Recursivitate

Definiție: Recursivitatea reprezintă proprietatea unor noțiuni de a se defini prin ele însele.

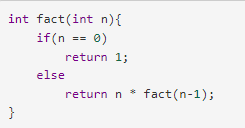
Exemple:

* factorialul unui număr: N!=N⋅(N−1)!;

Formula recursivă pentru n! se aplică numai pentru n>0 iar prin definiție 0!=1 .

Exemplu:

O variantă recursivă pentru determinarea lui n! este:



* șirul lui Fibonacci: Fn=Fn−1+Fn−2 ;

Cum funcționeaza recursivitatea?

Toate variabilele locale din definiția unei funcții precum și valorile parametrilor formali se memorează la apel în memoria de tip STIVĂ (STACK).

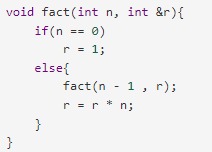
În cadrul unui program, pentru fiecare apel al unei funcții, sistemul adaugă o zonă de memorie pe stivă. Această zonă stochează variabilele locale și parametrii specifici pentru apelul respectiv. Această zonă de memorie rămâne pe stivă până când apelul se încheie, moment în care este eliberată. Dacă în cadrul apelului curent se face un alt apel către o altă funcție, sistemul adaugă o nouă zonă de memorie pe stivă, iar conținutul zonei anterioare devine inaccesibil până la finalizarea noului apel. Același proces are loc și în cazul în care al doilea apel este un autoapel al aceleiași funcții recursive.

Observație:

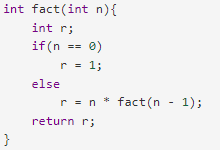
* este obligatoriu ca în definiția unei funcții recursive să apară cazul particular (în care să nu aibă loc autoapelul). În caz contrar autoapelurile vor avea loc ,,la nesfârșit”. În urma prea multor autoapeluri, stiva se va ocupa în totalitate și execuția programului se va întrerupe.

Cum facem autoapelul?

* dacă funcția recursivă este de tip non-void, autoapelul se va face într-o expresie;



* dacă funcția recursivă este de tip void, autoapelul se va face într-o instrucțiune de sine stătătoare; dacă funcția întoarce valori, se vor folosi parametri de ieșire.



Aplicații:

1) Subprogramul f este definit alăturat. Indicați valoarea lui f(2023).

int f(int n){

if(n==0) return 0;

if(n%10==2) return f(n/10)\*10+3;

return f(n/10)\*10+2;

}

Rezolvare:

f(2023) = f(202) \* 10 + 2 = 3232

f(202) = f(20) \* 10 + 3 = 323

f(20) = f(2) \* 10 + 2 = 32

f(2) = f(0) \* 10 + 3 = 3

f(0) = 0

2) Subprogramul f este definit alăturat. Scrieți toate valorile naturale din intervalul [1,10] pe care le poate avea x, astfel încât valoarea lui f(10,x) să fie un număr strict mai mare decât 20.

int f(int a, int b)

{ if(a>b) return a/b+f(a-b, b);

if(a<b) return b/a+f(a, b-a);

return 1;

}

Rezolvare:

f(10, x)

x = 1

f(10, 1) = 10 + f (9, 1) = 55

f(9, 1) = 9 + f(8, 1) = 45

f(8, 1) = 8 + f(7, 1) = 36

f(7, 1) = 7 + f(6, 1) = 28

f(6, 1) = 6 + f(5, 1) = 21

f(5, 1) = 5 + f(4, 1) = 15

f(4, 1) = 4 + f(3, 1) = 10

f(3, 1) = 3 + f(2, 1) = 6

f(2, 1) = 2 + f(1, 1) = 3

f(1, 1) = 1

55>20

x = 9

f(10, 9) = 1 + f(1, 9) = 46

f(1, 9) = 9 + f(1, 8) = 45

f(1, 8) = 8 + f(1, 7) = 36

f(1, 7) = 7 + f(1, 6) = 28

f(1, 6) = 6 + f(1, 5) = 21

f(1, 5) = 5 + f(1, 4) = 15

f(1, 4) = 4 + f(1, 3) = 10

f(1, 3) = 3 + f(1, 2) = 6

f(1, 2) = 2 + f(1, 1) = 3

f(1, 1) = 1

R: x = 1, 9

3) Subprogramul f este definit alăturat. Indicați valoarea f(102030).

int f (int x)

{ if(x>20) return 20+f(x/10);

return 2020; }

Rezolvare:

f(102030) = 20 + f(10203) = 2100

f(10203) = 20 + f(1020) = 2080

f(1020) = 20 + f(102) = 2060

f(102) = 20 + f(10) = 2040

f(10) = 2020

4) Subprogramul f este definit alăturat. Scrieți două numere naturale din intervalul [1,10], care pot fi memorate în variabilele întregi x1, respectiv x2, astfel încât valoarea lui f(10,x1) să fie 5, iar valoarea lui f(x2,10) să fie 1.

int f(int a, int b)

{if(a>b) return a/b+f(a-b,b);

if(a<b) return b/a+f(a,b-a);

return 1; }

f(10 ,x1) =>

x1 = 6

f(10, 6) = 1 + f(4, 6) = 5

f(4, 6) = 1 + f(4, 2) = 4

f(4, 2) = 2 + f(2, 2) = 3

f(2, 2) = 1

f(x2, 10) => x2 = 10

f(10, 10) = 1

R: x1 = 6

x2 = 10

5) Subprogramul f este definit alăturat. Indicați ce se afișează în urma apelului de mai jos. f(12345);

void f (int x)

{

cout<<”+”;

if(x>0)

{

f(x/100);

cout<<x;

}

cout<<”+”; }

f(12345)

f(123)

f(1)

f(0)

R: + + + + + + 1 + 123 + 12345 +

6) Subprogramele f1 și f2 sunt definite mai jos.

int f1 (int x, int y)

{ if(y==0) return x;

else return f1(x%y,y); }

int f2(int x, int y)

{ if(x==y) return x;

else if(x>y) return f2(x-y,y);

else return f2(x,y-x); }

f1(15,25) = f1(5, 25)

f1(5, 25) = f1(5 ,25) – Fals

f2(15, 25) = f2(15, 10)

f2(15, 10) = f2(5, 10)

f2(5, 10) = f2(5, 5)

f2(5, 5) = 5 – Adevarat

scrie 5

R: numai f2

7) Subprogramul f este definit alăturat. Pentru apelul f(20,2020), funcția se execută de:

int f(int x, int y)

{ if(x<=1 || y<=1) return 0;

if(x>y) return 1+f(f(x/y,y),y);

return 1+f(x,f(x,y/x)); }

f(20, 2020) = 1+ f(20, f(20, 101) = 1+ f(20,1) = 1

f(20, 101) = 1+ f(20, f(20, 5) = 1 + f(20, 1) = 1

f(20, 5) = 1 + f( f(4, 5), 5) = 1+ f(1, 5) = 1

f(4, 5) = 1 + f(4, 1) = 1 + f(4, 0) = 1

f(4, 1) = 0

R: 9 ori

8) Subprogramele f1 și f2 sunt definite mai jos. Indicați valoarea f2(41382).

int f1(int c)

{ if (c%2==1) return 1;

else return 2; }

int f2(int n)

{ if (n==0) return 0;

else return f1(n%10)+f2(n/10); }

f2(41382) = f1(2) + f2(4138) = 2+ 6 = 8, f1(2) = 2

f2(4138) = f1(8) + f2(413) = 2 + 4 = 6, f1(8) = 2

f2(413) = f1(3) + f2(41) = 1 + 3 = 4, f1(3) = 1

f2(41) = f1(1) + f2(4) = 1 + 2 = 3, f1(1) = 1

f2(4) = f1(4) + f2(0) = 2 + 0 = 2, f1(4) = 2

f2(0) = 0

R: 2

9) Pentru funcţiile f şi g definite mai jos, scrieţi care este rezultatul returnat la apelul g(11). Dar rezultatul returnat la apelul f(6)?

long g(long x)

{ if (x>9)

return (x/10 + x%10);

else

return x; }

long f(int c)

{ if (c<1)

return 1;

else

return g(c+f(c-1)); }

a) g(11) = 2

b) f(6) = g(6 + f(5)) = g(11) = 2

f(5) = g(5 + f(4)) = g(14) = 5

f(4) = g(4 + f(3)) = g(9) = 9

f(3) = g(3 + f(2)) = g(5) = 5

f(2) = g(2 + f(1)) = g(3) = 3

f(1) = g(1 + f(0)) = g(1) = 1

f(0) = 0

10) Subprogramul f este definit alăturat. Ce valoare are f(8,4)?

int f (int x,int y)

{

if(x<y)

return 1+f(x+1,y);

if(y<x)

return 1+f(y+1,x);

return 1;

}

f(8, 4) = 1 + f(5, 8) = 1 + 4 =5

f(5, 8) = 1 + f(6, 8) = 1 + 3 = 4

f(6, 8) = 1 + f(7, 8) = 1 +2 = 3

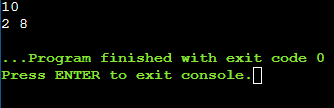
f(7, 8) = 1 + f(8, 8) = 1 + 1 = 2

f(8, 8) = 1

R: 5

11) Se citeşte un număr natural n. Să se scrie n ca sumă de puteri crescătoare ale lui 2.

Exemplu:

Intrare: 10

Ieșire: 2 8

#include <iostream>

using namespace std;

void suma(int a, int p)

{

if (a == 0)

return;

if (a % 2 == 1)

{

cout << p << " ";

suma(a / 2, p\* 2);

}

else {

suma(a / 2, p\* 2);

}

}

int main()

{

int x, y = 1;

cin >> x;

suma(x, y);

return 0;

}