(a) $S \to L . R \quad \{ S.s = L.s + R.s; \}$

$L \to B \, Ls \quad \{ Ls.i = B.s; \quad L.s = Ls.s; \}$

$Ls \to B \, Ls1 \quad \{ Ls1.i = B.s + Ls.i * 2; \; Ls.s = Ls1.s; \}$

$\quad | \; \varepsilon \quad \{ Ls.s = Ls.i; \}$

$R \to B \, Rs \quad \{ Rs.i = 4; \; R.s = B.s / 2 \quad + Rs.s; \}$

$Rs \to B \, Rs1 \quad \{ Rs1.i = Rs.i * 2; \; R.s = B.s / Rs.i + Rs1.s; \}$

$\quad | \; \varepsilon \quad \{ Rs.s = 0; \}$

$B \to 0 \quad \{ B.s = 0; \}$

$\quad | \; 1 \quad \{ B.s = 1; \}$

(b)
```
double S ( ) {
        double ss, ls, rs;
        switch ( token ) {
            case '0' :
            case '1' :
                ls = L ( );
                match ( '.' );
                rs = R ( );
                ss = ls + rs;
                break;
            default: error ( );
        }
        return ss;
}
```

```
double L ( ) {
    double bs, lsi, lss, ls;
    switch ( token ) {
        case 'o' :
        case '1' :
            bs = B ( );
            lsi = bs;
            lss = Ls ( lsi );
            ls = lss;
            break;
        default : error ( );
    }
    return ls;
}
double Ls ( ) {
    double bs, ls1i, lsi, ls1s, lss;
    switch ( token ) {
        case 'o' :
        case '1' :
            bs = B ( );   ls1i = bs + lsi * 2;
            ls1s = Ls ( ls1i );   lss = ls1s;
            break;
        case '.' :  case '$' :
            lss = lsi;      break;
        default : error ( );
    }
    return lss;
}
```

```
double R ( ) {
    double rsi, bs, rss, rs ;
    switch (token) {
        case 'o' :
        case '1' :
            rsi = 4 ;  bs = B ( ) ;  rss = Rs (rsi) ;
            rs = bs / 4  + rss ;  break ;
        default : error ( ) ;
    }
    return rs ;
}
double Rs ( ) {
    double rs1i, rsi, bs, rs1s, rss ;
    switch ( token) {
        case 'o' :
        case '1' :
            rs1i = rsi * 2 ;
            bs = B ( ) ;  rs1s = Rs (rs1i) ;
            rss = bs / rsi  + rs1s ;  break ;
        case $ :
            rss = 0 ;  break ;
        default = error ( ) ;
    }
    return rss ;
}
```

```
double B ( ) {
    double bs ;
    switch (token) {
        case 'o' :
            match('o') ;  bs = 0 ;   break ;
        case '1' :
            match ('1') ;  bs = 1 ;   break ;
        default = error ( ) ;
    }
    return bs ;
}
```