Project2

(1)main (5%)

INPUT:

```
main()
{
}
```

OUTPUT:

```
(program main ( ) { declarations statements })
```

(2)宣告 (5%)

INPUT:

```
main()
{
  int n;
  int s;
  int l;
}
```

OUTPUT:

```
(program main ( ) { (declarations int n ; (declarations int s ; (declarations int l ; declarations))) statements})
```

(3) return; (5%)

INPUT:

```
main()
{
   return;
}
```

OUTPUT:

```
(program main ( ) { declarations (statements (statement return ;) statements) })
```

(4)write (5%)

INPUT:

```
main()
{
   write 1;
}
```

OUTPUT:

```
(program main ( ) { declarations (statements (statement write (arith_expression
(arith_term (arith_factor (primary_expression 1)) arith_term1) arith_expression1) ;)
statements) })
```

(5)read (5%)

INPUT:

```
main()
{
    int n;
  read n;
}
```

OUTPUT:

```
(program main ( ) { (declarations int n ; declarations) (statements (statement read n ;) statements) })
```

(6)assign (6%)

INPUT:

```
main()
{
  int n;
  n = 0;
}
```

OUTPUT:

```
(program main ( ) { (declarations int n ; declarations) (statements (statement n = (arith_expression (arith_term (arith_factor (primary_expression 0)) arith_term1) arith_expression1) ;) statements) })
```

(7)加減法(6%)

INPUT:

```
main()

{

    write 1 + 2 - 3;

}
```

OUTPUT:

```
(program main ( ) {  declarations (statements (statement write (arith_expression
(arith_term (arith_factor(primary_expression 1)) arith_term1) (arith_expression1 +
(arith_term (arith_factor ( primary_expression 2)) arith_term1)(arith_expression1 -
(arith_term (arith_factor ( primary_expression 3)) arith_term1) arith_expression1))) ;)
statements) })
```

(8)正數的四則運算(6%)

INPUT:

```
main()

{

    write 3 - 4 * 6 / 8 % 2;

}
```

OUTPUT:

```
(program main ( ) {  declarations (statements (statement write (arith_expression
(arith_term (arith_factor ( primary_expression 3)) arith_term1) (arith_expression1 -
(arith_term (arith_factor ( primary_expression 4)) (arith_term1 * (arith_factor
( primary_expression 6)) (arith_term1 / (arith_factor ( primary_expression 8))
(arith_term1 % (arith_factor ( primary_expression 2)) arith_term1))))
arith_expression1)) ;) statements) })
```

(9)有負數的四則運算(6%)

INPUT:

```
main()

{

    write 3 + 4 * -6 / -8 % 2;

}
```

OUTPUT:

```
 (program main ( ) { declarations (statements (statement write (arith_expression
(arith_term (arith_factor (primary_expression 3)) arith_term1) (arith_expression1 +
(arith_term (arith_factor (primary_expression 4)) (arith_term1 * (arith_factor -
(arith_factor (primary_expression 6))) (arith_term1 / (arith_factor - (arith_factor
(primary_expression 8))) (arith_term1 % (arith_factor (primary_expression 2))
arith_term1)))) arith_expression1)) ;) statements) })
```

(10)多括號的有負號的四則運算(6%)

INPUT:

```
main()

{

    write (3 + 4) * -6 / -3 % 2;

}
```

OUTPUT:

```
(program main ( ) { declarations (statements (statement write (arith_expression
(arith_term (arith_factor (primary_expression ( (arith_expression (arith_term
(arith_factor (primary_expression 3)) arith_term1) (arith_expression1 + (arith_term
(arith_factor (primary_expression 4)) arith_term1) arith_expression1)) )))
(arith_term1 * (arith_factor - (arith_factor (primary_expression 6))) (arith_term1 /
(arith_factor - (arith_factor (primary_expression 3))) (arith_term1 % (arith_factor
(primary_expression 2)) arith_term1)))) arith_expression1) ;) statements) })
```

(11)簡單 if..fi(6%)

INPUT:

```
main()
{
  if( 2 > 1){
  }fi
}
```

OUTPUT:

```
(program main ( ) { declarations (statements (statement if ( (bool_expression
(bool_term (bool_factor (rel_expression (arith_expression (arith_term (arith_factor
(primary_expression 2)) arith_term1) arith_expression1) (relation_op >)
(arith_expression (arith_term (arith_factor (primary_expression 1)) arith_term1)
arith_expression1))) bool_term1) bool_expression1) ) { statements } else_statement
fi) statements) })
```

(12) 簡單 if..else..fi(6%)

INPUT:

```
main()
{
 if( 2 > 1){
 }else{
 }fi
}
```

OUTPUT:

```
(program main ( ) { declarations (statements (statement if ( (bool_expression
(bool_term (bool_factor (rel_expression (arith_expression (arith_term (arith_factor
(primary_expression 2)) arith_term1) arith_expression1) (relation_op >)
(arith_expression (arith_term (arith_factor (primary_expression 1)) arith_term1)
arith_expression1))) bool_term1) bool_expression1) ) { statements } (else_statement
else { statements }) fi) statements) })
```

(13)布林運算 > <= || (6%)

INPUT:

```
main()
{
   if (3 > 2 || 3 <= 2) {
   }fi
}
```

OUTPUT:

```
(program main ( ) { declarations (statements (statement if ( (bool_expression
(bool_term (bool_factor (rel_expression (arith_expression (arith_term (arith_factor
(primary_expression 3)) arith_term1) arith_expression1) (relation_op >)
(arith_expression (arith_term (arith_factor (primary_expression 2)) arith_term1)
arith_expression1))) bool_term1) (bool_expression1 || (bool_term (bool_factor
(rel_expression (arith_expression (arith_term (arith_factor (primary_expression 3))
arith_term1) arith_expression1) (relation_op <=) (arith_expression (arith_term
(arith_factor (primary_expression 2)) arith_term1) arith_expression1))) bool_term1)
bool_expression1)) ) { statements } else_statement fi) statements) })
```

(14)布林運算 < >= && ! != (6%)

INPUT:

```
main()
{
   if (3 < 2 || 3 >= 2 && ! 2 != 2 ){
   }fi
}
```

OUTPUT:

```
(program main ( ) { declarations (statements (statement if ( (bool_expression
(bool_term (bool_factor (rel_expression (arith_expression (arith_term (arith_factor
(primary_expression 3)) arith_term1) arith_expression1) (relation_op <)
(arith_expression (arith_term (arith_factor (primary_expression 2)) arith_term1)
arith_expression1))) bool_term1) (bool_expression1 || (bool_term (bool_factor
(rel_expression (arith_expression (arith_term (arith_factor (primary_expression 3))
arith_term1) arith_expression1) (relation_op >=) (arith_expression (arith_term
(arith_factor (primary_expression 2)) arith_term1) arith_expression1))) (bool_term1
&& (bool_factor ! (bool_factor (rel_expression (arith_expression (arith_term
(arith_factor (primary_expression 2)) arith_term1) arith_expression1) (relation_op !=)
(arith_expression (arith_term (arith_factor (primary_expression 2)) arith_term1)
arith_expression1)))) bool_term1)) bool_expression1)) ) { statements }
else_statement fi) statements) })
```

15. 布林運算 == (7%)

INPUT:

```
main()

{

    if (3 < 2 || 3 >= 2 &&  2 == 2 ){

    }fi

}
```

OUTPUT:

```
(program main ( ) { declarations (statements (statement if ( (bool_expression
(bool_term (bool_factor (rel_expression (arith_expression (arith_term (arith_factor
(primary_expression 3)) arith_term1) arith_expression1) (relation_op <)
(arith_expression (arith_term (arith_factor (primary_expression 2)) arith_term1)
arith_expression1))) bool_term1) (bool_expression1 || (bool_term (bool_factor
(rel_expression (arith_expression (arith_term (arith_factor (primary_expression 3))
arith_term1) arith_expression1) (relation_op >=) (arith_expression (arith_term
(arith_factor (primary_expression 2)) arith_term1) arith_expression1))) (bool_term1
&& (bool_factor (rel_expression (arith_expression (arith_term (arith_factor
(primary_expression 2)) arith_term1) arith_expression1) (relation_op ==)
(arith_expression (arith_term (arith_factor (primary_expression 2)) arith_term1)
arith_expression1))) bool_term1)) bool_expression1)) ) { statements }
else_statement fi) statements) })
```

16.while 迴圈 (7%)

INPUT:

```
main()
{
   int i;
   while ( i <= 10) {
         i = i + 1;
    }
}
```

OUTPUT:

```
(program main ( ) { ( declarations int i ;  declarations) (statements (statement while
(  (bool_expression (bool_term (bool_factor (rel_expression (arith_expression
 (arith_term (arith_factor (primary_expression i)) arith_term1) arith_expression1)
 (relation_op <=) (arith_expression (arith_term (arith_factor (primary_expression
10)) arith_term1) arith_expression1))) bool_term1) bool_expression1) )
{ (statements (statement i = (arith_expression (arith_term (arith_factor
(primary_expression i)) arith_term1) (arith_expression1 + (arith_term (arith_factor
(primary_expression 1)) arith_term1) arith_expression1)) ;) statements) ) })
statements) })
```

17.題目範例 sum (7%)

INPUT:

```
main()
{
  int n;
  int s;
  int i;
  read n;
  if ( n < 1 ) {
  write -1;
  return;
  } else {
   s = 0;
  } fi
  i = 1;
  while ( i <= n) {
  s = s + i;
  i = i + 1;
  }
  write s;
  return;
}
```

OUTPUT:

```
(program main ( ) { (declarations int n ; (declarations int s ; (declarations int i ;
declarations))) (statements (statement read n ;) (statements (statement if
( (bool_expression (bool_term (bool_factor (rel_expression (arith_expression
(arith_term (arith_factor (primary_expression n)) arith_term1) arith_expression1)
(relation_op <) (arith_expression (arith_term (arith_factor (primary_expression 1))
arith_term1) arith_expression1))) bool_term1) bool_expression1) ) { (statements
(statement write (arith_expression (arith_term (arith_factor - (arith_factor
(primary_expression 1))) arith_term1) arith_expression1) ;) (statements (statement
return ;) statements)) } (else_statement else { (statements (statement s =
(arith_expression (arith_term (arith_factor (primary_expression 0)) arith_term1)
arith_expression1) ;) statements) } fi)) (statements (statement i = (arith_expression
(arith_term (arith_factor (primary_expression 1)) arith_term1) arith_expression1) ;)
(statements (statement while ( (bool_expression (bool_term (bool_factor
(rel_expression (arith_expression (arith_term (arith_factor (primary_expression i))
arith_term1) arith_expression1) (relation_op <=) (arith_expression (arith_term
(arith_factor (primary_expression n)) arith_term1) arith_expression1))) bool_term1)
bool_expression1) ) { (statements (statement s = (arith_expression (arith_term
(arith_factor (primary_expression s)) arith_term1) (arith_expression1 + (arith_term
(arith_factor (primary_expression i)) arith_term1) arith_expression1)) ;) (statements
(statement i = (arith_expression (arith_term (arith_factor (primary_expression i))
arith_term1) (arith_expression1 + (arith_term (arith_factor (primary_expression 1))
arith_term1) arith_expression1)) ;) statements)) }) (statements (statement write
(arith_expression (arith_term (arith_factor (primary_expression s)) arith_term1)
arith_expression1) ;) (statements (statement return ;) statements)))))) })
```