CS117 (Programming Languages) Fall 2015 HW3 (20 pts.)

1. Consider the following C/C++ code for the interpreter for the simplified infix expression.

```
int Exp(), Term(), Exp2(int), Term2(int), Fact();

int main()
{ printf("result= %d\n", Exp()); }
int Exp()
{ return Exp2(Term()); }
int Term()
{ return Term2(Fact()); }

int Exp2(int inp)
{ int result = inp;
  char a;
  if ((a = getchar()) != '\n')
  { if (a == '+')
       result = Exp2(result + Term());
    else if (a == '-')
       result = Exp2(result - Term());
  }
  return result;
}

int Term2(int inp)
{ int result = inp;
  char a;
  if (( a = getchar()) != '\n')
  { if (a == '*')
       result = Term2(result * Fact());
    else if (a == '/')
       result = Term2(result / Fact());
    else if (a == '+' || a == '-' || a == ')' )
       ungetc(a, stdin);
  }
  return result;
}

int Fact()
{ int result;
  char a;
  a = getchar();
  if (a == '(')
  { result = Exp();
    return result;
  }
  else
    return atoi(&a);
}
```

Assume that the input to this program is: (2 + 5) / 7

(a) Show the activation tree. Please ignore getchar() and atoi().

(b) Show the run time stack (one snapshot) when the right-side parenthesis is processed. (2 + 5) / 7
You don't have to show the details in each A.R. – show only A.R. names.

2. Condider the following Java 2-D array declaration.

```
int[ ] [ ] A = new int[x][y];
```

The formula for computing the address of A[i][j] in the row-major layout is:

```
address(A[i][j]) = address(A[0][0]) + element_size * ( y * i + j )
```

Write the formula for computing the address of A[i][j] in the column-major layout.

3. Consider the following C/C++ code for the binary search program.

```
. . . .
int x[] = { 0, 2, 5, 9, 14, 20, 27, 35, 44, 54, 65, 77, 90 };
int binary_search (int low, int high, int key)
{ int k;
  if (low > high) return 0; //not found
  k = (low + high) / 2;
  if (key == x[k]) return 1; //found
  else if (key < x[k]) return binary_search(low, k-1, key);
  else if (key > x[k]) return binary_search (k+1, high, key);
}
int main()
{ if (binary_search(2, 12, 25)) cout<<"found";
  else cout<<"not_found";
  return 0;
}
```

Show the snapshots of the run time stack (step by step). Show details in the most recently activated frame only in each snapshot, i.e., static_link, dynamic_link, parameter names and values, local variable names and values.