Charlson So

CS 211

Hw3

Pg89-94

1. #1

The parts of a recursive algorithm are the base case and the general recursive case.

The base case is when the program terminates. This is shown in this program where:

If(n<=0)

Return 0;

This will terminate when n = 0;

The recursive case usually calls the function again and attempts to make break the problem into smaller pieces. This is show in the else statement where:

return getNumberEqual(x,n-1,desiredValue)+count;

This statement will check through the array one by one until there is no more elements to check, where the recursive function will terminate due to the base case.

1. #6

The problem has no base case. The function will loop and will not be able to terminate.

1. #11

getValue(1,7,7) -> c=4

getValue(1,4,7)->c=2

output will be 4

1. #18 a&b

#include <iostream>

using namespace std;

int power1(int x, int n){

int total = 1;

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

total\*=x;

}

return total;

}

int power2(int x, int n){

int total =1;

if (n==0){

return 1;

}

else{

return power2(x,n-1)\*x;

}

}

int main(){

cout<<power1(2,5);

cout<<power2(2,5);

}

1. #20a&b
2. f(6):8

f(7):11

f(8)=f(7)+3\*f(3)=11+3=14

f(9)=f(8)+3\*f(4)=14+9=23

f(10)=f(9)+3\*f(5)=23+15=38

f(12):f(11)+3\*f(7)=f(10)+3\*f(6)+3\*f(7)=f(9)+3\*f(5)…=f(8)+3\*f(4)…=f(7)+3\*f(3)+ 3\*f(4)+ 3\*f(5)+3\*f(6)+3\*f(7)=11+3+9+15+24+33=95

f(15):f(14)+3\*f(10)=f(13)+3\*f(9)+3\*f(10)=f(12)+3\*f(8)+3\*f(9)+3\*f(10)=

95+3\*14+3\*23+3\*38=95+42+69+114=320

b. #include <iostream>

using namespace std;

int rec(int n){

if (n==1||n==2||n==3){

return 1;

}

else if (n==4){

return 3;

}

else if (n==5){

return 5;

}

else{

return rec(n-1)+3\*rec(n-5);

}

}

int main(){

cout<<rec(15);

}

Terminal:

320