Please obey the rules of examination. If you violate the rules, your answer sheets will be invalid

Surname_____

Student No._____

Southeast University Examination Paper (A)

Course Name Principles of Compiling Examination Term 11-12-2 Score _____

Related Major Computer Science& Technology Examination Form Close test Test Duration 150 Mins

There are 8 problems in this paper. You can write the answers in English or Chinese on the attached paper sheets.

1. Please construct context-free grammars **with ε-free productions** for the following language (10%).

   {ω| ω∈(a,b,c,d)* and the numbers of a's and b's and c's occurred in ω are **odd**, and ω **starts with a , ends with b or c**}

2. Please construct a **DFA with minimum states** for the following regular expression. (15%)

   ((a|b)*|(ab)*)*(a|b)

3. Please **eliminate the left recursions (if there are)** and **extract maximum common left factors (if there are)** from the following context free grammar, and then decide **the resulted grammar** is whether a LL(1) grammar by **constructing the related LL(1) parsing table.**(15%)

   S→begin L end|if E then S|if E then S else S|while E do S|a

L→L;S|S

E→E and E|E or E|b

4. Please **construct a LR(1) parsing table for the following ambiguous grammar with your own defined additional conditions (You determine the required additional conditions by yourself).**(15%)

S→if E then S| if E then S else S|a

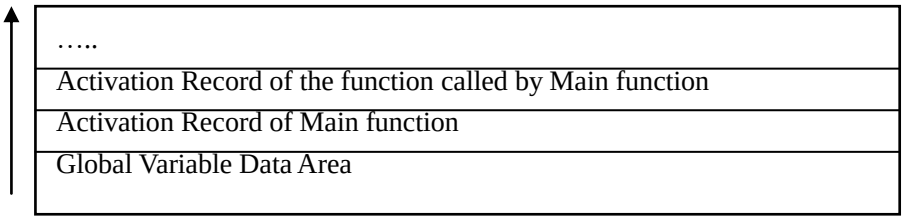
E→ E and E|E or E|b

5. Please construct **an annotated parse tree** for the input string 101.101 where the syntax-directed definition is as following (10%):

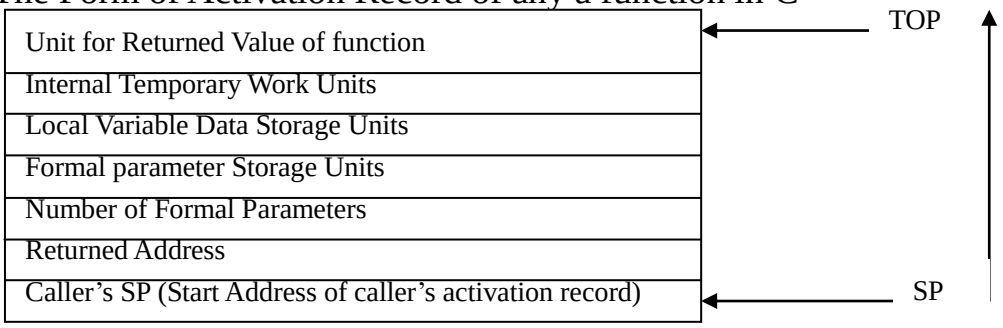
Productions	Semantic Rules
$S \rightarrow L_1.L_2$	$\{S.val=L_1.val+L_2.val/2^{L_2.len}\}$
$S \rightarrow L$	{S.val=L.val}
$L \rightarrow L_1B$	$\{L.val=L_1.val*2+B.val,\ L.len=L_1.len+1\}$
$L \rightarrow B$	{L.val=B.val,L.len=1}
$B \rightarrow 0$	{B.val=0}
$B \rightarrow 1$	{B.val=1}

6. We assume that the storage organization and the form of activation record used in C language program run-time stack storage allocation are as following. Please **construct the run-time stack map when it gets the maximum size** for the following C program (10%).

Storage Organization of C Language



The Form of Activation Record of any a function in C




```
#include <stdio.h>
int x,y,z;

 int main()
 {
    x=10;
    y=6;
    z=C(x,y);
 }

 int C(int m, int n)
{
   if (m>=n)
        if (n<=0) return 1;
        else if (n==1) return m;
```

```
        else {
                int t1;
                t1=m-n;
                if (t1<n)
                 {
                    int t2;
                    t2=C(m,t1);
                    return t2
                 }
                else {
                      int t3,t4,t5,t6,t7;
                      t3=m-1;
                      t4=n-1;
                      t5=C(t3,t4);
                      t6=m*t5;
                      t7=t6/n;
                      return t7;
                 }
            }
    else return 0;
}
```

Notes: 1) Here we assume that the caller's sp of Main function is the start address of global variable data area, and the returned address in the activation record of a function (including Main function) is filled by the operating system automatically, you might not care it.

2) The initial value of variable X is 10 and the initial value of variable Y is 6, the start address of stack used in the program is K.

7. Please translate the following program fragment into **three-address-code (TAC) sequence using short circuit code and back-patching techniques**. (15%)

```
i=1;
loop=0;
while (loop==0 && i<=10) {
   j=1;
   while (loop ==0 && j<=20)
```

```
        if (a[i,j] != b[i,j])
           {
              loop=1;
              m=i;
              n=j;
            }
        else j=j+1;
    if (loop==0) i=i+1;
};
if   (loop==1)   e=0;
else e=1;
```

Notes: Here we assume that the declaration of array A and B are array [1..10,1..20], each data element of array A or B would **use 4 storage unit,** and the start address of array A's storage area is addrA, the start address of array B's storage area is addrB.

8. Please **construct the DAG** for the following basic block, optimize the block and **rewrite the block** in optimized code form. Note that we assume **only Variable L would be used later**(10%)

B=3

D=A+C

E=A*C

F=D+E

G=B*F

H=A+C

I=A*C

J=H+I

K=B*5

L=K+J

M=L