CS101: Computer Programming

Lectures 26, 27, 28

VIDEO LECTURE RECAP QUIZ

Q1. Consider the declaration:

int myEncode (int a, int b);

Which of the following are TRUE?

A. *myEncode* is a variable of type *int*

B. *myEncode* is a function that returns a value of type *int*

C. *myEncode* is a function that can be called only from *main()*

D. *myEncode* is a function that can be called with 0, 1 or 2 parameters

Q2. Which of the following are true of *myEncode* declared as follows:

int myEncode(int a, int b);

A. *'a'* and *'b'* are formal parameters of myEncode

***B.* *'a'* and *'b'* are treated like local variables of myEncode**

***C. main* can have variables 'a' and 'b'**

***D. Within function myEncode, we* can declare local variables named 'a' &'b'**

Q3. Keeping track of the flow of control in a program with functions can be best done using

A. First In First Out Access

B. Last In Last Out Access

C. First in Last Out Access

D. Last in First Out Access

Q4.Which of the following is stored in the call stack of a running program?

A. **Number of statements in each function in the program**

B. **Where to resume execution in caller function once callee function returns**

C. **Number of times each function in the program is called**

D. **Values of local variables in caller function when it calls callee function**

Q5. Which of the following are true about activation records?

A. Each function call creates a new activation record

B. Activation record of callee is popped from call stack when callee returns

C. Activation record of caller is popped from call stack when callee returns

D. Contains information about where to resume execution after callee returns

Q6. Which of the following are TRUE about call-by-value parameter passing?

A. Parameter values copied from caller's activation record to that of callee

B. Changes done on local variables of callee lost when callee returns

C. Allows caller & callee to effectively share variables

D. Both variables and constants can be passed as parameters

Q7. Which of the following are TRUE about call-by-reference param. passing?

A. Can lead to significant savings in memory required for call stack

B. Both variables and constants can be passed as parameters

C. Allows caller and callee to effectively share variables

D. Cannot be used if another parameter is passed by call-by-value

Questions on S422AB

func is a recursive function. Which of the following is true about *func*?

A. *func* calls itself
B. *func(3,5)* can never call *func(3,5)* given other conditions are same
C. Call stack may contain more than one activation record for *func*
D. All calls of *func* should satisfy the pre-condition and post-condition

Consider Virahanka numbers (or Fibonacci Numbers)

A. 0, 1, 3, 5, 8 are Virahanka numbers
B. Number of function calls is greater in the iterative solution than in the recursive solution
C. Number of function calls in recursive solution grows exponentially with n
D. Iterative solution is better than recursive solution

Questions on S423, 424, 425

Write in one-line : what does *findMax(A,currTop, n)* in the following code do?

```
int main() {
  // Declaration, input validation and reading elements to sort
  // Bubble sort
  int currTop, currMaxIndex;
  for (currTop = 0; currTop < n; currTop ++) {
      currMaxIndex = findMax(A, currTop, n);
      swap(A, currTop, currMaxIndex);
  }
  // Rest of code
  return 0;
}
```

What is the value of array in next step of bubble sort in ascending order: 24, 18, 17, 25

A. 17, 18, 24, 25
B. 25, 24, 18, 17
C. 17, 18, 25, 24
D. 18, 17, 24, 25

Number of steps in bubble sort _____ with n

A. Increases Linearly
B. Increases Exponentially
C. Remains constant
D. Increases Quadratically