I, ME AND MYSELF !!!

TUESDAY, DECEMBER 8, 2009

Practice Recursions

Recursions are fun

You may like to try out some simple problems to practice recursions. Try to solve all of them without using any global variables. And try on your own before looking at the solutions. Also please notify any error to me (zobayer1[at]gmail[dot]com). Before looking at the problems, you may like to read this post about how to attack recursive problems.

Problem 1:

You will be given an array of integers, write a recursive solution to print it in reverse order.

```
Input:
5
69 87 45 21 47
Output:
47 21 45 87 69
```

see answer

Problem 2:

Write a recursive function to print an array in the following order. [0] [n-1] [1] [n-2] .... [(n-1)/2] [n/2]

```
Input:
5
1 5 7 8 9
Output:
1 9
5 8
7 7
```

see answer

Problem 3:

Write a recursive program to remove all odd integers from an array. **You must not use any extra array or print anything in the function.** Just read input, call the recursive function, then print the array in main().

```
Input:
6
1 54 88 6 55 7
Output:
54 88 6
```

see answer

Problem 4:

Write a recursive solution to print the polynomial series for any input n: $1 + x + x^2 + \ldots\ldots\ldots\ldots + x^{n-1}$

```
Input:
5
Output:
1 + x + x^2 + x^3 + x^4
```

see answer

Problem 5:

Write a recursive solution to evaluate the previous polynomial for any given x and n. Like, when x=2 and n=5, we have $1 + x + x^2 + \ldots\ldots\ldots\ldots + x^{n-1} = 31$

```
Input:
2 5
Output:
31
```

see answer

Problem 6:

Write a recursive program to compute n!

```
Input:
5
Output:
120
```

see answer

Problem 7:

Write a recursive program to compute n^{th} fibonacci number. 1^{st} and 2^{nd} fibonacci numbers are 1, 1.

```
Input:
6
Output:
8
```

see answer

Problem 8:

Write a recursive program to determine whether a given integer is prime or not.

```
Input:
49
999983
1
Output:
not prime
prime
not prime
```

see answer

Problem 9:

Write a recursive function that finds the gcd of two non-negative integers.

```
Input:
25 8895
Output:
5
```

see answer

Problem 10:

Write a recursive solution to compute lcm of two integers. You must not use the formula lcm(a,b) = (a x b) / gcd(a,b); find lcm from scratch...

```
Input:
23 488
Output:
11224
```

see answer

Problem 11:

Suppose you are given an array of integers in an arbitrary order. Write a recursive solution to find the maximum element from the array.

```
Input:
5
7 4 9 6 2
Output:
9
```

see answer

Problem 12:

Write a recursive solution to find the **second maximum** number from a given set of integers.

```
Input:
5
5 8 7 9 3
Output:
8
```

see answer

Problem 13:

Implement linear search recursively, i.e. given an array of integers, find a specific value from it. Input format: first n, the number of elements. Then n integers. Then, q, number of query, then q integers. Output format: for each of the q integers, print its index (within 0 to n-1) in the array or print 'not found', whichever is appropriate.

```
Input:
5
2 9 4 7 6
2
5 9
Output:
not found
1
```

see answer

Problem 14:

Implement binary search recursively, i.e. given an array of **sorted** integers, find a query integer from it. Input format: first n, the number of elements. Then n integers. Then, q, number of query, then q integers. Output format: for each of the q integers, print its index (within 0 to n-1) in the array or print 'not found', whichever is appropriate.

```
Input:
5
1 2 3 4 5
2
3 -5
Output:
2
not found
```

see answer

Problem 15:

Write a recursive solution to get the reverse of a given integer. **Function must return an int**

```
Input:
123405
Output:
504321
```

see answer

Problem 16:

Read a string from keyboard and print it in reversed order. **You must not use any array to store the characters**. Write a recursive solutions to solve this problem.

```
Input:
helloo
Output:
oolleh
```

see answer

Problem 17:

Write a recursive program that determines whether a given sentence is palindromic or not just considering the alpha-numeric characters ('a'-'z'), ('A'-'Z'), ('0'-'9').

```
Input:
madam, i'm adam
hulala
Output:
palindromic
not palindromic
```

see answer

Problem 18:

Implement strcat(), stracpy(), strcmp() and strlen() recursively.

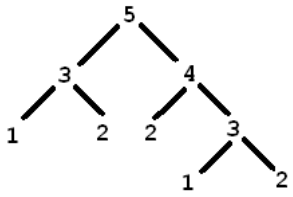
```
Input:
test on your own
```

Output:
test on your own

see answer

Problem 19:

If you already solved the problem for finding the n^{th} fibonacci number, then you must have a clear vision on how the program flow works. So now, in this problem, print the values of your fibonacci function in pre-order, in-order and post-order traversal. For example, when n = 5, your program calls 3 and 4 from it, from the call of 3, your program calls 1 and 2 again..... here is the picture:



Input:
5
Output:
```
Inorder: 1 3 2 5 2 4 1 3 2
Preorder: 5 3 1 2 4 2 3 1 2
Postorder: 1 2 3 2 1 2 3 4 5
```

see answer

Problem 20:

All of you have seen the tower of Hanoi. You have 3 pillars 'a', 'b' and 'c', and you need transfer all disks from one pillar to another. Conditions are, only one disk at a time is movable, and you can never place a larger disk over a smaller one. Write a recursive solution to print the moves that simulates the task, a -> b means move the topmost of tower a to tower b.

Input:
3
Output:
```
a -> c
a -> b
c -> b
a -> c
b -> a
b -> c
a -> c
```

see answer

Good Luck!!!

Posted by Zobayer Hasan at 2:19 AM

30 comments:

Lokesh January 15, 2011 at 1:46 PM

really nice :)

Reply

Unknown January 21, 2011 at 3:53 PM

need some critical recursion?

Reply

Zobayer Hasan January 21, 2011 at 5:23 PM

Haha, I have plenty :) I think these are enough for beginners to give them a head start, after these, they can practice on their own.

But If you think that you can help them further, you can post here the problems, or source links or whatever you have.

Reply

Sanjary Rahman October 24, 2012 at 1:36 AM

Truly Helpful Brother...May Allah Bless you :)

Reply

Anonymous December 28, 2012 at 6:48 PM

ইনফিনিটিভ(অসংখ্য) ধন্যবাদ।অনেক কিছু প্রেক্টিস করতে পারলাম।
সাকিব হাসান,ইবি,কুস্টিয়া,বাংলাদেশ।

Reply

Replies

Zobayer Hasan December 31, 2012 at 1:10 AM

I am really glad that it helped, my best wishes for you :)

Reply

Unknown March 14, 2013 at 2:17 AM

I am trying to learn recursion.I can write easy recursion.But when I am trying to write a recursion for backtracking problem or dp, most of the time I failed.But if I see a medium hard recursive func I relaize it easily.But if I try to write a new recusion most of the time I failed.

Reply

Replies

Zobayer Hasan March 14, 2013 at 2:23 AM

Actually, after solving lots of dp and recursive problem, this is what I would say: If you want to learn dp / backtracking algorithm, you have to solve as many dp as possible, the only way to improve dp skill is to solving more and more. Frustrating at some points, but I don't think there is any other options.

Reply

সোহাগ হোসেন January 23, 2015 at 10:08 AM

wow!! It is very helpful for me and anyone . Thank you so much for this good job.

Reply

nazmul sarker June 14, 2015 at 12:46 AM

thank u so much vai.....really helpful!

Reply

Anonymous December 7, 2015 at 2:24 PM

That's damn important and fun ..... Had a lot of fum :D

Reply

Replies

Zobayer Hasan December 7, 2015 at 2:27 PM

I'm glad that you liked it :)

Reply

Unknown January 18, 2016 at 7:05 PM

I have tested the following code and it give me correct result But the answer script has one extra line before recursive call. the line is "if(*sbest < a[i]) *sbest = a[i];". Is it necessary? If necessary, would you kindly explain to me.

```
void sMax(int i, int n, int *a, int *fbest, int *sbest)
{
if (i == n - 1)
{
*fbest = a[i];
return;

}
```

```
sMax((i + 1), n, a, fbest, sbest);

if (a[i] > *fbest)
{
*sbest = *fbest;
*fbest = a[i];
}
else if (a[i] > *sbest)
*sbest = a[i];

return;

}
```

Reply

Unknown February 5, 2016 at 4:31 PM

কিছু কিছু Problem বুঝতে পারলাম না !

Reply

Replies

Zobayer Hasan February 11, 2016 at 12:33 AM

Which ones?

Reply

Anonymous May 22, 2016 at 12:56 PM

```
int main(){
printf("Thank you\n");
main();
}
```

Reply

Replies

Zobayer Hasan May 22, 2016 at 4:38 PM

haha, nice recursion, (but your program cannot call main())
you're welcome!

Reply

Swad Tasnim June 29, 2016 at 3:24 PM

Can u plz tell me what's wrong with my code? All the time, it returns the correct ans with an extra value. why is there an extra?

```
int call(int i,int n,int a[])
{
if(i<=n)
{

cout<>n;
{
for(int i=0; i>a[i];
cout<<call(0,n-1,a)<<endl;
return 0;
}

}
```

Reply

Replies

Zobayer Hasan July 1, 2016 at 6:52 PM

Hi, your code is broken because of HTML characters. Can you post your code here and then give me the link?
http://codepad.org/

Reply

Unknown July 4, 2016 at 1:05 AM

Thanks so so so so so soooooooooooooooooooooooooooooooooooooooooooo much <3

Reply

Replies

Zobayer Hasan July 6, 2016 at 2:48 PM

You are welcome :)

Reply

Anonymous August 30, 2016 at 7:05 PM

how i can write a recursive problem to generate all permutations of a given n numbers?

Reply

Unknown September 2, 2016 at 11:11 PM

```
//============================================================================
// Name : TowerOfHanoiRecursion.cpp
// Author : Nitish Raj, Scientist, DRDO, raj.nitp@gmail.com
// Version :
// Copyright : No Copyright
// Description : Ansi-style
//============================================================================

#include
#include
#include

#include
using namespace std;


/*
| | |
___|___ __|__ __|__
Source Auxilary Destination

*/
/*Function say whenever you call this second argument will be always from where
you have to element and fourth argument says where you need to put and third used as
spare */

stack A,B,C;

void moveData(char Src, char Des){

switch(Src)
{
case 'a':
{
if(Des == 'b') B.push(A.top());
if(Des == 'c') C.push(A.top());
A.pop();
break;
}
case 'b':
{
if(Des == 'a') A.push(B.top());
if(Des == 'c') C.push(B.top());
B.pop();
break;
}
case 'c':
{
if(Des == 'b') B.push(C.top());
if(Des == 'a') A.push(C.top());
C.pop();
break;
}
}

}
```

```
void ShowDataofStack(){
stack temp;
cout<<"TOWER A :: ";
temp = A;

while(!temp.empty()){
cout<0)
{
HanoiTowerRec(n-1, Source,Destination, Aux); //

// Now nth element left on Source so put this to Destination;
cout<<"_____ "<"<>n;
for(int i =0 ; i<n;i++) A.push(n-i);
HanoiTowerRec(n, 'a', 'b', 'c');



return 0;
}
```

Reply

সাখাওয়াতের ব্লগ October 10, 2016 at 12:52 AM

```
/**
Write a recursive solution to evaluate the previous polynomial for any given x and n.
Like, when x=2 and n=5, we have 1 + x + x2 + ..... + xn-1 = 31

Input:
2 5
Output:
31

**/

#include
using namespace std;
int print(int i,int x,int n,int sum)
{
if(i==1)
{
sum=1;
}
if(i>1)
{
sum+=pow(x,i-1);
}
if(i==n)
return sum;
print(i+1,x,n,sum);
}
int main()
{
cout<<print(1,2,5,0)<<endl;;
return 0;
}
```

ভাইয়া আমি এইভাবে করছি। আমারটা কি হইছে? নাকি কোন বাগ আছে? জানালে ভালো হত।

Reply

Replies

Zobayer Hasan October 10, 2016 at 11:25 PM

Why do you set sum = 1 when i == 1.

Reply

সাখাওয়াতের ব্লগ October 10, 2016 at 1:01 AM

sakhawatshamim35@gmail.com এই ইমেইল ও জানাতে পারেন।

Reply

Programming is fun October 30, 2016 at 11:19 AM

really its too helpful.

Reply

Assasin December 5, 2017 at 12:57 AM

problem 12,20 seems very difficult than others .

Reply

Anonymous February 14, 2018 at 6:18 PM

Can anyone tell what is use of return statement while returning a value in recursion...
What if I don't write return in front of function naming which is not of void type?

Reply

Replies

Zobayer Hasan February 21, 2018 at 3:29 PM

It will raise warning during compilation, also, the method will return a garbage value if any statement that called the function initially was expecting a value. If you do not expect a return value from a function call, then there is not problem. Look at the following example:

```
int a() {
// do not return anything
}

---- in main ---
a(); // no problem
int x = a(); // now there is a problem
```

I hope this clears up the confusion.

Reply

Enter Comment

Subscribe to: Post Comments (Atom)

CATAGORIES

academic study (23) access modifiers (1) ajax (1) algorithm (53) analysis (7) apache (1) backtrack (1) bash (1) beginner (17) bfs (2) bigint (1) binary indexed tree (2) binary tree (1) bit (1 blogger (5) bpm (3) brainfuck (1) brute force (1) bst (1) c (5) c++ (41) changes (1) character device driver (1) chat (3) client (3) combinatorics (2) command prompt (1) common (1) comparator (1) compression (1) (3) confusion (1) connected component (1) console (1) constructible polygon (1) contest (12) crc (1) cse (5) css (1) customize (1) data mining (2) data structure (16) database (3) DCEPC206 (1) decoding (1) dfs (1) disjoint set (1) divide and conquer (3) dp (4) driver (1) dual boot (1) dynamic programming (10) encoding (1) encryption (1) error (2) esoteric language (2) euler circuit (3) euler path (2) euler phi (1) expression evaluation (1) extended euclid (1) facebook (3) factorization (2) fibonacci (1) fix time (1) function (1) funny (15) gcd (2) geometry (6) git (3) github (2) gns3 (2) graph (9) GUANGGUN (1) hacker c hiding window (1) hints (16) holi (1) hopcroft karp (1) huffman (1) incorrect clock (1) inner class (1) instance (1) jar (1) java (8) javascript (2) jdbc (1) jquery (1) kernel programming (2) lab (5) lazy propagation (1) linux (6) ls (1) makefile (1) math (21) matrix (3) matrix algebra (2) matrix exponentiation (2) matrix multiplication (2) maxflow (2) maximum bipartite matching (3) maximum flow (2) merge sort (3) mista (2) module compiling (2) multichat (3) mysql (1) networking (2) number system (1) number theory (8) online judge (4) operating system (1) os (1) other (8) parallel programming (3) pattern (1) phi (1) poj (1) practice (2) primes (5) primit (1) priority queue (1) problem (17) problem classifier (4) problem solving (57) problems solving (1) programming (70) pruning (1) pthreadec qualification round (1) queen (1) queue (1) range maximum query (1) recursion (6) reflection (1) repository (4) rip (1) rmq (1) sample (1) segment tree (2) server (3) shell (1) shell script (1) sieve (4) simulation (3) sc (3) spacing (1) sphere online judge (28) spoj (29) static routing (1) syntax highlighting (1) system programming (4) table tag (1) tc (1) template (4) thread (3) time mismatch (1) time setting (1) topcoder training (6) tree (3) tutorial (10) ubuntu (1) usaco (2) uva (5) uva online judge (5) vector (1) version control (1) web server (1) windows (3)

I am only one, but still I am one.
I cannot do everything, but still I can do something.
And because I cannot do everything I will not refuse to do the something that I can do.
{Helen Keller}