>>num;
9     } while (num<=0);
10    aux=num;
11    for (int i=0;i<10;i=i+1){
12        while(aux>0){
13            digito = aux % 10;
14            if (digito == i)
15                coni++;
16            aux = aux/10;
17        }
18        if (coni>0)
19            concifdif++;
20        coni = 0;
21        aux = num;
22    }
23    cout<<"tiene "<<concifdif<<" cifras diferentes"<<endl;
24 }
```

Algoritmia: almacenamiento en números enteros

Tarea

Escribir un algoritmo que lea las edades de n niños, todos menores de 10 años y determine la edad promedio, y cuantos tiene una edad igual mayor que el promedio.

Estrategia

Leer las edades e ir construyendo un número entero con ellas, luego calcular y finalmente descomponer el número en dígitos para recuperar las edades y determinar cuantos tiene una edad mayor o igual al promedio.

Edades:
6, 4, 8, 5, 7, 3, 5, 6

Número
construido:
64857356

Promedio:
5.5

Al descomponer el número en dígitos, 5 niños tienen edades mayores o iguales al promedio

Algoritmia: almacenamiento en números enteros

Tarea

Escribir un algoritmo que lea las edades de n niños, todos menores de 10 años y determine la edad promedio, y cuántos tienen una edad igual mayor que el promedio.

Solución

```
1 #include <iostream>
2 using namespace std;
3 main(){
4     int n,edad,i,numero=0,cont=0;
5     float suma=0;
6     do {
7         cout<<"Digite la cantidad de edades a ingresar: ";
8         cin>>n;
9     } while (n<=0);
10    for(int i=0;i<n;i++){
11        do {
12            cout<<"Ingrese edad menor a 10 años ";
13            cin>> edad;
14        } while(edad<=0 || n>9);
15        numero=numero*10+edad;
16        suma=suma+edad;
17    }
18    cout<<"El promedio es "<< suma/n<<endl;
19    for(int i=0;i<n;i++){
20        edad=numero%10;
21        if(edad>= suma/n)
22            cont++;
23        numero=numero/10;
24    }
25    cout<<"hay "<< cont<<" edades mayores o iguales al promedio "<<endl;
```