

Auxiliatura INF-143 “A”

Algoritmos Básicos Recursivos

Univ. Miguel Angel Quispe Mamani

Universidad Mayor de San Andrés

Carrera de Informática

I/2022

1 Recursividad

La recursividad es una técnica de programación en la que un módulo hace una llamada a si mismo, con el fin de resolver el problema. La llamada a si mismo se conoce como llamada recursiva.

Dicho formalmente, un algoritmo se dice recursivo si calcula instancias de un problema en función de otras instancias del mismo problema hasta llegar a un caso base, que suele ser una instancia pequeña del problema, cuya respuesta generalmente está dada en el algoritmo y no es necesario calcularla.

1.1 Primeros números

Dado un número n imprimir la secuencia de 1 a n , mediante un algoritmo recursivo.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 //Primera funcion (1, 2, 3, 4,..., n):
5 void f(int i){
6     if(i == 0) return; //caso base
7     f(i - 1); //llamada recursiva
8     cout << i << "\n"; //mostramos i
9 }
10
11 //Segunda funcion (n, n - 1, n - 2,..., 2, 1):
12 void g(int i){
13     if(i == 0) return; //caso base
14     cout << i << "\n"; //mostramos i
15     g(i - 1); //llamada recursiva
16 }
17
18
```

```
19 int main(){
20     int n;
21     cin >> n;
22     f(n);
23     g(n);
24     return 0;
25 }
```

1.2 Valor Absoluto

El valor absoluto de un número entero es el número natural que resulta al suprimir su signo. Dado un número n imprimir su valor absoluto, mediante un algoritmo recursivo.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int f(int i){
5     if(i == 0) return 0; //caso base
6     if(i > 0) return f(i - 1) + 1; //si n es positivo
7     else return f(i + 1) + 1; //si n es negativo
8 }
9
10 int main(){
11     int n;
12     cin >> n;
13     cout << f(n) << "\n";
14     return 0;
15 }
```

1.3 Sumatoria

Dado un número n imprimir su sumatoria, mediante un algoritmo recursivo.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int g(int n){
5     if(n == 0) return 0; //caso base
6     return g(n - 1) + n;
7 }
8
9 int main(){
10     int n;
11     cin >> n;
12     cout << g(n) << "\n";
13     return 0;
14 }
```

1.4 Factorial

Dado un número n imprimir su factorial, mediante un algoritmo recursivo.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int f(int n){
5     if(n == 0) return 1; //caso base, ya que el factorial de 0 es 1
6     return f(n - 1) * n;
7 }
8
9 int main(){
10     int n;
11     cin >> n;
12     cout << f(n) << "\n";
13     return 0;
14 }
```

1.5 N-ésimo Fibonacci

Dado un número n imprimir el n -ésimo Fibonacci(1, 1, 2, 3, 5, 8, ...), mediante un algoritmo recursivo.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int f(int n){
4     if(n == 1 || n == 2) return 1;
5     return f(n - 1) + f(n - 2);
6 }
7 int main(){
8     int n;
9     cin >> n;
10    cout << f(n) << "\n";
11    return 0;
12 }
```

1.6 Dígitos de un número

Dado un número n ($n > 0$), hallar su suma de sus dígitos.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int f(int n){
4     if(n == 0) return 0;
5     return f(n / 10) + n % 10;
6 }
7 int main(){
8     int n;
9     cin >> n;
10    cout << f(n) << "\n";
11    return 0;
12 }
```

1.7 Invertir un número

Dado un número n imprimir el número invertido, mediante un algoritmo recursivo.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int f(int n, int m){//el primer parametro corresponde al numero original,
4     el segundo al nuevo numero invertido
5     if(n == 0) return m;
6     return f(n / 10, (m * 10) + (n % 10));
7 }
8 int main(){
9     int n;
10    cin >> n;
11    cout << f(n, 0) << "\n";
12    return 0;
13 }
```

1.8 Invertir una cadena

Dado un cadena s imprimir la cadena inversa, mediante un algoritmo recursivo.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 string f(int i, string cad, string aux){//posicion, cadena original,
4     cadena inversa
5     if(i == cad.size()) return aux;
6     return f(i + 1, cad, cad[i] + aux);
7 }
8 int main(){
9     string s;
10    cin >> s;
11    cout << f(0, s, "") << "\n";
12    return 0;
13 }
```