INTERVIEW QUESTIONS

TECHNICAL DISCUSSION AND INTERVIEW QUESTION AND ANSWERS

PRESCREEN FOR EMBEDDED LINUX POSITION

5. What is a memory leak?
6. How would you debug kernel code?
7. What is virtual memory?
8. What is the pre-requisite in hardware for supporting virtual memory?
9. What are you ideally looking for in your next role?
10. Ability to work in a Startup environment?

STARTUP PRESCREEN QUESTIONS

Home Quiz

For questions 1-4, assume that an array stores integers and has a maximum size of 100. The array is used to store two distinct data structures, BOTH a queue and a stack. The answer should be written in C or C++ and should not use any existing data structure libraries.

1. Write a function that pushes an integer into the stack.
2. Write a function that pops an integer from the stack.
3. Write a function that enqueues an integer into the queue.
4. Write a function that dequeues an integer from the queue.

5. For the given C function, below, please answer the 3 questions below:

```
int Function()
{
    int intArray[N];
    int arraySum = 0;

    for (int i = 0;i< N;i++)
    {
         intArray[i] = i;

         For(int J = 1;j<N;j++)
         {
           intArray[i] = J * intArray[i]
      }
    }
```

```
            For(int k = 0;k<N;k++)
            {
                  arraySum += intArray[k];
            }

            return arraySum;
      }
```

a) Give the O(n) runtime complexity for the function.
b) Give an optimized version (if one exists) for the following pseudo-code function.
c) Give the O(n) runtime complexity for the optimized version.

END OF QUIZ
Answers for Questions 1-4

```
--------------------------------
#define N 100
int intArray[N];
int tos = 0;
int bos = 0;

/* intArray is used for both stack and queue operations
 * so its implemented as a circular array.tos and bos variables
 * are used to control stack and queue start and end
 */

/* Function push
 * pushes element to the stack.
 * If TOS has reached N, checks for position before BOS
 * and inserts the element
 */
void push(int element)
{
  if(tos < N)
  {
    intArray[tos] = element;
    tos ++;
  }
  else if (bos >0)
  {
    intArray[bos-1] = element;
    tos = bos;
  }
  else
    printf("\n Array full");
}

/* Function pop
 * pops element from the stack.
 */
int pop(void)
{
  int element = -1;

  if(tos <= N)
 {  if(intArray[tos-1] == -1)
   { printf("\n Array empty");
     return element;
   }

   element = intArray[--tos];
   intArray[tos] = -1;
   /* tos wrapping around so assign to N */
   if(!tos) tos = N;
   return element;
 }
 else
   printf("\n TOS out of bounds");
  return 0;
}

/* Function Enqueue
 * Enqueues element to the queue
 */
void enqueue(int element)
{
  /* Reusing the push function for enqueue */
  push(element);

}
/* Function Dequeue
 * Dequeues element from the queue
 */
int dequeue(void)
{
 int element = -1;
 if(bos < N)
 {
  if(intArray[bos] == -1)
   { printf("\n Array empty");
     return element;
   }

  element = intArray[bos];
  intArray[bos] = -1;
  bos++;
  }
  else printf("\n BOS out of bounds");
  return element;
}
```

Answers for Question 5
----

a. Order of complexity – O(n^2)
b. Optimized version of the Function

```
int Function()
{
 int arraySum = 0;
 int fact = 1;
/* Find the factorial of N-1 */
for(int i = 1;i<N;i++)

{
  fact *=i;
}

/*Multiply the factorial with i and add*/
for(i=1;i<N;i++)
{
 arraySum += (fact*i);
}
return arraySum;
}
```

c. Complexity of the optimized version – O(N)

MICROSOFT PRESCREEN QUESTIONS

BASIC PROGRAMMING

1. On a modern 64-bit OS such as Windows or Unix, what is sizeof(s) if s is defined as follows:

(a)

```
struct s
{
   char c;
   int i;
};
```

(b)

```
struct s
{
   int i;
   double d;
};
```

(c)

```
struct s
{
   char c;
   char *pc;
};
```

2. Function f represents a simple operation performed on its arguments, and is defined as follows:

```
unsigned int f(unsigned int a, unsigned int b)
{
    return a ? f((a&b) << 1, a^b) : b;
}
```

What is the operation that f performs?

3. What does the following function return:

```
int f(unsigned int a)
{
    int c = 0;
    if (a != 0)
    {
        do
            ++c;
        while (a = a & (a - 1));
    }

    return 0;
}
```

DEBUGGING

1. Find the bug in the following code:

```
// Find the first break in monotonically increasing sequence
// Returns c if there is no break.
int r(int *p, size_t c)
{
    int i = 0;
    while(p[i + 1] >= p[i] && i < c - 1)
        ++i;

    return i + 1;
}
```

MATH, PROBABILITY, COMPLEXITY

1. A rare disease afflicts 1% of the population. A medical test for this disease has 1% false positive rate
(if a person is healthy, there is a 1% probability that the test will show that the person is ill),
and 1% false negative rate (if a person is ill, there is a 1% probability that the test will that the person is healthy).

A person tests as having the disease. What is the probability that the person actually has the disease?

2. An evil dictator captured you and made you play a game. You are in front of three glasses of wine. Two of them are
poisoned; one is not. You must pick one can and drink it. If you survive, the evil dictator will release you.

When you pick one of the glasses, the dictator reveals which one of the other two is poisoned, and offers you to stay
with your original choice, or switch.

Should you switch?

3. In front of you there is a black box. The box can perform two operations: push(N) adds a number to its internal storage;
pop-min() extracts the current minimum of all numbers that are currently stored, and makes the box forget it. The numbers
are mathematical objects: there is no upper bound. Both push(N) and pop-min() execute in O(1) time.

Design and algorithm that could be used to implement such a box.

4. You are asked to design a plotter. A plotter is a computer-controlled device that picks a pen, carries it to a point
on paper using mechanical maniplator, lowers it so that it touches the paper, and drags it to the next point drawing a line.

In your plotter there will be 3 pens, red, green, and blue. Computer uploads a picture to the plotter which consists of list of
segments and colors in which these segments must be drawn. You are asked to reorder the segments such that the work performed
by mechanical manipulator is optimal.

Can you design an algorithm that would do so?

5. What is the time complexity of the following algorithm:

```
unsigned int Rabbits(unsigned int r)
{
   return (r < 2) ? r : Rabbits(r - 1) + Rabbits(r - 2);
}
```

BASIC ALGORITHMS

1. What is the best data structure to implement priority queue?

2. What are the worst case time complexity of the following algorithms, performed on containers of size N:
(a) Locating a number in an unsorted array.
(b) Locating a number in a sorted array.
(c) Inserting a number in a balanced binary tree
(d) Deleting a number from a balanced binary tree
(e) Inserting a number in an unbalanced binary tree
(f) Deleting a number from an unbalanced binary tree
(g) Building a heap
(h) Adding a number to a hash table
(i) Sorting an array

3. What is the relation (less, greater, equal) between O(n) and O(2n)? O(log2 N) and O(log10 N)?

CODING

1. In a two dimensional array of integers of size m x n, for each element which value is zero, set to zero the entire row and
the column where this element is located, and leave the rest of the elements untouched.

EXPECTED TIME TO COMPLETE: 20-30 minutes

2. Write a function that takes an integer argument and returns the corresponding Excel column name.
For instance 1 would return 'A', 2 would return 'B', ..., 27 would return 'AA' and so on

EXPECTED TIME TO COMPLETE: 20-30 minutes

3. Write code to merge 3 sorted arrays of integers (all different sizes) into a single sorted array.

EXPECTED TIME TO COMPLETE: 20-30 minutes

Check this out
http://www.egyed.com/faq/sip_faq.html (http://www.egyed.com/faq/sip_faq.html)

AMAZON PHONE SCREEN

Write c/C++ code to find a given tree is binary tree or not?
I gave the answer, I know it is flawed from all directions. lets discuss in sometime what went wrong here

```
int findbinarytree(tree *btree)
{
static int tmp1 =0;
static int tmp2  =0;
static int isBinaryTree=1;
if(btree == NULL) return 0;
if(!isBinaryTree) return 0;

if(btree->left !=NULL)
{
isBinaryTree = findbinarytree(btree->left);
if(btree->left->data <btree->data) && (tmp1<btree->data){

tmp1 = btree->left->data;
return 1;
```

```
}
else return 0;
if(btree->right !=NULL)
{
isBinaryTree = findbinarytree(btree->right);
if(btree->right->data >btree->data) && (tmp2<btree->data){

tmp2 = btree->right->data;
return 1;
}
```